Computer Standards & Interfaces 97 (2026) 104114

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Assessing the quantum readiness of cryptographic standards: Recommendations toward quantum-era compliance

Vikas Chouhan [a,*], Mohammed Aldarwbi [a], Somayeh Sadeghi [a], Ali Ghorbani [a], Aaron Chow [b], Robby Burko [b]

[a] Canadian Institute for Cybersecurity (CIC), University of New Brunswick, Canada
[b] Scotiabank, Toronto, Canada

ARTICLE INFO

Keywords:
Cryptography
Quantum computing
Post-quantum computing
Quantum-readiness
Standardization
Security

ABSTRACT

Cryptography is fundamental to securing digital data and communications, yet established algorithms face increasing risk from emerging quantum capabilities. With the progression of quantum computing, the urgency for cryptographic standards that remain secure in both classical and quantum settings has intensified, governed not only by cryptanalytic risk but also by compliance, interoperability, and country-specific regulatory frameworks. This paper presents a structured evaluation framework that depicts the hierarchy of cryptographic standards, encompassing block ciphers, stream ciphers, hash and MAC functions, key establishment mechanisms, digital signatures, lightweight cryptography, entity authentication, public key infrastructure, and authentication and communication protocols. We define a standards-to-protocol recommendation flow that propagates compliant guidance across layers, from foundational primitives to PKI/authentication and hybridization, and extends to country-specific recommendations and protocols. Our contributions include explicit decision criteria for assessing cryptographic primitives under classical and quantum threat models, yielding both immediate and alternative deployment recommendations aligned with NIST-compliant guidelines. We further analyze hybrid schemes to ensure backward compatibility and secure integration, quantifying storage and network overheads for signatures, encryption, and key exchange to identify practical engineering trade-offs. Consolidated results are presented in reference tables detailing standardization year, purpose, notes, and migration recommendations for both classical and post-quantum contexts. Additionally, we examine the security strength of cryptographic primitives that are currently classically secure or quantum-resistant. This framework offers a reproducible, extensible path toward quantum-ready cryptographic systems.

1. Introduction

The rapid evolution of quantum computing poses an imminent threat to the foundational security of cryptographic systems [1–3]. Traditional cryptographic algorithms, while robust against classical adversaries, are rendered vulnerable in the presence of quantum adversaries due to their reliance on computational problems that quantum computers can solve efficiently [1,3]. This emerging reality underscores the urgent need for the development and adoption of quantum-resistant cryptography to safeguard sensitive information and digital communications in a post-quantum era. Despite notable progress in standardizing post-quantum cryptographic algorithms [4–6], a comprehensive analysis of existing cryptographic standards that cover both classical and post-quantum approaches is still lacking. Moreover, the transition to quantum-resilient systems is fraught with challenges, such as ensuring backward compatibility and balancing security with performance requirements [3,7]. These gaps highlight the critical need for systematic recommendations and hybrid approaches that facilitate a seamless transition while maintaining robust security for current existing systems. This work is motivated by the pressing need to assess the "quantum readiness" of existing cryptographic standards. By analyzing the efforts of standardization organizations and evaluating cryptographic primitives in both symmetric and asymmetric algorithms, this research provides actionable insights and recommendations for achieving compliance in the quantum era. These insights are crucial for guiding researchers, policymakers, and practitioners in adopting cryptographic systems that are resilient against classical and quantum adversaries. Furthermore, this work seeks to emphasize the importance

* Corresponding author.
E-mail addresses: vikas.chouhan@unb.ca (V. Chouhan), m.aldarwbi@unb.ca (M. Aldarwbi), s.sadeghi@unb.ca (S. Sadeghi), ghorbani@unb.ca (A. Ghorbani), aaron.chow@scotiabank.com (A. Chow), robby.burko@scotiabank.com (R. Burko).

https://doi.org/10.1016/j.csi.2025.104114
Received 25 April 2025; Received in revised form 6 November 2025; Accepted 8 December 2025
Available online 17 December 2025

of continued innovation in quantum-resistant cryptography to ensure a secure digital future.

Standards serve as universal benchmarks for cryptographic algorithms and protocols, offer recommendations for secure computer system design, and provide guidance on managing cryptographic information and services. Numerous organizations exist to develop and establish standards, each with distinct origins and purposes. The British Standards Institution (BSI[1]) identifies eight advantages that companies can obtain by adhering to standards. These benefits encompass a variety of areas, such as meeting customer expectations, being cost-effective and efficient in terms of time, conforming to legal requirements, enhancing management practices, exhibiting integrity and building trust, establishing a stronger brand image, and facilitating exportation while maintaining credibility in the global arena. Trust, cost-effectiveness, and time-effectiveness are the main benefits obtained through the use of cryptographic standards.

Standardization involves the establishment and definition of technical standards through the agreement and collaboration of different entities, which can include industry groups, interest groups, standards organizations, and governmental bodies. In the following subsection, we will present an introduction to notable organizations engaged in standardization. This will be followed by an overview of quantum-resistant cryptography, an analysis of standardization recommendations, and a delineation of the contributions and organization of this paper.

1.1. Standardization organizations

Standardization organizations are pivotal in shaping secure and interoperable cryptographic systems by developing consistent global or national standards. At the international level, major bodies like **ISO**,[2] **IEC**,[3] and **ITU**[4] focus on cryptographic standards for information security, telecommunications, and electrical technologies. ISO and IEC jointly manage cryptographic standards through **JTC1**, with subcommittees such as SC27 and SC17 covering IT security and personal identification. The **IETF**[5] and **IANA**[6] play critical roles in Internet-related protocols and resource coordination, with the IETF producing widely respected RFCs. National organizations, including **NIST**[7] (USA), **ANSI**,[8] and **BSI**[9] (UK), contribute by tailoring standards to their domestic needs while aligning with international efforts, NIST notably leads the standardization of post-quantum cryptography.

In the industrial sector, entities such as **IEEE**,[10] **ETSI**,[11] **3GPP**,[12] and consortia like **PKCS** (RSA Laboratories), **SECG**,[13] and the **CA/Browser Forum**[14] contribute to specialized cryptographic standards in wireless networks, ECC, SSL/TLS, and public-key systems. Emerging coalitions like the **PQC Alliance**[15] and **PQC Coalition**[16] have formed to promote and standardize quantum-resistant cryptographic primitives, ensuring preparedness for future quantum threats. These organizations work collaboratively across government, academia, and industry to address evolving security demands, with a growing emphasis on post-quantum cryptography and secure interoperability.

For a detailed overview, a summary of the relevant information is provided in Table 1, with further elaboration available in Appendix A.

1.2. Quantum-resistant cryptography

The impact of quantum computing on cryptography is a major concern for the security community. The potential of quantum computers to quickly solve problems that classical computers cannot solve efficiently, such as factoring large numbers, threatens to compromise many of the currently employed cryptographic algorithms. To address this challenge, significant efforts are underway to develop Post-Quantum Cryptography (PQC) [8] that can resist attacks from quantum computers. NIST has emphasized the importance of developing post-quantum cryptography to ensure the long-term security of digital communication and data. It has issued draft standards for the use of PQC algorithms in various applications, including key establishment, digital signatures, and encryption. However, these standards are still under development and subject to change. Since 2016, NIST has been conducting a project to evaluate and standardize PQC algorithms. The project aims to identify quantum-resistant cryptographic algorithms that can replace the currently used algorithms that will become vulnerable to quantum computer attacks.

NIST has published several rounds of evaluations and results. The first round of submissions included 69 algorithms, which were narrowed down to 26 candidates in the second round. In the third round, they chose 15 candidate algorithms, seven of which were designated as finalists and the remaining eight as alternatives. In 2022, NIST announced that four candidates for standardization had been selected, including one KEM algorithm (CRYSTALS-KYBER) and three Signature algorithms (CRYSTALS-Dilithium, FALCON, and SPHINCS+), with four alternative KEM algorithms (Classic McEliece, HQC, BIKE, and SIKE) continuing into the fourth round [6]. It is important to note that the evaluation process is ongoing, and new algorithms may still be added, or existing algorithms may be modified based on further analysis and feedback from the research community.

Beyond algorithm development, NIST has also addressed the broader challenges of migration. Its recent white paper, Mappings of Migration to PQC Project Capabilities to Risk Framework Documents [9], outlines how organizations can align post-quantum transition activities, such as cryptographic discovery, inventory, and interoperability testing, with established risk management frameworks (e.g., CSF 2.0 and SP 800-53). In parallel, Executive Order (EO) 14144 (January 16, 2025) directs federal agencies to prepare for PQC adoption, including maintaining a CISA-maintained list of product categories where PQC-capable products are widely available and incorporating PQC support into federal procurements [10].

As of 2023, NIST had published three draft Federal Information Processing Standards (FIPS) related to post-quantum cryptography algorithms. On August 13, 2024, NIST released the final versions of the first three post-quantum cryptography standards: FIPS 203, FIPS 204, and FIPS 205. These standards mark a transition from their original algorithm names to new official designations:

(i) CRYSTALS-KYBER is now known as Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), covered under FIPS 203.[17]
(ii) CRYSTALS-DILITHIUM has been renamed to Module-Lattice-Based Digital Signature Standard (ML-DSA), governed by FIPS 204.[18]

[1] https://www.bsigroup.com
[2] https://www.iso.org
[3] https://www.iec.ch
[4] https://www.itu.int
[5] https://www.ietf.org
[6] https://www.ietf.org/process/iana
[7] https://www.nist.gov
[8] http://www.ansi.org
[9] https://www.bsigroup.com
[10] https://standards.ieee.org
[11] http://www.etsi.org
[12] http://www.3gpp.org
[13] http://www.secg.org
[14] https://cabforum.org
[15] https://pqca.org
[16] www.mitre.org/news-insights/news-release/post-quantum-cryptography-coalition-launches
[17] https://csrc.nist.gov/pubs/fips/203/final
[18] https://csrc.nist.gov/pubs/fips/204/final

Table 1
Overview of standardization organizations and their contributions.

Organization Type	Organization	Focus area(s) and contributions
International	ISO/IEC JTC1	IT security, cryptography, biometrics. ISO 27001, ISO/IEC 8730, ISO/IEC TR 14516
	ITU (ITU-T)	Telecommunications, cryptographic network services. X series recommendations (PKI, cryptography)
	IETF/IRTF	Internet protocols and cryptographic RFCs. RFCs (TLS, IPsec, etc.)
	IANA	Internet resource management. DNS root zone, IP address space, protocol registries
National	NIST (USA)	Federal standards, post-quantum cryptography. AES, FIPS series, PQC standardization
	ANSI/X9 (USA)	Financial cryptography. ANSI X9 standards, cooperation with ISO TC68
	BSI (UK)	Information security management systems. BS 7799-2 (basis of ISO 27001)
Industrial	IEEE	Wireless and asymmetric cryptography. IEEE 802.11 (Wi-Fi), IEEE 1363 (asymmetric cryptography)
	3GPP	Mobile communication security. Cryptographic standards for cellular networks
	ETSI	Telecom interoperability. 3GPP collaboration, European telecom standards
	SECG	Elliptic Curve Cryptography (ECC). SEC 1, SEC 2 (ECC standards)
	PKCS (RSA Labs)	Public key cryptography. PKCS #1 - #15: RSA, CMS, key management
	CA/Browser Forum	SSL/TLS certificate policies and PKI trust. Baseline Requirements, EV Guidelines
	OASIS	Security standards, XML, cloud. SAML, ODF, open cybersecurity standards
	PQC Alliance, PQC Coalition	Post-quantum cryptography standardization and awareness. Migration tools, open-source PQC libraries, collaboration across academia and industry

Details of standardization organizations and their contributions are presented in Appendix A.

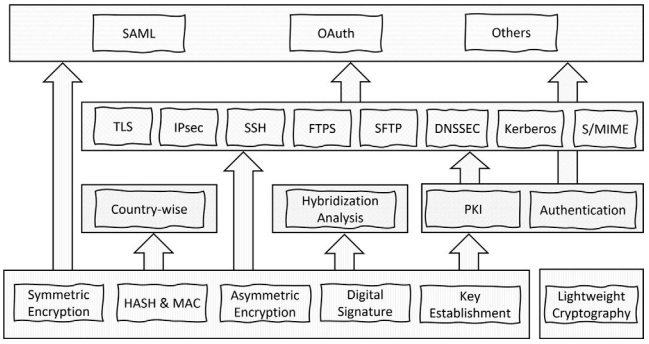

Fig. 1. A structured framework depicting the hierarchy of cryptographic standards for defining the order of recommendation flow.

(iii) SPHINCS is now referred to as Stateless Hash-Based Digital Signature Standard (SLH-DSA), defined under FIPS 205.[19]

These updates mark a significant milestone in adapting cryptographic standards to mitigate the risks posed by quantum computing, ensuring robust security for digital communications and transactions. In addition, NIST has announced that the code-based algorithm HQC will be standardized as a backup key-encapsulation mechanism.[20]

1.3. Standardization and analysis of recommendations

In this subsection, we conduct a detailed analysis of cryptographic standards and recommendations, focusing on various cryptographic primitives and protocols that impact both classical and post-quantum security. We examine a wide range of cryptographic functions, such as symmetric encryption (including block ciphers and stream ciphers), asymmetric encryption, hash functions, message authentication codes (MACs), digital signatures, key establishment mechanisms, lightweight cryptography, entity authentication, and public key infrastructure (PKI). Furthermore, we evaluate widely used cryptographic protocols for secure communication, including HTTPS, SSH, and others, to assess their robustness in both classical and quantum threat environments.

Beyond evaluating existing cryptographic standards, we briefly highlight the emerging role of hybrid solutions as a transitional approach toward quantum security. A detailed analysis of hybridization strategies is provided in Section 7. We also examine global cryptographic standards, considering how different nations are adapting their policies in response to the growing capabilities of quantum computing.

We introduce a structured framework that illustrates the hierarchy of cryptographic standards to define the order of recommendation flow. This framework outlines a systematic migration approach, applying recommendations from the foundational primitives at the bottom to the top-level protocols. The recommendation process begins with foundational cryptographic primitives, such as symmetric encryption, asymmetric encryption, hash functions, and digital signatures. These primitives serve as the core building blocks for more complex security systems. Once these basic primitives are standardized, the process moves to the next layer, which addresses country-specific regulations and the implementation of hybrid public key infrastructure (PKI) and authentication mechanisms. This layer builds on the foundation laid by the primitives, ensuring that cryptographic systems comply with regional and regulatory requirements.

The third layer of our analysis focuses on established protocols for secure communication, including well-known protocols such as TLS/SSL, SSH, and Kerberos. These protocols are essential to ensure secure data exchange over networks and rely on the cryptographic primitives and authentication mechanisms defined in the lower layers. As we move to the top layer, we consider specialized protocols such as SAML (Security Assertion Markup Language) and OAuth, which are commonly used for identity federation and authorization. These higher-level protocols depend on the security established in the lower layers and are critical for ensuring secure access control and identity management in distributed systems.

Through this layered approach, we demonstrate how cryptographic primitives influence the design and security of more complex systems. To support the transition to quantum-resilient systems, we provide detailed recommendations for classical and post-quantum cryptographic systems. These recommendations are intended to guide organizations and researchers in securing their cryptographic infrastructures against both classical and quantum threats.

In this work, our analysis and recommendations are summarized in tables that provide guidance for quantum-resistant cryptography. The tables include detailed columns for clarity: the "Algorithms/Protocols" column lists each cryptographic algorithm or protocol, the "Standards/RFC" column specifies the corresponding standard, the "Year" column indicates when each standard was established, and the "Purpose" column outlines its intended applications. The "Note" column provides additional details, while the final columns present quantum-safe (classical and post-quantum) recommendations for each algorithm or protocol, based on the criteria defined in Section 1.4.

[19] https://csrc.nist.gov/pubs/fips/205/final
[20] https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption

To further illustrate the relationships and flow of cryptographic standards, Fig. 1 presents a structured framework illustrating the hierarchy of cryptographic standards for defining the order of the recommendation flow. This figure visualizes a layered approach to analyzing security standards. The base layer (layer 1) establishes standardized recommendations for the fundamental building blocks of cryptography. These primitives serve as the foundation for the higher layers of the hierarchy, ensuring that cryptographic systems are secure and resilient to both classical and quantum threats. Regarding recommendation flow, in our structured framework, we refer to the foundational or lower-level recommendations of primitives in the higher layers without redefining them.

1.4. Quantum readiness and evaluation criteria

Quantum readiness refers to the preparedness of cryptographic systems to withstand adversarial capability from quantum computers. Quantum computers, due to their ability to solve certain mathematical problems exponentially faster than classical computers, pose a threat to widely used cryptographic algorithms such as RSA and ECC (Elliptic Curve Cryptography). A quantum-ready cryptographic system is one that is secure against quantum computational threats, either through the use of post-quantum cryptographic algorithms, strong symmetric cryptography, or hybrid cryptographic solutions.

To evaluate cryptographic systems based on their readiness for the quantum era, our analysis considers both classical and quantum adversarial models.

1. **Classical Recommendations:** we assume adversaries are limited to classical computing capabilities. Under this assumption, we align with NIST Special Publication NIST SP 800-57 Part 1 Rev. 5, which provides general key-management guidance and classical key-strength recommendations [11]. Based on this guidance, we adopt a target of at least 128-bit classical security (or equivalent as defined in Section 2, Table 3). This target is sufficient to meet the baseline security levels recommended by NIST for pre-quantum threat models.
2. **Post-Quantum (PQ) Recommendations:** we assume adversaries may access or exploit quantum computing capabilities. In this context, we adopt evaluation criteria aligned with NIST's PQC "Security Evaluation Criteria" guidance, which defines five broad security-strength categories (Levels 1–5) (see Section 2) [12]. Based on both key-strength recommendations [11] and NIST's higher security-strength levels [12], we adopt a target of 128-bit quantum-equivalent security (roughly 256-bit classical equivalent) as our migration benchmark and map that target to one of the appropriate NIST PQC levels. These recommendations are categorized into:
    - **Available Recommendations:** PQC algorithms, protocols, and symmetric primitives whose security-strength equivalence is considered under FIPS PQ standards (for example, FIPS 203 for KEMs, FIPS 204 and FIPS 205 for signatures) [13–15].
    - **Alternative Recommendations:** Viable fallback or interim options for scenarios where the primary PQ recommendations are impractical (for example, due to hardware or configuration constraints) but which still meet equivalent security-strength targets.

The evaluation criteria for these quantum-resistant recommendations are critical for reorganizations planning a migration strategy to PQ-secure systems. They guide the assessment of cryptographic primitives under both classical and quantum threat models and were applied in our structured framework to provide quantum-safe recommendations. Our methodology unifies established standards into a migration-oriented evaluation framework. While NIST SP 800-57 Part 1 Rev. 5 addresses classical key management and strength, and the PQC security categories cover quantum-resistant algorithms, neither provides a combined workflow spanning both threat models with migration guidance. Our criteria fill that gap by enabling reorganizations to assess primitives side-by-side, map systems from current classical compliance through transition to full PQ-readiness, and prioritize algorithm selection, risk mitigation, and transition steps in a structured way.

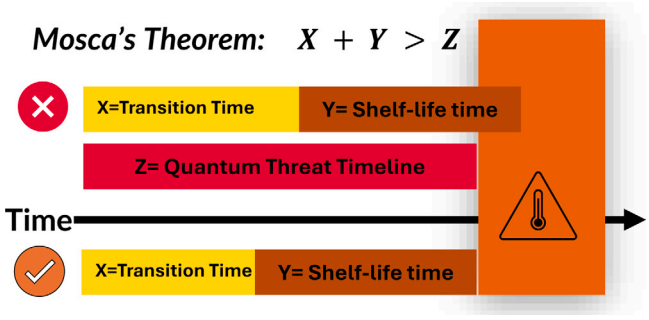

Fig. 2. Mosca's theorem illustrating the relationship between migration time, data lifetime, and the estimated arrival of quantum computing capabilities [16].

1.5. Motivations

The motivation for this work stems from two converging pressures on the cryptographic ecosystem:

1. **The Quantum Threat:** Mosca's Theorem (Fig. 2) provides a simple but powerful framework to reason about urgency. It states that if the time required to migrate to new cryptographic standards (x) plus the required confidentiality period of sensitive data (y) exceeds the estimated time to a cryptographically relevant quantum computer (z), i.e., $x+y \geq z$, then today's encrypted information is already at risk. In other words, data protected with classical algorithms can be collected now and retroactively broken once quantum capabilities arrive. x
   According to the Quantum Threat Timeline Report 2024 [16], experts largely agree that while the next five years pose little immediate risk, the likelihood of a cryptographically relevant quantum computer (CRQC) increases significantly within 10 to 15 years. Within 30 years, nearly 90% of experts expect such machines to exist, with most assigning probabilities above 70%. Although the short-term danger is limited, the long-term threat is widely regarded as inevitable, making proactive preparation essential.
2. **The Compliance Threat:** even before a CRQC materializes, organizations face regulatory and market pressures. Governments and agencies worldwide have issued clear PQC adoption timelines, with many mandating exclusive PQC use by the early 2030s. Failure to comply not only increases technical exposure, but also risks loss of market access and trust (see Table 2).

In summary, both the inevitability of the quantum threat (as captured by Mosca's Theorem and expert forecasts) and the accelerating pace of compliance requirements underline the urgency of migrating to post-quantum cryptography. Since such transitions are complex and multi-year in nature, action must begin well before the threat fully materializes.

Table 2
International guidelines on preferred and mandatory dates for the adoption of PQC. Max lifetime (yrs) indicates the number of years a product is expected to remain on the market.

Guideline	PQC preferred	Only PQC	Max lifetime (yrs)
ASD Australia [17]	2025	2030	5
CNSA 2.0 (NSA) [18]	2025	2033	8
EU NIS CG [19]	2025	2030	5
NIST IR 8547 [20]	2026	2035	10
ANSSI (France) [21]	2025	2030	5
UK NCSC [22]	2027	2035	10

1.6. Contributions

This paper presents a systematic evaluation of global cryptographic standards, encompassing symmetric and asymmetric encryption, hash functions, MACs, digital signatures, key establishment, and secure communication protocols such as TLS, SSH, SAML, and OAuth, with a focus on post-quantum readiness. By integrating classical and quantum threat models, it provides actionable guidance for the design and deployment of secure, future-proof cryptographic infrastructures. The key contributions are:

- We develop a structured evaluation framework to analyze cryptographic primitives and protocols under both classical and quantum adversaries. The framework aligns with NIST guidelines and is illustrated through a layered model that depicts the hierarchy of cryptographic standards and defines the order of recommendation flow.
- We present an integrated perspective that links foundational primitives to higher-level architectures, regulatory compliance, and protocol design. Hybrid approaches are examined to ensure backward compatibility while enabling smooth migration toward quantum-resilient systems.
- We construct structured reference tables that consolidate key attributes, including algorithm, standard, year, purpose, notes, and deployment recommendations. These tables serve as a practical guide for migration planning and cryptographic infrastructure design, offering clear recommendations for both classical and post-quantum contexts based on defined criteria that evaluate quantum readiness by considering their respective security strengths.

Overall, this work bridges existing cryptographic standards with emerging post-quantum requirements, offering a reproducible and extensible framework to support informed decision-making by researchers, policymakers, and practitioners.

1.7. Organization

This research paper conducts a comprehensive investigation into cryptographic standards and recommendations across multiple domains. The structure of the paper is organized as follows: In Section 2, we review the NIST's security strength framework to define security levels. Section 3 delves into the realm of ciphers, exploring established standards and recommended practices. Section 4 scrutinizes the cryptographic standards and recommendations for hash and MAC functions, followed by Section 5, which focuses on cryptographic standards and recommendations for digital signatures. The key establishment algorithms are examined in Section 6, while Section 7 explores the hybridization of cryptographic primitives, considering the prevailing standards and recommendations. We survey the global landscape of cryptographic standards in Section 8, highlighting the variations across countries. Furthermore, Section 9 addresses lightweight cryptography, and Section 10 delves into authentication standards, including public key infrastructure and authentication protocols. Additional communication protocols are discussed in Section 11. A synthesized discussion in Section 12 draws connections between these findings, and the paper concludes in Section 13.

2. Security strength

The main factor to consider when evaluating a cryptographic scheme is how secure it is. However, it is difficult to accurately assess the security of post-quantum cryptosystems, as noted by NIST. This is because there is a risk of discovering new quantum algorithms that could be used to attack these systems, and it is difficult to predict the future capabilities of quantum computers in terms of cost, speed, and memory size.

NIST will create a set of broad security strength categories instead of quantifying the strength of a submitted method using specific estimates of the amount of "bits of security". Each category will be defined by a very simple-to-analyze reference primitive, whose security will serve as a baseline for a wide range of metrics that NIST considers to be potentially relevant to practical security. To fit into multiple categories, a cryptosystem might be implemented with distinct parameter sets. The classification objectives are as follows:

1. To make meaningful performance comparisons between the provided methods easier, make sure that the parameter sets being compared are as secure as possible.
2. To enable NIST to make informed judgments on when to switch to longer keys in the future.
3. To assist contributors in making consistent and rational decisions about the symmetric primitives to utilize in padding mechanisms or other symmetric cryptography-related components of their schemes.
4. To get a better understanding of the security/performance tradeoffs that a particular design style entails.

NIST will base its classification on the range of security strengths offered by existing NIST standards in symmetric cryptography, which NIST expects to offer significant resistance to quantum cryptanalysis, in accordance with the second and third goals above. NIST will create a distinct category for each of the security standards listed below.

1. Any attack that breaks the applicable security criteria must necessitate computing resources equal to or higher than those required for key search on a 128-bit block cipher (e.g., AES128)
2. Any attack that breaks the applicable security criteria must use computational resources equal to or more than those necessary for collision search on a 256-bit hash function (e.g., SHA256/SHA3-256).
3. Any attack that defies the applicable security criteria must necessitate computing resources equal to or higher than those required for key search on a 192-bit block cipher (e.g., AES192)
4. Any attack that violates the applicable security requirement must necessitate computing resources equal to or more than those necessary for collision search on a 384-bit hash function (e.g., SHA384 /SHA3-384)
5. Any attack that violates the applicable security criteria must necessitate computing resources equal to or higher than those required for key search on a 256-bit block cipher (e.g., AES 256)

Computational resources may be quantified using a variety of measures in this case (e.g., number of classical elementary operations, quantum circuit size, etc.). For a cryptosystem to meet one of the aforementioned security requirements, any attack must necessitate computing resources equal to or greater than the specified threshold for all metrics considered by NIST to be potentially relevant to practical security.

Table 3
NIST security levels with classical/quantum security and required gates.

Level	Algorithm	Classicalsecurity	Quantumsecurity	Classical gates	Quantum gates
I	AES 128:	128bits	64bits	2^{143}	2^{157}/MAXDEPTH
II	SHA256/SHA3-256	128bits	80bits	2^{146}	
III	AES192:	192bits	96bits	2^{207}	2^{221}/MAXDEPTH
IV	SHA384/SHA3-384	192bits	128bits	2^{210}	
V	AES256:	256bits	128bits	2^{272}	2^{285}/MAXDEPTH

NIST proposes restricting quantum attacks to a defined operating period or circuit depth. This option is known as MAXDEPTH. This limitation results from the difficulties of performing extremely lengthy serial calculations. Possible MAXDEPTH values range from 240 logical gates (the approximate number of gates that currently envisioned quantum computing architectures are expected to serially perform in a year) to 264 logical gates (the approximate number of gates that current classical computing architectures can serially perform in a decade), with no more than 296 logical gates being possible (the approximate number of gates that atomic-scale qubits with the speed of light propagation times could perform in a millennium) [23].

The complexity of quantum attacks might also be expressed as a function of the size of the circuit. These figures can be compared with the resources needed to decrypt AES and SHA3. At the moment, NIST estimates the classical and quantum gate counts for the optimum key recovery and collision attacks on AES and SHA3, respectively, when the circuit depth is restricted to MAXDEPTH [24,25].

It is worth noting that levels I, III, and V are, for example, characterized in terms of block ciphers that can be cracked using Grover's technique with a quadratic quantum speedup. However, Grover's approach necessitates a lengthy serial calculation, which is difficult to apply in reality. In a genuine attack, numerous smaller instances of the algorithm must be performed in parallel, making the quantum speed-up less dramatic [23].

We have explored the four main types of post-quantum cryptosystems: code-based, lattice-based, supersingular elliptic curve isogeny, and hash-based schemes [26,27]. They have been divided into two main types: encryption candidates and signature candidates. For each type, we compare them based on the underlying problem, the claimed classical security level, the claimed quantum security level, public key size, private key size, and signature size. The comparison of the KEM schemes is shown in Table 4. The comparison of the signature-based schemes is shown in Table 5. However, the practicality of Grover-based attacks remains debated, and recent studies highlight challenges such as circuit depth, fault-tolerant overhead, and quantum decoherence, all of which significantly limit real-world applicability [24,28]. Consequently, while symmetric schemes such as AES-256 and CMAC are often cited as post-quantum resistant under Grover's model, such claims should be interpreted with this caveat in mind.

According to the NIST guidelines for classical systems up to 2030 and beyond, as detailed in the NIST Recommendation for Key Management [11], we strongly recommend ensuring the security of all cryptographic systems based on their equivalence to the security levels (refer to Table 3) of classical systems, as mentioned in Tables 4 and 5.

3. Cryptographic standards and recommendations for ciphers

3.1. Block ciphers standards and recommendations

Block ciphers are a crucial building block in cryptography. They are encryption algorithms that use a symmetric key to operate on fixed-length data blocks, usually 64 or 128 bits. When using a block cipher to encrypt data bits or plaintext, it is recommended to follow a specific mode of operation. Remember that a block cipher can only encrypt a single n-bit string, not a complete message. Additionally, when encrypting a string of data, it is essential to use a mode of operation to prevent information from leaking due to repeated patterns in the data. These modes ensure the encrypted data's confidentiality and integrity. They also prevent specific attacks, such as replay attacks and message modification attacks.

3.1.1. Standards for modes of operation

Organizations such as NIST, ISO, and ANSI have standardized modes of operation that are widely used in various cryptographic libraries and applications. Several standards for modes of operation are available, each providing different security features. Choosing the right mode of operation is crucial since each mode has advantages and disadvantages. Therefore, evaluating the specific requirements of the application is necessary before selecting an appropriate mode of operation [37].

It is vital to employ a mode of operation when using a block cipher such as the Advanced Encryption Standard (AES), which defines applying the cipher to plaintext. The modes of operation most frequently employed for AES block ciphers are Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter (CTR), Cipher Feedback (CFB), Output Feedback (OFB), XEX-based Tweaked Codebook with Ciphertext Stealing (XTS), Galois/Counter Mode (GCM), Cipher-based Message Authentication Code (CMAC), and Counter with CBC-MAC (CCM). Several modes are available for the 3DES cipher block, including TECB-I, TCBC-I, TCFB-I, and TOFB-I. However, it should be noted that these modes will no longer be allowed after 2023 [37,38].

3.1.2. Modes of operation in quantum

Except for ECB and XTS, the stated modes of operation are recognized to be secure against indistinguishability under Chosen Plaintext Attack (IND-CPA) in classical settings, provided that the block cipher used is a Pseudo-Random Function (PRF). Although the security of XTS is uncertain, ECB is generally deemed insecure for many applications for different reasons. The security of ECB can be significantly impacted by the block size, which determines the amount of data that can be safely encrypted using a single key. The ECB mode always generates the same ciphertext block for a given plaintext block encrypted with the same key. When the block size is small, patterns in the plaintext can be exposed in the ciphertext, thereby decreasing the overall security of the ECB [39].

Regarding quantum computing, two types of IND-CPA notions exist, namely "standard IND-CPA" and "IND-qCPA". The standard IND-CPA security notion allows a quantum attacker to make classical encryption queries. In the classical setting, it remains secure, assuming that the underlying block cipher is a PRF. By contrast, IND-qCPA allows the attacker to make quantum encryption queries [39].

In addition, with IND-qCPA, security is ensured even when messages are encrypted using the encryption key that is in a superposition state. The security concept of IND-qCPA aims to offer a type of passive protection by protecting against potential attackers who could gain access to the encryption of specific plaintexts while in a superposition state. In the context of quantum computing, there are two versions of the classical Pseudo-Random Function (PRF): standard secure PRF and quantum secure PRF. In the first version, the function appears to be random when classical queries are made. In the second version, quantum queries that involve multiple superposition inputs cannot determine the function [39].

3.1.3. Recommendations

As depicted in Table 6, current secure block ciphers offer a variety of options for cryptographic operations, ensuring robust protection for sensitive data. In Table 6, we present cryptographic standards for block ciphers, detailing several key columns. For detailed guidelines on table columns and cryptographic recommendations, including the basic

Table 4
Security strength of NIST selected algorithms and 4th round KEM/ENC candidates.

Type	Algorithm	NIST Security level	Classical security (bits)	Quantum security (bits)
Lattice-based	CRYSTALS–Kyber-512	1	128	107
	CRYSTALS–Kyber-768	3	182	165
	CRYSTALS–Kyber-1024	5	256	132
Code-based	BIKE-L1	1	128	–
	BIKE-L3	3	192	–
	BIKE-L5	5	256	–
	HQC-128	1	128	64
	HQC-192	3	192	96
	HQC-256	5	256	128
	Classic-McEliece-348864	1	128	–
	Classic-McEliece-460896	3	192	–
	Classic-McEliece-8192128	5	256	–

Table 5
Security strength of NIST selected signature candidates.

Type	Algorithm	NIST security level	Classical security (bits)	Quantum security (bits)
Lattice-based	CRYSTALS–Dilithium2	2	123	112
	CRYSTALS–Dilithium3	3	182	165
	CRYSTALS–Dilithium5	5	252	229
	Falcon-512	1	120	108
	Falcon-1024	5	277	252
Hash-based	SPHINCS$^+$−128	1	128	64
	SPHINCS$^+$−192	3	192	96
	SPHINCS$^+$−256	5	256	128

Table 6
Standards for block cipher.

Algorithm	Standards	Year	Purpose	Note	Recommendations		
					Classical	PQ	
					(till 2030 & beyond)	(Available)	(Alternatives)
AES	FIPS PUB 197-1 [29]	2023[a]	Description of AES for block ciphers	There are three Rijndael family members: AES-128/192/256. Each member converts data into 128-bit blocks, and the numeric suffix indicates the length of the cryptographic keys used.	AES-128/192/256	AES-256	Camellia-256
	ISO/IEC 18033-3 [30]	2020[a]	Encryption algorithms for block ciphers	Specifies 128-bit block ciphers: AES, Camellia, SEED			
	ISO/DIS 20038 [31]	2017	Banking and related financial services – Key wrap using AES	Describes a method of transporting and storing cryptographic keys using the AES block cipher algorithm.			
3DES	NIST SP 800-67r2 [32]	2017	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher	Determines the Triple Data Encryption Algorithm (TDEA) and its primary component, the Data Encryption Algorithm (DEA).	Disallowed after 2023 (By NIST [32])		
	ISO/IEC 19790 [33]	2018[a]	Security requirements for cryptographic modules	Outlines the encryption systems (known as ciphers) that are used to maintain data confidentiality.			
Camellia	RFC 3713 [34]	2004	A Description of the Camellia Encryption Algorithm	Camellia is a 128-bit block cipher	Camellia-128/192/256	Camellia -256	AES-256
	RFC 5529 [35]	2009	Modes of Operation for Camellia for use with IPsec	Explains how the Camellia block cipher algorithm is utilized in CTR and CCM mode as additional, optional-to-implement IKEv2 and ESP mechanisms	Camellia-CBC-128/192/256	Camellia-CBC-256	
	RFC 4312 [36]	2005	The Camellia Cipher Algorithm and its use With IPsec	Explains how the Camellia block cipher algorithm is utilized in CBC mode.	Camellia-CTR-128/192/256	Camellia-CTR-256	
	ISO/IEC 18033-3:2010 [30]	2020	Encryption algorithms for block ciphers	128-bit block ciphers: AES, Camellia, SEED	Camellia-CCM-128/192/256	Camellia-CCM-256	

[a] Standard was last reviewed and confirmed in the mentioned year.

criteria we follow, refer to Sections 1.3 and 1.4. Based on these standards, for classical security, AES-128/192/256 and Camellia-128/256 are recommended, while 3DES is disallowed. For post-quantum security, AES-256 and Camellia-256 are preferred, with Camellia-256 serving as a viable alternative if AES-256 is impractical. Recommended modes of operation include CBC, CTR, CFB, OFB, GCM, CCM, and CMAC, while ECB and XTS are discouraged due to security weaknesses, particularly against quantum attacks.

3.2. Stream ciphers standards and recommendations

A stream cipher is an encryption method that transforms plaintext into unreadable code byte by byte, using a proper key. This linear encryption technique uses the same key for both the encryption and the decryption of messages. Although stream ciphers are not easily cracked, hackers have found ways to do so.

ChaCha20 is a widely used symmetric cipher, but it does not provide resistance against post-quantum attacks. Under a quantum adversary model, Grover's algorithm yields a quadratic speedup for exhaustive key search (reducing a 256-bit key to 128-bit effective security), and concrete quantum resource estimates for applying Grover's algorithm to ChaCha20 have been published [40]. For more details on the relevant standards, see Table 7.

AES is inherently a block cipher with a fixed block size of 128 bits. However, several standardized modes of operation enable it to function as a stream cipher. Specifically, CTR, OFB, and CFB modes transform AES into a keystream generator, allowing bit- or byte-oriented encryption similar to conventional stream ciphers. Among these, CTR mode is the most widely used due to its efficiency, parallelizable structure, and identical procedures for encryption and decryption. All three modes (CTR, OFB, and CFB) are approved by NIST in SP 800-38 A, confirming their suitability for secure data encryption applications that require stream-like processing.

3.2.1. Recommendations

Stream ciphers are designed for fast, bytewise encryption and are well suited to low-latency applications (e.g., streaming media, VoIP, and real-time data channels). Care must be taken to prevent key or nonce reuse, as keystream compromise can catastrophically break confidentiality. Table 7 summarizes the relevant standards.

Among stream-cipher-style constructions, we recommend *AES-CTR/OFB/CFB* with a 256-bit key as the sole option to retain a conservative security margin against quantum adversaries (via Grover-type search), assuming proper nonce uniqueness and high-entropy keys. Classical stream ciphers (e.g., ChaCha20, RC4, Trivium, Grain, Rabbit) do not provide post-quantum resistance and should be restricted to classical threat models. For confidentiality *and* integrity, pair AES-CTR/OFB/CFB (256) with a strong MAC in an encrypt-then-MAC composition or use an AES-based AEAD where available.

3.3. Asymmetric ciphers standards and recommendations

Asymmetric ciphers provide a way to achieve secure communication between two parties without sharing a secret key, making it easier to establish secure communication between parties who have never communicated before. However, asymmetric ciphers, which require a pair of keys for encryption and decryption, are generally slower and less efficient than symmetric ciphers that use a single key. Additionally, asymmetric ciphers require more computational power and resources, which can be a disadvantage in resource-limited environments.

3.3.1. Standardization

We reviewed classical and post-quantum encryption standards for asymmetric cryptography in Table 8 and provided recommendations for each standard based on the criteria defined in Section 1.4.

The RSA encryption algorithm is extensively used for secure data transmission, digital signatures, and key exchange. The IETF has recommended guidelines for implementing public-key cryptography using the RSA algorithm, which are standardized in RFC 8017.

In contrast, Elliptic Curve Cryptography (ECC) leverages the mathematics of elliptic curves to provide comparable security with significantly smaller key sizes. ECC has become widely adopted in modern protocols such as SSH, PGP, and TLS. For example, the X25519 curve, which is both efficient and secure, serves as the default key exchange method in TLS 1.3 [57]. Furthermore, recent security initiatives have investigated hybrid key-exchange mechanisms that combine classical ECC with post-quantum cryptography to ensure protection against both classical and quantum adversaries. Real-world deployments include hybrid key exchange implementations in Google Chrome, Mozilla Firefox, and OpenSSL 3.0, where schemes such as X25519 together with Kyber are tested in production to achieve a balance between performance and forward-looking security.

On the other hand, NIST is actively working on standardizing post-quantum algorithms and has conducted several rounds of evaluations with their respective results. In 2022, NIST announced that the CRYSTALS-KYBER key encapsulation algorithm to be standardized, along with four alternative KEM algorithms (Classic McEliece, HQC, BIKE, and SIKE) [6].

3.3.2. Recommendations

The asymmetric encryption standards and specifications are listed in Table 8, including both classical and post-quantum algorithms. These standards and specifications are crucial for securing communication applications, such as email and web security, as well as other encryption systems that require key exchange over public networks. By following these recommended standards and algorithms (as per the criteria defined in Section 1.4), users can enhance the security of their cryptographic systems and protect against potential future threats.

3.3.3. Use-case: Phased RSA to kyber migration in financial systems

A global bank aims to strengthen its public-facing TLS infrastructure against prospective quantum threats by transitioning from classical RSA-2048 key exchange to *hybrid* RSA plus Kyber1024 handshake, in accordance with emerging IETF draft ciphersuites and NIST guidance [58].

1. Phased migration plan.

- **P1 Inventory and Compatibility.** Identify all Internet-facing endpoints. Enable hybrid ciphersuites on pilot load balancers and conduct TLS 1.3 interoperability tests with major browsers and HSM firmware.
- **P2 Parallel Deployment.** Serve dual RSA and Kyber key shares during the handshake. Legacy clients utilize the RSA share, while quantum-ready clients process the Kyber share.
- **P3 Gradual Client Rollout.** Employ feature flags and canary releases to incrementally expand hybrid support. Monitor latency and certificate-chain telemetry to ensure performance and stability.
- **P4 Legacy Retirement.** Once at least 95% of sessions negotiate the Kyber share, deprecate pure RSA ciphersuites and rotate remaining keys.

2. Performance & bandwidth impact. Recent evaluations show that Kyber1024 introduces minimal computational overhead compared to classical RSA operations on commodity servers, remaining within acceptable latency thresholds for time-sensitive applications such as online banking and trading [58]. Furthermore, hybrid Kyber+ RSA handshakes have comparable message sizes to traditional RSA-only profiles, avoiding bandwidth penalties for mobile or low-power clients.

Table 7
Standards for stream cipher.

Algorithm	Standard	Year	Purpose	Note	Recommendations		
					Classical	PQ	
					(till 2030 & beyond)	(Available)	(Alternatives)
ChaCha20	RFC 8439 [41]	2018	Defines the ChaCha20 stream cipher as well as the use of the Poly1305 authenticator	Describes the ChaCha20 stream cipher and Poly1305 authenticator as stand-alone algorithms or Authenticated Encryption with Associated Data (AEAD) algorithm.	ChaCha20		AES-CTR/OFB/CFB (256)
	RFC 8103 [42]	2017	Using ChaCha20-Poly1305 Authenticated Encryption in the Cryptographic Message Syntax (CMS)	Outlines the guidelines for utilizing ChaCha20-Poly1305 Authenticated Encryption within the Cryptographic Message Syntax (CMS).			
	RFC 7905 [43]	2016	ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)	Explains how the ChaCha stream cipher and Poly1305 authenticator are utilized in the Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols.			
	RFC 7634 [44]	2015	ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec	ChaCha20 stream cipher and Poly1305 authenticator combined into AEAD algorithm for IKEv2 and IPsec			
RC4	RFC 6229 [45]	2011	Vectors for the Stream Cipher RC4	Includes test vectors designed for the RC4 stream cipher.	RC4 is not recommended		
	ISO/IEC 18033-4:2011 [46]	2022[a]	IT - Security techniques - Encryption algorithms - Part 4: Stream ciphers	Specifies how to combine a keystream with plaintext and provides keystream generators for producing the keystream.			
	IEEE 802.11i-2004 [47]	2004	Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements	Defines MAC and PHY specifications for wireless connectivity for STAs within a local area.			
Enocoro	ISO/IEC 29192-3:2012 [48]	2018[a]	IT - Security techniques - Lightweight cryptography - Part 3: Stream ciphers	Defines lightweight keystream generators with 80 or 128-bit key size for stream ciphers	Enocoro-128		AES-CTR/OFB/CFB (256)
Trivium	ISO/IEC 29192-3:2012 [48]	2018[a]	IT - Security techniques - Lightweight cryptography - Part 3: Stream ciphers	Two dedicated keystream generators are specified for lightweight stream ciphers: Trivium, with a key size of 80 bits	None		AES-CTR/OFB/CFB (256)
	NISTIR 8114 [49]	2018[a]	Presents a summary of NIST's lightweight cryptography project.	Outlines proposals for the standardization of lightweight cryptographic algorithms.			
Grain	ISO/IEC 29167-13:2015 [50]	2021[a]	Specifies the Crypto Suite for Grain-128A for RFID devices based on the ISO/IEC 18000 air interface standards	ISO committees may refer to this common crypto suite for RFID devices used in air interface and application standards.	Grain-128AEAD		AES-CTR/OFB/CFB (256)
	NISTIR 8369 [51]	2021	Select one or more AEAD schemes with optional hashing for constrained environments.	Authenticated Encryption with Associated Data (AEAD)			

(*continued on next page*)

3. Key-management considerations. The referenced evaluation classifies Kyber key material in the *low-size* category, meaning it already fits within standard X.509 v3 extensions with no format changes required. Migration, therefore, focuses on light-touch updates, registering new post-quantum OIDs in certificate transparency logs, and enabling PQC-aware keywrap support in HSM s, rather than wholesale PKI redesign. The operation of TLS in a hybrid RSA+ Kyber keeps legacy clients working while providing an immediate rollback path if post-quantum parameters or standards evolve, all without the protocol-level upheaval sometimes necessary for larger post-quantum key sets.

4. Benefits. The approach delivers quantum resilience without disrupting legacy ecosystems, satisfies payment-card and regulatory crypto-agility requirements, and aligns with the algorithm-selection framework that identifies Kyber as the top-performing KEM for financial use-cases [58].

Table 7 (*continued*).

Algorithm	Standard	Year	Purpose	Note	Recommendations		
					Classical	PQ	
					(till 2030 & beyond)	(Available)	(Alternatives)
Rabbit	ISO/IEC 18033-4:2011 [46]	2022[a]	IT - Security techniques - Encryption algorithms - Part 4: Stream ciphers	Specifies the functions that combine a keystream with plaintext, as well as the keystream generators used to produce the keystream.	Rabbit-128		AES-CTR/OFB/CFB (256)
	RFC 4503 [52]	2006	A Description of the Rabbit Stream Cipher Algorithm	Provides information for the Internet community. It does not specify an Internet standard of any kind			
AES	RFC3686 [53]	2018[a]	AES can act as a stream cipher using CTR, OFB, or CFB modes. CTR mode is the most efficient.	Explains how to use AES Counter Mode with an explicit initialization vector as an IPsec Encapsulating Security Payload confidentiality method.	AES-CTR/OFB/CFB (128, 192, 256)	AES-CTR/OFB/CFB (256)	AES-CTR/OFB/CFB (256)

[a] Standard was last reviewed and confirmed in the mentioned year.

Table 8
Encryption standards for asymmetric cryptography.

Algorithm	Standards	Year	Purpose	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	Alternatives
RSA	IETF RFC 8017 [54]	2016	PKCS #1: RSA Cryptography (Specifications)	MS ≥ 3072	• NIST's KEM PQC • Quantum-safe Hybrid PKE	
ECC	PKCS #13	1998[a]	ECC Standard	KS ≥ 256		
SM2	ISO/IEC 14888 [55]	2018	Provide secure digital signatures and key exchange	KS ≥ 256		
Kyber	NIST[b]	2022	Encapsulation	Kyber-512/768/1024	Kyber-768/1024	McEliece-256, HQC-256, BIKE-256
McEliece				McEliece-128/192/256	McEliece-256	Kyber-768/1024, HQC-256, BIKE-256
HQC				HQC-128/192/256	HQC-256	Kyber-768/1024, McEliece-256, BIKE-256
BIKE				BIKE-128/192/256	BIKE-256	Kyber-768/1024, McEliece-256, HQC-256
SIKE				Not Recommended		

Note: Abbreviations are defined as follows: KS: Key Size and MS: Modulus Size.

[+] The Chinese national standard GM/T 0003-2012 was subsequently adopted and harmonized into the international standard ISO/IEC 14888.

[a] Apparently abandoned, only reference is a proposal from 1998.

[b] In 2022, NIST completed the 4th round of the PQC standardization process [56]. Later, on August 13, 2024, NIST released the final versions of the first three post-quantum cryptography standards: FIPS 203, FIPS 204, and FIPS 205, including Kyber.

3.3.4. Use-case: PQC rollout aligned with EO 14144 for federal systems

A Federal Civilian Executive Branch (FCEB) agency that operates citizen-facing API gateways must comply with Executive Order 14144, which (i) directs CISA to publish a list of product categories where post-quantum cryptography is widely available; (ii) requires agencies to include PQC-support in solicitations for any such product categories within 90 days; (iii) mandates implementation of PQC key establishment or hybrid key establishment (including a PQC algorithm) as soon as practicable when supported by deployed network-security products; and (iv) requires TLS 1.3 (or its successor) support by January 2, 2030 [10].

1. Compliance-driven migration plan (per EO 14144).

E1 Track PQC-Available Categories. Monitor CISA's list of product categories where PQC is widely available (due within 180 days of the executive order). For each in-scope category already deployed (e.g., TLS terminators, service-mesh gateways, email transport, DNS resolvers), record current vendor/firmware and PQC capability status [10].

E2 Procurement Requirements. For any solicitation issued after a category appears on CISA's list, include a requirement that products support PQC (with a 90-day window from listing) [10].

E3 Enable PQC/Hybrid as Soon as Practicable. Where deployed products already provide support, enable PQC key establishment (e.g., ML-KEM) or hybrid key establishment including a PQC algorithm on external and internal TLS 1.3 endpoints, prioritizing public-facing API gateways and cross-agency interfaces [10].

E4 TLS Baseline. Ensure agency systems support TLS 1.3 (or its successor) *no later than January 2, 2030* (OMB for non-NSS; DoD for NSS), with configuration baselines updated accordingly [10].

E5 Audit Evidence. Log negotiated key-exchange groups (e.g., X25519ML-KEM768), certificate OIDs/policies, and configuration state to the SIEM to demonstrate conformance with EO timelines and solicitation clauses [10].

Table 9
Performance evidence supporting EO 14144 compliance: Observed PQC/hybrid impacts for TLS 1.3 show modest overhead, validating the practicability of near-term federal deployment. TTLB: Time-To-Last-Byte. Handshake overhead from Chrome desktop rollout [59]; TTLB results from MADWeb'24 [60]; broader PQ-TLS performance from CCS'23 [61].

Context	KEX/Auth	Median Handshake Overhead	TTLB Overhead (typical)	Key/Msg Size (on-wire)	Source
Web desktop (field)	X25519+ML-KEM-768 (hybrid), ECDSA	~4%	–	ClientHello split	[59]
Stable high-bw nets	ML-KEM-768, ML-DSA-44/65	–	<5% TTLB	~10 KB handshake	[60]
Low-bw (50 KiB+ data)	ML-KEM-768, ML-DSA-44/65	–	<15% TTLB	~11–14 KB auth	[60]
Emulated Web envs	Kyber/HQC; Dilithium/Falcon	On par w/ classical	No hybrid drawback	Param-dependent	[61]

2. Performance & bandwidth impact. Published field measurements and controlled studies report modest overheads for PQC and hybrid key exchanges in TLS 1.3. Chrome's desktop rollout of hybrid Kyber reports a median *~4% handshake-latency increase* due to ClientHello size (packet split), while broader studies show that overall *time-to-last-byte (TTLB)* impact diminishes as payload size grows [59–61] (see Table 9).

3. Key-management & protocol notes. EO 14144 allows agencies to *adopt hybrid key establishment including a PQC algorithm* while existing authentication remains classical, easing near-term deployment on current PKI/HSM stacks. Agencies should prefer TLS 1.3 key-exchange groups that are listed by vendors as compliant with EO-aligned PQC categories (e.g., X25519+ML-KEM-768) and should record policy/OID usage for auditing purposes [10].

4. Benefits. This plan directly satisfies EO 14144's procurement triggers and deployment timelines, achieves *harvest-now/decrypt-later* risk reduction via PQC/hybrid key establishment, and keeps latency within measured bounds for public services [10,59,60].

4. Cryptographic standards and recommendations for hash and MAC functions

A hash function is a mathematical computational procedure that can accept inputs of any size and produce a consistent hash code of a predetermined length that is hard to predict. In cryptography, hash functions are peculiar in that they usually do not depend on secret keys, and therefore their use is restricted to a limited range of security services. As a result, they are often used as a basis for other security systems. If a weak hash function is employed, the security of the overall scheme is undermined. Hash functions are utilized to ensure data integrity in Message Authentication Codes (MACs), create message digests to support digital signature schemes, and generate manipulation detection codes in key establishment and entity authentication procedures. In this section, we reviewed Hash and MAC function standards in Tables 10 and 11, and provided recommendations for specific standards based on the criteria defined in Section 1.4 .

4.1. Hash functions standards

Three major organizations, ISO/IEC JTC1, NIST, and IETF, have developed widely recognized hash function standards, indicating a significant level of standardization in this area.

The NIST secure hash standard, also known as NIST FIPS Pub. 180, has been amended many times, with the most recent version (NIST FIPS Pub. 180-4) released in 2015. This standard defines specialized hash functions such as SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512. The length of the message digests can range from 160 to 512 bits, depending on the algorithm used. The adoption of hash functions has been standardized by the IETF. The SHA family is one of their standardized hash functions and is widely used. However, it is recommended to avoid using SHA-1 for such purposes, as it can no longer guarantee strong cryptographic security. Instead, more secure and advanced hashing algorithms, such as SHA-256 or SHA-3, should be utilized to ensure the confidentiality and integrity of sensitive data.

4.2. Message authentication codes standards

The financial sector utilizes MACs to protect the integrity and millions of daily banking communications, as well as to verify the authenticity of mobile phones. In order for a MAC algorithm to be effective, it must meet the specific requirements outlined in ISO/IEC 9797-1. This standard requires that a MAC algorithm demonstrate the following two properties:

- MAC generation from the plaintext message and a secret key must be straightforward (usability).
- It must be practically impossible to derive an MAC for a particular plaintext message without authorized access to the secret key, even if the intruder has access to accurate MACs for various other messages, including some that the attacker may choose (security).

ISO/IEC 9797 is an extensive collection of standardized MAC algorithms, divided into three parts. Part 1 comprises six distinct variants of CBC-MAC. On the other hand, Part 2 and Part 3 encompass a range of MAC techniques that rely on hash functions.

4.3. Hash and MAC functions recommendations

Tables 10 and 11 list the standards and specifications for the hash and MAC functions. Hash functions provide information on the relevant standards and offer recommendations for their usage in applications such as digital signatures, password storage, and data integrity checks to ensure data integrity. However, MAC functions provide information on relevant standards and offer recommendations for their usage in applications such as network communication protocols, payment systems, and digital rights management systems to ensure data integrity and authenticity. By adhering to the recommended algorithms and standards for hash and MAC functions (as per the criteria defined in Section 1.4), users can enhance data protection and secure their cryptographic systems against potential threats. Our assumption for hash functions is based on prioritizing the highest level of security in the context of potential quantum threats. We consider NIST Level 5 as the minimum requirement because it provides the strongest protection against both classical and quantum attacks. Level 5 ensures that cryptographic algorithms maintain their security even against powerful quantum computers. Therefore, we do not classify SHA-384 as quantum-resistant, as it does not meet the stringent security standards required for NIST Level 5. We view SHAKE256 as offering a degree of quantum resistance when used with longer message digests, such as 512 bits or more. Although SHAKE256 alone does not provide complete quantum resistance, utilizing it with longer output lengths helps mitigate the impact of quantum attacks, including those posed by Grover's algorithm, which effectively reduces security levels.

5. Cryptographic standards and recommendations for digital signatures

Digital signatures have several important properties that make them a powerful tool for ensuring the authenticity and integrity of digital documents and messages. These properties, such as security, efficiency, integrity, and authentication, make it an essential tool to ensure the security and authenticity of digital transactions.

Table 10
Standards for hash functions.

Algorithm	Standards	Year	Purpose	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
SHA-1	IETF RFC 3174 [62]	2001	Secure Hash Algorithm 1	NIST deprecated the use of SHA-1		
	IETF RFC 6234 [63]	2011	The USA Secure Hash Algorithms suite includes SHA, SHA-based HMAC, and HKDF			
SHA-1 and SHA-2	NIST FIPS Pub. 180-4 [64][b]	2015	Secure Hash Standard for both SHA-1 and SHA-2	HL ≥ 256	HL ≥ 512	SHA-2/SHA-3
SHA-3	NIST FIPS Pub. 202 [65]	2015	Permutation-based hash functions (SHA-3-224, SHA-3-256, SHA-3-384, SHA-3-512)			
SHAKE			Extendable-Output functions (SHAKE128 and SHAKE256)			
BLAKE2	IETF RFC 7693 [66]	2015	Provides an explanation of the BLAKE2 cryptographic hash function			
SM3	draft-sca-cfrg-sm3-01 [67]	2018	Provide a secure and efficient cryptographic hash function for data integrity verification and digital signatures			
HASH[c]	ISO/IEC 101184: 1998 [68]	2022[a]	Security techniques for Hash-functions (using modular arithmetic)			
	ISO/IEC 10118-3 [69]	2018	Security techniques for dedicated hash functions			

Note: Abbreviations are defined as follows: HL: Hash Length.
[a] Standard was last reviewed and confirmed in the mentioned year.
[b] In 2023, NIST decided to revise FIPS 180-4.
[c] Comprises a set of security and cryptography standards designed explicitly for HASH.

Several digital signature algorithms are widely used in practice. The choice of digital signature algorithm depends on various factors, such as security requirements, performance considerations, and compatibility with existing systems.

The digital signature employs three primary algorithms: key generation, signing, and verification algorithms. It is important to note that the security of the digital signature system depends on the confidentiality of the private key. If the security of the private key is compromised, malicious entities can generate fraudulent digital signatures that appear legitimate. Therefore, strong cryptographic algorithms and the following best practices for key management and storage are essential. In addition, it is crucial to update the keys to maintain system security periodically.

Various cryptographic algorithms are available to generate public and private keys within digital signature systems. Examples of algorithms that are frequently utilized to generate keys for digital signatures include Rivest-Shamir-Adleman (RSA), Digital Signature Algorithm (DSA), Elliptic Curve Digital Signature Algorithm (ECDSA), and Edwards-curve Digital Signature Algorithm (EdDSA).

5.1. Digital signature standards

Digital signature standards are rules and criteria that specify how digital signatures should be created and verified to guarantee their security and interoperability. Several organizations and governments around the world have developed different digital signature standards. Some of the commonly used digital signature standards include the following:

- **Public Key Cryptography Standard#1 (PKCS#1)** is a widely used standard for RSA digital signatures. It defines the syntax and structure of digital signatures using RSA encryption.
- **Digital Signature Standard (DSS)** is a US government standard developed by NIST for DSA. The DSS specifies the requirements for creating and verifying digital signatures using the DSA. The DSA is a public key cryptography algorithm that is based on the discrete logarithm problem. It uses a pair of keys, a private key and a public key, to sign and verify digital signatures. DSS specifies the key sizes, hash algorithms, and other parameters required for creating and verifying DSA digital signatures. DSS-5, the latest version of the DSS, specifies the DSA for creating and verifying digital signatures.
- **NIST SP 800-131 A Rev. 2** [87] is the revised version of NIST SP 800-131 A. Includes updated guidelines for the use of cryptographic algorithms and key sizes. Specifically, stronger elliptic curves and larger key sizes are now recommended for RSA and DSA. The updated guidelines also provide additional guidance on using the SHA-256 and SHA-384 hash functions and recommend using validated entropy sources for random number generation. Finally, the guidance on the use of digital certificates has been clarified and updated to include recommendations for the use of Certificate Transparency and Online Certificate Status Protocol (OCSP) to enhance certificate revocation checking. It also provides guidance on the transition to more secure cryptographic algorithms and key sizes and best practices for key management, digital signatures, and other security-related processes. It contains a plan to phase out the use of the Triple Data Encryption Algorithm (TDEA).
- **ANSI X9.30.1** [88] is a standard published by the Accredited Standards Committee (ASC) X9 of the ANSI. The standard specifies the requirements for financial institutions and payment processors to securely exchange Electronic Funds Transfer (EFT) data over public networks.
  The purpose of the standard is to ensure and maintain a high level of security for EFT transactions. It is widely used in the financial industry for secure electronic payments and other types of financial transactions.

Table 11
Standards for symmetric digital signature and MAC.

Algorithm	Standards	Year	Purpose	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
CMAC-AES	NIST SP 800-131A Rev. 1 [70]	2015	Symmetric	AES-128,192,256	AES-256	
	ISO/IEC 15946-5 [71]	2022[a]				
	ISO/IEC 19790 [72]	2018[a]				
	ISO/IEC 24727 [73]	2021[a]				
CMAC-3DES	NIST SP 800-131A Rev. 1 [70]	2015		Disallowed after 2023		
	ISO/IEC 19790[72]	2018[a]				
	ISO/IEC 24727 [73]	2021[a]				
MAC[b]	ISO/IEC 29192-6:2019 [74]	2019	IT Lightweight cryptography for MACs (Part 6)	KS ≥ 128 and HL ≥ 256	KS ≥ 256 and HL ≥ 512	
	ISO/IEC 9797-3: 2011/Amd 1: 2020	2020	Security techniques for MACs (universal hash-function)			
	ISO/IEC 9797-2: 2021 [75]	2021	Info. Security for MACs (dedicated hash-function)			
	ISO/IEC 9797-1: 2011 [76]	2022[a]	Security techniques for MACs (using a block cipher)			
HMAC	IETF RFC 2202 [77]	1997	Test cases for HMAC-MD5 and HMAC-SHA-1			
	NIST FIPS 1981 [78]	2008	Keyed-Hash Message Authentication Code (HMAC)			
	IETF RFC 6151 [79]	2011	Updated Security Considerations for the HMAC-MD5			
cSHAKE, KMAC, TupleHash, and ParallelHash	NIST SP 800-185 [80]	2016	SHA-3 Derived Functions			
BLAKE2	IETF RFC 7693 [66]	2015	Provides an explanation of the BLAKE2 cryptographic hash function and Message Authentication Code			
Poly1305	RFC 8439 [41]	2018	Describes the utilization of the Poly1305 authenticator			

Note: Abbreviations are defined as follows: KS: Key Size and HL: Hash Length.
[a] Standard was last reviewed and confirmed in the mentioned year.
[b] It comprises a set of security and cryptography standards designed explicitly for MACs.

- **ANSI X9.62** [81] is a ANSI standard published by the ASC X9. The standard specifies the requirements for the ECDSA, a cryptographic algorithm for generating digital signatures. ANSI X9.62 defines the mathematical operations used in ECDSA and the procedures for generating and verifying digital signatures. The standard also specifies the parameters for elliptic curves that can be used with ECDSA and the key generation and management requirements. Overall, ANSI X9.62 provides a comprehensive standard for the ECDSA and is widely used in the financial industry and other applications where secure digital signatures are required.
- **IEEE 1363** [89] is a standard published by the IEEE for public key cryptography. It specifies several digital signature algorithms, including the DSA and ECDSA.
- **ISO/IEC JTC1** [90] is significant in developing international standards for digital signatures, ensuring that the algorithms and protocols are secure, interoperable, and usable in a wide range of applications. It has developed several standards related to digital signature, including ISO/IEC 14888, ISO/IEC 9796, ISO/IEC 15946, ISO/IEC 19790, and ISO/IEC 24727.
  - ISO/IEC 14888-3 [55] defines the ECDSA digital signature algorithm for use with elliptic curve cryptography. The standard defines the key size requirements for the ECDSA algorithm as follows: (i) The elliptic curve domain parameters must be generated using a cryptographic random number generator. (ii) The key size for the private key should be at least 160 bits. (iii) The key size for the public key depends on the size of the elliptic curve domain parameters. For example, if the elliptic curve has a 160-bit prime field, the public key should be at least 160 bits. (iv) The security level of the ECDSA algorithm is related to the size of the elliptic curve domain parameters. The standard recommends using elliptic curves with a security level equivalent to at least 112 bits (the same level of security as a 3072-bit RSA key).
  - ISO/IEC 9796-3 [84] specifies digital signature schemes that use a hash function, such as DSA and the RSA digital signature algorithm.
  - ISO/IEC 15946-5 [71] is the latest version of ISO/IEC 15946, a standard specifying digital signature schemes based on ECC. It defines multiple digital signature schemes that utilize elliptic curve cryptography, such as the ECDSA and the EdDSA.
  - ISO/IEC 19790:2012 [33] can accommodate various cryptographic algorithms, such as symmetric-key cryptography, public-key cryptography, and elliptic curve cryptography. It is recommended to use keys with a minimum length of 112 bits for symmetric encryption algorithms like AES and a minimum of 2048 bits for public key cryptography algorithms such as RSA. This standard was last reviewed and confirmed in 2018.

Table 12
Standards for asymmetric digital signature.

Algorithm	Standards	Year	Purpose/Type	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
ECDSA	ANSI X9.62 [81] IEEE 1363 (I) [82] FIPS 186-5 [83]	2023 2004 2023	Asymmetric	KS ≥ 256	Not Quantum-Resistant	PQ Signature (Dilithium 3/5, Falcon-1024, SPHINCS-256)
RSAES-PKCS1-v1_5	RFC 8017 ISO/IEC 9796-3 [84] ISO/IEC 14888-1 [85] ISO/IEC 15946-5 [71]	2016 2018[a] 2019[a] 2022[a]		MS ≥ 3072		
RSASSA-PSS	RFC5756 [86] ISO/IEC 24727 [73] NIST-DSS(FIPS 186-5) [83] RFC 8017 [54]	2010 2021[a] 2023 2016				
RSA-OAEP	IEEE 1363 [82] RFC5756 [86]	2004 2010				
Dilithium	NIST [56]	2022[b]	**Post-Quantum**	Dilithium 2/3/5	Dilithium 3/5	Falcon-1024, SPHINCS-256
Falcon				Falcon-512/1024	Falcon-1024	Dilithium 3/5, SPHINCS-256
SPHINCS+				SPHINCS-128/192/256	SPHINCS-256	Dilithium 3/5, Falcon-1024

(*I*) Represents Inactive.
Note: Abbreviations are defined as follows: KS: Key Size and MS: Modulus Size.
Note: RSASSA-PSS combines the RSASP1 and RSAVP1 primitives with the EMSA-PSS encoding method. Similarly, RSASSA-PKCS1-v1_5 combines the RSASP1 and RSAVP1 primitives with the EMSAPKCS1-v1_5 encoding method.
[a] Standard was last reviewed and confirmed in the mentioned year.
[b] Algorithms selected for standardization this year. Later in 2024, NIST finalized the first three post-quantum cryptography standards: FIPS 203, FIPS 204, and FIPS 205, which include Dilithium and SPHINCS+.

  - ISO/IEC 24727-1:2014 It defines a common framework for Identity Management (IdM) systems. This standard was last reviewed and confirmed in 2021. Therefore, this version remains current.

- **FIPS 186** is a US government standard for digital signature algorithms. It specifies the requirements for creating and verifying digital signatures using the DSA and the RSA algorithm.
  FIPS 186-4 [91] specifies several key sizes that can be used with the DSA and the ECDSA. DSA is a widely used algorithm for generating digital signatures and is based on the difficulty of solving the discrete logarithm problem. The standard specifies the following key sizes: 2048, 3072, and 4096 bits. ECDSA is a newer algorithm based on elliptic curve cryptography. It is known for its efficiency and smaller key sizes; the standard specifies the following key sizes: 224, 256, 384, and 521 bits. It will be withdrawn on February 03, 2024.
  For financial applications, a key size of 3072 bits or higher is recommended by some security experts. This provides a high level of security against brute-force attacks and other types of attacks. However, it is important to note that the larger key sizes may also impact the performance and scalability of the system.
  In addition to DSA and ECDSA, FIPS 186-5 [83] specifies several other cryptographic algorithms for key generation and encryption. These include the Secure Hash Algorithm (SHA) family of hash functions, the AES for symmetric encryption, and the RSA algorithm for key exchange and digital signatures.
- **X.509** [11] is a standard for digital certificates that includes digital signature specifications. It defines the format and structure of digital certificates used to verify the signer's identity. They are used to verify the identity of a party in a communication, authenticate the parties involved, and ensure the confidentiality and integrity of the communication.

Table 12 lists a comprehensive checklist of standards and specifications used for digital signatures. This table illustrates a summary of Symmetric, Hash-Based, Asymmetric, and Post-Quantum algorithms usable in the digital signature, with their properties, such as standardization body, last version publication date, type, and our recommendations in terms of classical and post-quantum based on the criteria defined in Section 1.4. As presented, asymmetric algorithms such as RSA, DSA, and ECDSA are not quantum-resistant. In addition, symmetric algorithms will prohibit the use of 3DES after 2023.

5.2. Recommendations

The different algorithms used in the digital signatures in Table 12 ensure that authenticity and integrity are maintained securely and reliably. These algorithms also provide valuable information on the standards that should be followed and offer recommendations on how they can be used in various applications, such as digital documents, messages, and transactions. To ensure the authenticity and integrity of the data, users can improve secure and reliable verification methods by following the recommended algorithms and standards for digital signatures. Our recommendations for post-quantum algorithms (based on the criteria defined in Section 1.4) are aligned with the NIST standardized algorithms. Dilithium, Falcon, and SPHINCS+ are considered secure in both classical and post-quantum contexts, with adjustments to their security levels.

6. Cryptographic standards and recommendations for key establishment

Key establishment is a critical aspect of cryptography, enabling secure communication between two or more parties. It involves the process of generating, exchanging, and managing cryptographic keys. The Key establishment can be broadly categorized into three primary techniques: key exchange, key transport, and post-quantum key encapsulation.

Key exchange is a method through which two or more parties can securely establish a shared secret key, which is then used for

Table 13
Standards for key establishment algorithms: A comparison of standards and recommendations for key exchange, key transport, and post-quantum key encapsulation.

Algorithm	Standards	Issue year	Purpose/type	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
DH	ANSI-X9.63 [92] ANSI-X9.42 [93] RFC-2631 [94] NIST SP-800-56A [95]	2017 2013 1999 2018	Agreement	256, 3072	Not Quantum-Resistant	KEM
ECDH	NIST SP-800-56A [95] RFC 7748 [96] RFC 6278 [97] ISO/IEC 18033-1 [98]	2018 2016 2011 2021	Agreement	256, 3072		KEM
MQV	ANSI X9.63 (I) [92] IEEE-P1363 (I) [99] NIST SP-800-56A [95]	2017 2000 2018	Agreement	256		KEM
RSA	NIST.SP.800-56Br2 [100] NIST.SP.800-131Ar2 [101]	2019 2019	Transport	3072		KEM
Kyber	NIST-PQC [6,13,56]	2022	Encapsulation	Kyber-512/768/1024	Kyber-768 (key size: 9472), Kyber-1024 (key size: 12544)	McEliece(6944,5208,136), BIKE-192(149), BIKE-256(189), HQC-256(128)
McEliece	NIST-PQC [56]	2022	Encapsulation	McEliece-128/192/256	(n, k, t) : (6944, 5208, 136)	Kyber-768, Kyber-1024, BIKE-192(149), BIKE-256(189), HQC-256(128)
HQC	NIST-PQC [56]	2022	Encapsulation	HQC-128/192/256	HQC-256(128)	Kyber-768, Kyber-1024, BIKE-192(149), BIKE-256(189), McEliece(6944,5208,136)
BIKE	NIST-PQC [56]	2022	Encapsulation	BIKE-128/192/256	BIKE-192(149), BIKE-256(189)	Kyber-768, Kyber-1024, McEliece(6944,5208,136), HQC-256(128)
SIKE	NIST-PQC [56]	2022	Encapsulation	Not recommended		

(I) Represents Inactive.

subsequent encryption and decryption of messages. A popular key exchange algorithm is the Diffie–Hellman (DH) protocol. DH is a public-key cryptosystem that allows two users to generate a shared secret key, even if they are communicating over an insecure channel. The security of DH relies on the difficulty of solving the discrete logarithm problem.

Key transport refers to the process of securely transmitting a cryptographic key from one party to another. This technique often employs public-key cryptography to protect the key during transmission. In a typical scenario, the sender encrypts the key with the recipient's public key, and the recipient decrypts it using their private key. The RSA cryptosystem is commonly used for key transport.

Post-quantum key encapsulation is a technique that employs asymmetric cryptographic algorithms believed to be resistant to quantum computer attacks. Quantum computers have the potential to solve certain problems much faster than classical computers, which could potentially break widely used cryptographic schemes such as RSA and ECC. Post-quantum Key Encapsulation Mechanisms (KEMs) are designed to provide key establishment that remains secure even in the presence of quantum computers. One notable post-quantum KEM is the kyber-based KEM. Kyber is a lattice-based cryptosystem that is considered to be resistant to quantum attacks. The kyber-based KEM operates as follows:

1. The recipient generates a public/private key pair based on the kyber algorithm.
2. The sender generates a random "encapsulated key" and encrypts it using the recipient's public key, creating a ciphertext.
3. The sender transmits the ciphertext to the recipient.
4. The recipient decrypts the ciphertext using their private key to obtain the encapsulated key.
5. The encapsulated key can then be used for symmetric encryption and decryption of messages.

Key establishment is a fundamental aspect of secure communication in cryptography. Key exchange, key transport, and post-quantum key encapsulation are three techniques that enable the secure generation, transmission, and management of cryptographic keys. By understanding these methods and selecting the most suitable approach for a given scenario, secure communication can be achieved, protecting sensitive information from unauthorized access and ensuring robustness against potential quantum computer threats.

6.1. Key establishment standards

In this section, we will explore several key establishment standards, including those developed by ANSI, IETF, NIST, ISO, etc. These standards provide guidelines for the implementation of secure key establishment schemes, including symmetric and asymmetric key algorithms, key management practices, and protocols for key exchange and transport. Additionally, we will examine the emerging field of post-quantum cryptography and the ongoing efforts to develop key establishment algorithms that can resist attacks from quantum computers. Table 13 outlines these standards and offers recommendations based on the criteria defined in Section 1.4.

- **ANSI X9**: In 1985, ANSI released its initial standards related to key management, including ANSI X9.17 [102] and ANSI X9.24 [103]. These standards were designed to address the key management issues facing financial institutions. Specifically, ANSI X9.17 focused on key management between banking establishments,

while ANSI X9.24 dealt with key management between retail devices and the financial institutions that managed them, such as point-of-sale devices.

  - **ANSI X9.17** [102]: This standard was primarily aimed at using single-length DES keys for secure key establishment within financial institutions. However, it became apparent that the protection provided by single-length DES keys did not meet the stringent security requirements of these institutions (refer to Section 3.1 for an in-depth discussion on DES). Consequently, the ANSI X9.17 standard was officially **withdrawn** in 1999.
  - **ANSI X9.24** [103]: Focuses exclusively on symmetric key management and standardization techniques that employ symmetric cryptography. Although it does not explicitly outline a key management framework, the standard implies one by imposing stringent rules governing the usage and manipulation of keys.

- **IETF RFC 2631:** This standard, established by the IETF, outlines the Diffie–Hellman key agreement method for public key cryptography. It enables secure key exchange between parties over an insecure communication channel, without requiring any pre-shared secrets. By specifying the protocol, message formats, and cryptographic calculations, RFC 2631 facilitates secure communication and data protection over the Internet.
- **NIST SP 800-56A** [95]: This standard, developed by the NIST, provides guidelines for implementing key establishment schemes using asymmetric key pairs. Covers key agreement and transport protocols, specifying requirements for secure key exchange and key derivation functions. The purpose of NIST SP 800-56 A is to ensure secure communication and data protection by promoting the proper use of cryptographic techniques in key establishment processes.
- **RFC 7748:** This standard, published by the IETF, specifies two elliptic curves for use in cryptography: Curve25519 and Curve448. These curves are designed for key agreement using the Elliptic Curve Diffie–Hellman (ECDH) protocol. RFC 7748 provides detailed information on the mathematical properties, coordinate systems, and encoding formats of these curves. The primary goal of the standard is to enhance security and performance in cryptographic applications, such as key exchange and digital signatures, by promoting the use of these modern elliptic curves.
- **IEEE-P1363 (I)** [89]: IEEE P1363 is a set of standards developed by the IEEE for public-key cryptography. It includes several related standards, each addressing different aspects of public-key techniques. The P1363 standard defines methods for key agreement using public-key cryptography, such as Diffie–Hellman and Elliptic Curve Cryptography. It also specifies methods for key transport based on encryption, with RSA commonly used in these schemes.
- **RFC 6278:** This standard outlines the utilization of the static-static ECDH key agreement scheme within the Cryptographic Message Syntax. In this approach, both sender and receiver employ long-term, static Diffie–Hellman values, which are stored in certificates. The document provides guidance on implementing this key agreement method to improve secure communication between parties.
- **ISO/IEC 18033-1** [98]: This standard, published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), provides an overview of the ISO/IEC 18033 series on encryption algorithms. As the first part of the series, ISO/IEC 18033-1 defines general concepts, terminology, and security requirements for various cryptographic techniques, including symmetric key algorithms, asymmetric key algorithms, and mechanisms for key exchange and management. The purpose of this standard is to establish a common foundation for the development, evaluation, and comparison of encryption algorithms and protocols, ensuring secure communication and data protection.
- **NIST SP 800-56B Rev. 2** [104]: This standard, issued by NIST, focuses on key establishment schemes using Integer Factorization Cryptography (IFC), primarily the RSA algorithm. In the context of key transport, NIST SP 800-56B Rev. 2 provides guidelines for secure key-wrapping and unwrapping, ensuring the protection of cryptographic keying material during transmission between parties. The standard aims to promote the proper use of cryptographic techniques for key transport in order to maintain secure communication and data protection.
- **NIST SP 800-131 A Rev. 2** [105]: This standard, published by NIST, provides guidance on the transition from older cryptographic algorithms and key lengths to more secure ones in order to enhance the security of federal information systems. NIST SP 800-131 A Rev. 2 outlines recommendations for key management practices, focusing on the usage of secure cryptographic algorithms and key sizes for various applications, including key establishment, digital signatures, and encryption. The standard aims to improve the overall security of information systems by promoting the adoption of up-to-date cryptographic techniques and key management practices.
- **NIST Post-Quantum Cryptography** [6]: NIST initiated the Post-Quantum Cryptography (PQC) project to address the potential threats posed by quantum computers to current cryptographic algorithms. The project aims to develop, standardize, and promote the adoption of new cryptographic algorithms that can withstand attacks from quantum computers, ensuring long-term security for communication and data protection. NIST is in the process of evaluating and selecting candidate algorithms for key exchange, digital signatures, and encryption that are resistant to quantum attacks, with the goal of incorporating them into future cryptographic standards.

6.2. Recommendations

Table 13 presents an overview of key establishment algorithms, including key exchange, key transport, and post-quantum key encapsulation mechanisms. It provides information on the relevant standards and offers recommendations based on the criteria defined in Section 1.4 for their use in secure communication and data protection applications.

According to traditional recommendations, all surveyed algorithms generally provide sufficient security, with the exception of SIKE, which has been reported as compromised. In the case of RSA, a key length of 3072 bits is considered secure. For MQV, a key length of 256 bits is recommended. For DH and ECDH, key pair lengths of (256, 3072) are considered robust. Regarding post-quantum algorithms, aside from SIKE, all other analyzed algorithms appear to offer satisfactory security.

On the other hand, none of the classical algorithms has demonstrated the ability to effectively resist quantum attacks. Remarkably, not all post-quantum algorithms are equipped to withstand such attacks either. This implies that not all versions of these algorithms are quantum-resistant; only those that ensure a 128-bit security level against post-quantum attack, namely: Kyber-768, Kyber-1024, McEliece (6944, 5208, 136), HQC-256, BIKE-192, and BIKE-256, have proven to be resistant.

7. Hybridization of cryptographic primitives: Standards, recommendations, and analysis

Post-quantum hybridization analysis is a rapidly developing area of research that seeks to create solutions by combining classical and post-quantum algorithms to withstand the threat of quantum computers. This approach is a promising way to address the challenges posed by

Table 14
Number of messages stored/sent for storage and network.

Operation	Storage				Network		
	PK	Pr	Sign	Cipher	PK	Sign	Cipher
Signature	2	1	2	0	1	1	0
Encryption	2	1	0	2	1	0	1
Key Exchange	2	2	0	2	2	0	2

quantum computing and plays a critical role in developing effective hybrid cryptographic solutions. By leveraging the strengths of both classical and post-quantum mechanisms, we can design robust protocols that protect sensitive data. As we move toward a post-quantum era, this analysis remains essential to ensure data security against emerging quantum threats.

The post-quantum hybridization analysis aims to evaluate various combinations of classical and post-quantum cryptographic primitives, such as hash functions, encryption schemes, and signature schemes, to identify the most promising combinations that can offer high levels of security, efficiency, and compatibility with existing systems. Researchers in this field use techniques such as mathematical modeling, simulation, and experimentation to analyze the complexity of operations, available resources, and the size of the data to be processed. The outcome of post-quantum hybridization analysis is recommendations for designing and implementing secure and efficient cryptographic protocols that are resistant to quantum attacks. These recommendations are critical for organizations that rely on cryptography to protect sensitive data, such as healthcare providers, governments, and financial institutions.

This section begins by assessing the feasibility of the resulting hybrid cryptographic algorithms in detail. Our analysis covers various aspects such as signature, encryption, and key exchange. Furthermore, we thoroughly evaluate the storage and network overhead associated with hybrid algorithms, examining their computational complexity and resistance to attacks.

7.1. Hybridization analysis

A hybrid system employs at least two algorithms, such as classical and post-quantum algorithms. The anticipated storage and network costs of the hybrid approach are the sum of the respective individual algorithm approximation costs utilized in the hybrid configuration. The analysis provided here delves into the employment of RSA as the classical standard, post-quantum algorithms, and hybrid algorithms (incorporating both classical and post-quantum approaches) across three key functionalities: digital signatures, encryption, and key exchange. Our evaluation encompasses all classical and post-quantum algorithms operating at level V security to ensure a comprehensive performance assessment and comparison. The primary focus of this comparison lies in scrutinizing key storage and network costs, measured in bytes. We have excluded the SIKE algorithm as it has been reported to be broken [106]. The costs associated with these functionalities, as mentioned in Table 14, are interpreted based on the required number of messages as follows:

7.1.1. Storage cost

In terms of storage costs, presented below is a concise overview of the incurred costs:

(i) Digital Signature Cost Computation:

- For Classical or PQ
  - At Sender End: Pk, Pr, Signature;
  - At Receiver End: Pk, Signature;
  - Total Cost: 2Pk + Pr + 2Signature;
- For Hybrid:
  - At Sender: 2Pk, 2Pr, Signature
  - At Receiver End: 2Pk, Signature;
  - Total: 4Pk + 2Pr + 2Signature;

(ii) Encryption Hybrid Cost Computation:

- For Classical or PQ
  - Receiver: Pk, Pr, Cipher;
  - Sender: Pk and Cipher;
  - Total Cost: 2Pk + Pr + 2Cipher;
- For Hybrid:
  - Receiver: 2Pk, 2Pr, Cipher;
  - Sender: 2Pk and Cipher;
  - Total Cost: 4Pk + 2Pr + 2Cipher;

(iii) Key Exchange Cost Computation:

- Classical Key Exchange Cost:
  - At User End (for each user): Pk, Pr, Cipher;
  - Received from the other user: Pk and Cipher;
  - Final key (256-bit): 32byte;
  - Total Cost: 2Pk + 2Pr + 2Cipher + 2*32
- Post-quantum Key Exchange Cost:
  - At User End (for each user): Pk, Pr, Cipher;
  - Received from the other user: Pk, Cipher;
  - Final key (256-bit): 32byte;
  - Total Cost: 2Pk + 2Pr + 2Cipher +2*32

Table 15
Digital signature performance analysis.

Approach	Algorithm	Storage cost	Network cost
Classical	RSA-15360	9600	3840
PQ	Dilithium5	19 238	7187
	Falcon-1024	**8551**	**3123**
	SPHINCS+	99 968	49 920
Hybrid	RSA-Dilithium5	28 838	11 027
	RSA-Falcon-1024	**18 151**	**6963**
	RSA-SPHINCS+	109 568	53 760

Table 16
Key exchange performance analysis.

Approach	Algorithm	Storage cost	Network cost
Classical	RSA-15360	19 264	7680
PQ	McEliece	4218140	2095090
	HQC-256	101490	43 428
	Kyber1024	**18944**	**6272**
Hybrid	RSA-McEliece	4237340	2102770
	RSA-HQC-256	120690	51108
	RSA-Kyber1024	**38144**	**13952**

Table 17
Encryption performance analysis.

Approach	Algorithm	Storage cost	Network cost
Classical	RSA-15360	9600	3840
PQ	McEliece	2 109 038	1047545
	HQC-256	50 713	21 714
	Kyber-1024	**9440**	**3136**
Hybrid	RSA-McEliece	2 118 638	1051385
	RSA-HQC-256	60 313	25 554
	RSA-Kyber-1024	**19 040**	**6976**

- Hybrid:
  - At User End (for each user): 2Pk, 2Pr, Cipher;
  - Received from the other user: 2Pk, Cipher;
  - Final key (256-bit): 32byte;
  - Total Hybrid Cost: 4Pk + 4Pr + 2Cipher + 2*32

7.1.2. Network cost

In terms of network costs, presented below is a concise overview of the incurred costs:

(i) Digital Signature Cost:

- For Classical or PQ
  - Sender Transmits: Pk, Signature;
  - Total Cost: Pk + Signature
- Total Hybrid Cost: 2Pk + 2Signature

(ii) Encryption Cost:

- For Classical or PQ
  - Sender Transmits: Pk, Cipher;
  - Total Cost: Pk + Cipher
- Total Hybrid Cost: 2Pk + 2Cipher

(iii) Key Exchange Cost:

- Classical Key Exchange:
  - Sender Transmits Pk, Cipher;
  - Total Cost: Pk + Cipher
- Post-quantum Key Exchange:
  - Sender Transmits: Pk, Cipher;
  - Total Cost: Pk + Cipher
- Hybrid:
  - Sender Transmits: 2Pk + 2Cipher
  - Total Cost: 2Pk + 2Cipher

7.2. Findings

In analyzing the performance of the digital signature presented in Table 15, Falcon-1024 emerges as the optimal choice in terms of storage and network costs among the post-quantum candidates. This characteristic makes Falcon-1024 the most economically viable option within the hybrid approach for RSA-Falcon-1024. As we delve into encryption and key exchange performance, Tables 16 and 17 exhibit Kyber-1024's efficiency from a cost perspective. Consequently, a fusion of Kyber-1024 with RSA (RSA-Kyber1024) yields the most favorable overall cost result.

7.3. Standards and recommendations

Table 18 presents several hybrid approaches, some of which have already been standardized or are currently undergoing standardization in draft form. Based on our extensive research, we have identified several notable hybrid approaches, specifically KEM, encryption, and key exchange. For each standard, we provide information on the year of publication or proposal, the intended purpose, any noteworthy aspects, and recommendations based on the criteria defined in Section 1.4 regarding the use of classical and post-quantum algorithms to achieve hybridization.

8. Worldwide cryptographic standards and security recommendations across countries

This section provides an overview of publicly available cryptographic standards or guidelines for several countries, including Canada, Australia, USA, UK, Japan, and others. These standards and guidelines are designed to enhance the security of cryptographic systems and mitigate potential threats.

8.1. Recommendations

We have compiled country-specific recommendations from official sources, such as government or standards websites, in Table 19. These recommendations are categorized according to different algorithm types, including Hash and MAC, Symmetric Encryption, Asymmetric Encryption, Digital Signature, and Key Establishment.

We examined various cryptographic standards and offered potential classical and post-quantum recommendations to protect against potential threats. Referring to our suggested recommendations, the entries highlighted in bold in Table 19 indicate algorithms that, according to our analysis, are deemed inappropriate. Furthermore, in Table 22, we present our assessment of the security strengths of established cryptographic algorithms. The table reflects our analysis of whether these algorithms are currently classically secure, projected to remain so beyond 2030, or are quantum-resistant, based on the criteria defined in Section 1.4. It also provides alternative recommendations for the quantum era derived from our evaluation.

9. Cryptographic standards and recommendations for light-weight cryptography

Since 2015, NIST has led the effort to standardize lightweight cryptography that is suitable for resource-constrained environments such as IoT devices and embedded systems. The aim was to identify Authenticated Encryption with Associated Data (AEAD) schemes and optional hashing capabilities that share common components to minimize implementation size and complexity.

Table 18
IETF RFCs/Drafts on hybrid approaches.

Algo-rithm	RFC/Draft	Year	Purpose	Note	Recommendations
KEM	draft-ounsworth-cfrg-kem-combiners-03 [107]	2023	Defines combiner function for Hybrid KEMs	Expired in September 2023	• Any classical algorithm should provide at least a 128-bit security level, such as RSA-3072, ECC-256, etc. • Only NIST's KEM PQC that provides 128 security bits should be used
Encryption	9180 [108]	2022	Hybrid Public Key Encryption (HPKE)	HPKE applies to various combinations of asymmetric KEM, KDF, and AEAD encryption functions	
	draft-westerbaan-cfrg-hpke-xyber768d00-02 [109]	2023	**Defines X25519Kyber768Draft00, a hybrid PQ KEM, for HPKE (RFC9180)**	Expired in October 2023	
	draft-irtf-cfrg-dnhpke-04 [110]	2022	Deterministic Nonce-less Hybrid Public Key Encryption	Expired in August 2024	
Key Exch.	draft-kampanakis-curdle-ssh-pq-ke-01 [111]	2023	**Defines PQ hybrid KE methods**	These methods are formulated for application within the context of SSH/TLS	
	draft-tls-westerbaan-xyber768d00-03 [112]	2023	**Defines X25519Kyber768Draft00-03, a hybrid PQ KE for TLS 1.3**	Expired in March 2024	
	draft-ietf-tls-hybrid-design-10 [113]	2023	Hybrid key exchange within the context of the TLS 1.3	Expired in October 2024	

Table 19
Country-wise recommendation.

Algorithm		Canada	Australia	USA-NIST	USA-NSA	Japan (till next rev)	UK	ECRYPT-CSA
HASH and MAC	Hash	SHA-2/SHA-3≥ **224** (256 beyond 2030)	SHA-2≥ **224**	SHA-2/SHA-3≥**224** (256 beyond 2030)	SHA-384 or SHA-512	SHA-1; SHA-2≥ **224**; SHA-3≥ 256; SHAKE128, SHAKE256	SHA-256	SHA-2/SHA-3≥ **224** (256 beyond 2030)
	MAC	KS ≥ 128	—	KS ≥ 128	—	KS ≥ 128	—	—
Symmetric Encryption	Algorithm	AES≥ 128	AES≥ 128; TDES(using 3 Distinct Keys (3DES)); (must not use ECB mode)	AES≥ 128	AES-256	AES≥128; Camellia≥128	—	AES/Camellia/Serpent ≥ 128
	Mode	ECB (not allowed after 2023), CBC, CTR, CFB, OFB, XTS, GCM, CMAC, CBC-MAC, CCM	ECB (Must not use), CBC, CTR, OFB, CFB	ECB (not allowed after 2023), CBC, CTR, CFB, OFB, XTS, GCM, CMAC, CBC-MAC, CCM	—	ECB (not allowed after 2023), CBC, CTR, CFB, OFB, XTS, GCM, CMAC, CBC-MAC, CCM	ECB (not allowed after 2023), CBC, CTR, CFB, OFB, XTS, GCM	OFB, CFB, CTR, CBC, ECB (No. acceptable), XTS, EME, FFX, CCM, CWC, GCM, OCB, EAX, GCM, ChaCha20+Poly1305
Asymmetric Encryption		RSA/DSA ≥**2048** (3072 beyond 2030); ECDSA ≥224 (256 beyond 2030)	DH ≥**1024**; RSA/DSA ≥**2048**; ECDSA/ECDH ≥**160**	FFC(DSA, DH,MQV): K_pu≥**2048** (3072 beyond2030), K_pr≥**224** (256 beyond2030); IFC(RSA) ≥ **2048** (3072 beyond2030); ECC(ECDSA, EdDSA, DH,MQV): f ≥ **224** (256 beyond2030);	—	RSA-OAEP (≥**2048**), ECC (key sizes 256, 384 or 521)	—	RSA/DSA≥3072, ECC ≥ **224**
Digital Signatures					XMSS; LMS (SHA-256/192); CRYSTALS-Dilithium (Level V); ECDSA (Curve P-384); RSA≥3072	DSA p≥**2048**, q≥**224**; ECSDA ≥224; RSASSA-PKCS1-v1_5/ RSASSA-PSS≥**2048**;	RSA **2048**; ECDSA-256 P-256 Curve	
Key Establishment		FFC/DH/MQV ≥ **2048** (3072 beyond 2030); ECC-CDH or ECC-MQV ≥ **224** (256 beyond 2030)			ECDH (Curve P-384); DH ≥ 3072; RSA≥3072; CRYSTALS-Kyber(Level V)	DH/MQV p≥**2048**, q≥**224**; ECDH/ECMQV ≥**224**	—	

Note: p is the prime modulus and q is the prime divisor, f is the range, and "–" means not available/mentioned.
Abbreviations are defined as follows: LMS: Leighton–Micali Signature, MQV: Menezes–Qu–Vanstone, FFC: Finite Field Cryptography, PSS: Probabilistic Signature Scheme, XMSS: eXtended Merkle Signature Scheme, MS: Modulus Size, KS: Key Size, HL: Hash Length.

9.1. Lightweight AEAD standards and recommendations

During March 2021, NIST revealed a list of 10 finalists, including ASCON, Elephant, GIFT-COFB, Grain-128AEAD, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, and Xoodyak. These schemes were chosen to proceed to the final round of the selection process, which is shown in Table 20. In February 2023, NIST announced its decision to standardize the ASCON family for lightweight cryptography

Table 20
Authenticated Encryption with Associated Data (AEAD) standardized algorithms.

Algorithm	Standards	Year	Purpose (building block)	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
Ascon [116]	NIST [117]	2023	A permutation-based scheme that uses the monkeyDuplex construction	Ascon-128, Ascon-128a, Ascon-80pq	–	SCHWAEMM256-256, TinyJAMBU-256
Elephant [118]			A permutation-based scheme that follows a nonce-based encrypt-then-MAC construction	Jumbo (127), Delirium (127)	–	SCHWAEMM256-256, TinyJAMBU-256
GIFT-COFB [119]			A block-cipher based where the underlying block cipher is GIFT-128 and the mode is Combined Feedback	GIFT-COFB(128)	–	SCHWAEMM256-256, TinyJAMBU-256
Grain [120]			A stream cipher optimized for hardware implementations	Grain-128AEAD	–	SCHWAEMM256-256, TinyJAMBU-256
ISAP [121]			A permutation-based with a nonce-based encrypt-then-MAC construction mode	ISAP-A-128a, ISAP-K-128a, ISAP-A-128, ISAP-K-128	–	SCHWAEMM256-256, TinyJAMBU-256
PHOTON [122]			A permutation-based approach with a combined feedback mode	PHOTON-Beetle-AEAD	–	SCHWAEMM256-256, TinyJAMBU-256
Romulus [123]			A scheme that is based on the tweakable block cipher Skinny with MAC-then-Encrypt mode	Romulus-N, Romulus-M	–	SCHWAEMM256-256, TinyJAMBU-256
SPARKLE [124]			SPARKLE permutations-based operations that apply multiple distinct instances of Alzette – a four-round 64-bit block cipher	SCHWAEMM192–192(184), SCHWAEMM256-256(248)	SCHWAEMM256-256	TinyJAMBU-256
TinyJAMBU [125]			A keyed permutation that is based on a 128-bit nonlinear feedback shift register	TinyJAMBU-192, TinyJAMBU-256	TinyJAMBU-256	SCHWAEMM256-256
Xoodyak [126]			Built from a fixed 384-bit permutation (called Xoodoo) operated in Cyclist mode	Xoodyak-128	–	SCHWAEMM256-256, TinyJAMBU-256

applications. On June 20, 2023, NIST's report provides a public record of the final round of the standardization process and explains the evaluation of the finalists that were chosen for standardization [114]. The finalized standard, NIST SP 800-232, was released in August 2025 and specifies Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 [115].

9.1.1. Recommendations

In the quantum setting, Grover's algorithm roughly halves the effective security of symmetric keys. Thus, a 128-bit key provides only about 2^{64} quantum security. Larger keys (or hash output) are required to restore the margin. NIST's LWC status report [114] and the final standard [115] both highlight these post-quantum considerations and note which finalists or standardized variants offered > 128-bit options.

- **Default for ultra-constrained IoT (≈64-bit PQ):** *Ascon-AEAD 128/128a*, as standardized in SP 800-232 [115], remains the recommended baseline choice. Other 128-bit key finalists without known quantum-specific weaknesses (e.g. *GIFT-COFB*, *Grain-128AEAD*, *ISAP*, *PHOTON-Beetle*, *Romulus*, *Xoodyak*) also achieve similar PQ levels. Public analyses and NIST reviews indicate only generic Grover-type impacts or preimage bounds; no faster quantum attacks are known for these schemes [114,127,128].
- **Small PQ bump with minimal overhead:** *ASCON-80pq* (160-bit key) was added by the designers specifically to increase resistance to quantum key search, while keeping the same design philosophy as ASCON-128 [129].
- **High PQ margin (targeting ≥100–128-bit PQ):** choose *SPARKLE/Schwaemm256-256* or *TinyJAMBU-256*. Independent work estimates Grover-attack resources for Schwaemm and finds no shortcut beyond the square-root speed-up; NIST also points to these larger-key options for PQ-sensitive deployments [114,130].

Note. NIST remarks that the standardized ASCON variants do not include a 256-bit key option. If approximately 128-bit post-quantum security (symmetric-only) is required, a 256-bit candidate such as AES-GCM-256 should be used [114].

9.2. Lightweight hash standards and recommendations

Table 21 shows the standard lightweight hash algorithms that have been recognized as standards by NIST. Five algorithms have the ability to be used as hash functions in addition to their normal functionality as (AEAD) algorithms.

9.2.1. Recommendations

Every lightweight hash standard algorithm presented in Table 21 provides at least one version that guarantees adequate security for 2030 and beyond. However, according to our research, none of these hash algorithms appears to offer security against quantum attacks.

Table 21 lists the standardized lightweight hash functions (some finalists also offer hash/XOF modes). NIST's final-round report for the LWC process summarizes published *quantum preimage* resource estimates for ASCON-Hash, PHOTON-Beetle-Hash, and Xoodyak-Hash; independent studies provide similar estimates for SPARKLE/Esch [131,132]. These results support the standard guidance:

Table 21
Lightweight hash functions standardized algorithms.

Algorithm	Standards	Year	Purpose (Building Block)	Recommendations		
				Classical	PQ	
				(till 2030 & beyond)	(Available)	(Alternatives)
Ascon [116]	NIST [117]	2023	A permutation-based approach that uses the monkeyDuplex construction	ASCON-Hash-256, ASCON-Hasha-256		–
PHOTON-Beetle-hash [122]			A permutation-based is based on a sponge structure	PHOTON-Beetle-Hash-256		–
Romulus [123]			A permutation-based with a combined feedback mode	Romulus-H-256		–
SPARKLE [124]			SPARKLE permutations-based that apply multiple distinct instances of Alzette – a 4-round 64-bit block cipher	ESCH256-256, ESCH384-384		–
Xoodyak [126]			Built from a fixed 384-bit permutation (called Xoodoo) operated in Cyclist mode	Xoodyak-256		–

- For **preimage** security at approximately 128-bit strength against quantum adversaries, select a **256-bit output** (e.g., Ascon-Hash-256, PHOTON-Beetle-Hash-256, Esch-256, Xoodyak-Hash-256/XOF) [131,132].
- If **collision** resistance near 128-bit *quantum* strength is required, use a **384-bit output** (e.g., ESCH-384), since collisions admit an $O(2^{n/3})$ quantum attack [133,134].

In short, these lightweight hashes remain *quantum-resistant up to the generic bounds*. Choose the output length to meet your target quantum security level.

10. Cryptographic standards and recommendations for authentication

10.1. Entity-based: Authentication standards

We reviewed standards for authentication mechanisms as detailed in Table 23, listing recommendations for each standard based on its cryptographic support. The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have collaboratively introduced a multipart standard known as ISO/IEC 9798. This standard outlines a collection of authentication mechanisms designed for general-purpose applications. The series currently consists of six sections, each addressing specific aspects as follows:

- **ISO/IEC 9798-1:** serves as a foundational component, providing definitions for key terms and notation. It also presents a basic model of an authentication mechanism.
- **ISO/IEC 9798-2:** details mechanisms that rely on symmetric encryption for their operation.
- **ISO/IEC 9798-3:** outlines mechanisms centered on the utilization of digital signatures.
- **ISO/IEC 9798-4:** specifies mechanisms that make use of MACs.
- **ISO/IEC 9798-5:** focuses on a variety of zero-knowledge techniques.
- **ISO/IEC 9798-6:** involving manual data transfer mechanisms.

The protocols outlined in ISO/IEC 9798 have been designed with versatility to cater to various application domains, aiming to achieve maximum robustness.

Additional organizations, namely ITU-T, NIST, and the IETF, have also introduced standards for entity authentication. For instance, ITU-T X.509, recognized for its specifications related to public key and attribute certificates, encompasses three authentication protocols that rely on digital signatures. Similarly, NIST FIPS Pub. 196 presents a collection of authentication protocols centered on digital signatures, although these are a subset of those defined in ISO/IEC 9798-3.

Internet RFC 4120 specifies Version 5 of the Kerberos network authentication protocol, which is based on symmetric cryptography. Another protocol, named S/KEY, tailored for user authentication to remote servers is detailed in RFC 1760 [144]. This protocol relies on a cryptographic hash function. In addition, RFC 1507 [142] introduces an authentication protocol (DASS) that uses digital signatures. It should be noted that specific authentication requisites for the Internet context are discussed in RFC 1704 [143].

Multiple cryptographic mechanisms are available to ensure the integrity and origin authentication of individual protocol messages. We examine five primary options, forming the foundation of the mechanisms standardized within ISO/IEC 9798.

- Symmetric encryption
- Use of a MAC function
- Use of a digital signature
- Use of asymmetric encryption
- Use of other asymmetric cryptographic techniques

10.1.1. Recommendations

Table 23 presents a comprehensive list of authentication standards and specifications. It includes recommendations for each standard based on their cryptographic support, which are crucial for ensuring the security of various authentication applications, such as web security, online banking, and secure remote access. Implementing these recommended standards and algorithms allows users to strengthen their cryptographic systems and protect themselves against potential future threats.

10.2. Certificate-based: Public key infrastructure

The primary emphasis of the Public Key Infrastructure (PKI) revolves around producing, disseminating, and continually administering public key certificates. These certificates encompass a set of data elements that comprise a public key, multiple identifiers linked to the key holder, and additional information (such as an expiration date). All of these components are subjected to digital signing by the Trusted Third Party (TTP), recognized as a Certificate Authority (CA). This digital signature ensures the cohesive association of the diverse elements within the certificate.

A PKI can consist of diverse components that vary based on the technology and organizational framework employed to distribute and ensure the accuracy of public keys.

This encompasses a variety of entity types, comprising:

Table 22
Security Strength Assessment of Cryptographic Algorithms: classical security and post-quantum readiness.

Type	Algorithm	Variant	Security Level	Classically Secure (Current)	Classically Secure (till 2030 & beyond)	Quantum Resistance	Quantum Era (Alternatives)
Symmetric Block cipher	AES	AES-128	128	Yes	Yes	No	Camellia-256
		AES-192	192	Yes	Yes	No	
		AES-256	256	Yes	Yes	Yes	
	DES			Deprecated			AES-256, Camellia-256
	Triple DES						
	Camellia	Camellia-128	128	Yes	Yes	No	Camellia-256
		Camellia-192	192	Yes	Yes	No	
		Camellia-256	256	Yes	Yes	Yes	
Symmetric Stream Cipher	RC4			Deprecated			AES-CTR/OFB/CFB (256)
	Grain	Grain-128	128	Yes	Yes	No	
	Trivium	Trivium-80	80	No	No	No	
	Rabbit	Rabbit-128	128	Yes	Yes	No	
	Enocoro	Enocoro-128	128	Yes	Yes	No	
	AES-CTR/OFB/CFB	AES-CTR/OFB/CFB-128	128	Yes	Yes	No	AES-CTR/OFB/CFB (256)
		AES-CTR/OFB/CFB-192	192	Yes	Yes	No	
		AES-CTR/OFB/CFB (256)	256	Yes	Yes	Yes	
	ChaCha	ChaCha20	256	Yes	Yes	No	AES-CTR/OFB/CFB (256)
Asymmetric	RSA	RSA-2048	112	Yes	No	No	NIST PQ
		RSA-3072	128		Yes		
		RSA-7680	192				
		RSA-15360	256				
	ECC	ECC-224	112	Yes	No	No	
		ECC-256	128		Yes		
		ECC-384	192				
		ECC-512	256				
	SM2	SM2-256	256	Yes	Yes	No	
HASH	SHA-1			Deprecated			SHA/SHA3-512, SHAKE-256
	SHA-2	SHA-224	112	Yes	No	No	SHA3-512, SHAKE-256
		SHA-256	128	Yes	Yes		
		SHA-384	192	Yes	Yes		
		SHA-512	256	Yes	Yes	Yes	
	SHA-3	SHA-224	112	Yes	No	No	SHA-512, SHAKE-256
		SHA-256	128	Yes	Yes		
		SHA-384	192	Yes	Yes		
		SHA-512	256	Yes	Yes	Yes	
	SHAKE	SHAKE-128	128	Yes	Yes	No	SHA/SHA3-512
		SHAKE-256	256	Yes	Yes	Yes	
	SM3	SM3-256	256	Yes	Yes	No	SHA/SHA3-512, SHAKE-256

(*continued on next page*)

- **CAs:** are accountable for producing public key certificates following a specified certification practice statement. These certificates can be interpretable within the framework of a defined certificate policy.
- **Registration Authorities (RAs):** fulfill the role of authenticating the identities of individuals who seek to obtain a certificate from a CA
- **Certificate Repositories:** are entities that securely store and provide access to public key certificates.
- **Certificate Status Servers:** are entities that, upon request, provide online information about the current validity status of a certificate.

Usually, each certificate is assigned a distinctive serial number. Consequently, the Certificate Revocation List (CRL) only has to encompass the serial numbers of certificates that have been revoked.

10.2.1. Recommendations

Public key cryptography standards and specifications play a crucial role in ensuring the security of communications and data protection. Tables 24 and 25 list these standards, providing classical and post-quantum recommendations for each standard based on the types of cryptography it supports. These PKI standards and specifications are vital to establishing secure and trusted communication channels. They provide a framework for secure key exchange, digital signatures, and encryption. By adhering to these recommended standards

Table 22 (*continued*).

Type	Algorithm	Variant	Security Level	Classically Secure (Current)	Classically Secure (till 2030 & beyond)	Quantum Resistance	Quantum Era (Alternatives)
MAC	HMAC	SHA-1	Deprecated				CMAC, GMAC, BLAKE2-512
		SHA-2	Similar to the SHA-2 algorithm assessment listed above				
		SHA-3	Similar to the SHA-3 algorithm assessment listed above				
	CMAC	AES					HMAC, GMAC, BLAKE2-512
	GMAC						HMAC, CMAC, BLAKE2-512
	Poly1305	Poly1305-128	128	Yes	No		HMAC, CMAC, GMAC, BLAKE2-512
	UMAC	UMAC-64	64	No			
		UMAC-128	128	Yes	Yes	No	
	BLAKE2	BLAKE2-256	128	Yes	Yes	No	HMAC, CMAC, GMAC
		BLAKE2-512	256	Yes	Yes	Yes	
Asymmetric Signature	ECDSA	Similar to the ECC algorithm assessment listed above					NIST PQ
	RSAES-PKCS RSASSA-PSS RSA-OAEP	Similar to the RSA algorithm assessment listed above					
PQ KEM	Kyber	Kyber-512	128	Yes	Yes	No	McEliece-192/256 HQC-256 BIKE-192/256
		Kyber-768	192			Yes	
		Kyber-1024	256				
	McEliece	McEliece-128	128	Yes	Yes	No	Kyber-768/1024 HQC-256 BIKE-192/256
		McEliece-192	192			Yes	
		McEliece-256	256				
	HQC	HQC-128	128	Yes	Yes	No	Kyber-768/1024 McEliece-192/256 BIKE-192/256
		HQC-192	192			Yes	
		HQC-256	256				
	BIKE	BIKE-128	128	Yes	Yes	No	Kyber-768/1024 McEliece-192/256 HQC-256
		BIKE-192	192			Yes	
		BIKE-256	256				
PQ Signature	Dilithium	Dilithium-2	128	Yes	Yes	No	Falcon-1024 SPHINICS-192/256
		Dilithium-3	192			Yes	
		Dilithium-5	256				
	Falcon	Falcon-512	256	Yes	Yes	No	Dilithium-3/5 SPHINICS-192/256
		Falcon-1024	512			Yes	
	SPHINCS+	SPHINCS+-128	128	Yes	Yes	No	Dilithium-3/5 Falcon-1024
		SPHINCS+-192	192			Yes	
		SPHINCS+-256	256				
Key Establishment	DH	Similar to the RSA algorithm assessment listed above					PQC
	ECDH	Similar to the ECC algorithm assessment listed above					
	MQV	Similar to the RSA algorithm assessment listed above					

and algorithms, organizations can achieve the following benefits and advantages: robust security, confidentiality, authentication, integrity, non-repudiation, and interoperability. In summary, adhering to the recommended PKI standards and algorithms ensures the security and reliability of the communication channels. By implementing PKI, organizations can protect sensitive information, establish trust among parties, and mitigate the risks associated with unauthorized access, tampering, and denial of involvement.

10.3. Standards and recommendations for authentication protocols

10.3.1. Kerberos standards and recommendations

The Kerberos protocol is widely used to authenticate service requests between trusted hosts over untrusted networks, such as the Internet. Ensure secure communication by employing cryptographic standards. Table 26 provides a comprehensive overview of these standards, listing both classical and post-quantum recommendations for each standard based on its cryptographic support. Through a thorough examination of the security aspects of these standards, we provide recommendations to enhance the effectiveness of Kerberos. The protocol utilizes symmetric key cryptography to establish strong authentication mechanisms, encryption algorithms to protect data integrity, and secure key distribution techniques for session key establishment. In general, Kerberos relies on robust cryptographic standards to secure communication channels and prevent unauthorized access in untrusted environments.

10.3.2. DNSSEC standards and recommendations

The DNSSEC protocol protects Internet users and applications from forged DNS data by utilizing public key cryptography to electronically

Table 23
Standards for authentication mechanisms.

Objective	Standard	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
Authentication	ISO/IEC 9798-1 [135]	2021[a]	Security approaches for entity authentication	• Defines an authentication model along with overarching prerequisites and limitations • Support: Asymmetric encryption and signatures Symmetric encryption Hash algorithms	• Refer to Table 8 for asymmetric encryption • Refer to Table 12 for asymmetric digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 for hash algorithm
	ISO/IEC 9798-2 [136]	2019		• Approaches utilizing authenticated encryption • Support: Symmetric encryption	• Refer to Table 6 for symmetric encryption
	ISO/IEC 9798-3 [137]	2019		• Procedures utilizing digital signature methods • Support: Asymmetric digital signature	• Refer to Table 12 for asymmetric digital signature
	ISO/IEC 9798-4 [138]	2021[a]		• Approaches incorporating cryptographic check function • Support: MAC algorithm	• Refer to Table 11 for MAC algorithm
	ISO/IEC 9798-5 [139]	2018[a]		• Incorporation of zero-knowledge procedures • Support: Asymmetric encryption Hash algorithms	• Refer to Table 8 for asymmetric encryption • Refer to Table 10 for hash algorithm
	ISO/IEC 9798-6 [140]	2021[a]		• Procedures employing manual data transfer • Support: Hash and MAC functions	• Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	NIST FIPS Pub. 196 [141]	1997	Entity Authentication Using Public Key Cryptography	• Withdrawn on October 2015	• Inactivated
	IETF RFC 1507 [142]	1993	DASS - Distributed Authentication Security Service	• Based on the use of digital signatures • Support: Asymmetric encryption and signature (RSA) Symmetric encryption (DES) Hash function (MD2)	• Refer to Table 8 and Table 12 for asymmetric encryption and signature, respectively • Refer to Table 6 for symmetric encryption • Refer to Table 10 for hash algorithm
	IETF RFC 1704 [143]	1994	Internet Authentication	• Provide guidance for selecting suitable authentication technologies based on specific use cases, accessibility, and network connectivity • Support: Asymmetric encryption (RSA) and signatures (RSA/DH) Symmetric encryption (DES) Hash function (MD5)	• Refer to Table 8 for asymmetric encryption • Refer to Table 12 for asymmetric digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 for hash algorithm
	IETF RFC 1760 [144]	1995	S/KEY: One-Time Password System for authentication	• Tailored for secure user authentication with remote servers • Support: Hash functions (MD4)	• Refer to Table 10 for hash algorithm
	IETF RFC 4120 [145]	2005	Kerberos Network Authentication Service (V5)	• Furnishes an outline and specifications for version 5, along with a comprehensive protocol description • Support: Symmetric encryption (DES-CBC, AES-CTS-128/256) Hash (MD5, SHA-1) and MAC (HMAC) functions	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively

[a] Standard was last reviewed and confirmed in the mentioned year.

sign authoritative zone data upon entry into the DNS and subsequently verify these data upon reaching its intended endpoint. Table 27 provides a comprehensive overview of the cryptographic standards used in the DNSSEC protocol, listing both classical and post-quantum recommendations based on the cryptographic support of each standard. We thoroughly analyze the security aspects of these standards and offer relevant recommendations to enhance their implementation.

The purpose of DNSSEC is to ensure the integrity and authenticity of DNS data. Digitally signing authoritative zone data prevents tampering and unauthorized modifications during transit. This authentication process mitigates the risk of DNS cache poisoning and man-in-the-middle attacks, preserving the accuracy and reliability of Internet traffic routing. Table 27 provides a comprehensive resource for understanding and selecting suitable DNSSEC cryptographic standards based on their intended purpose and cryptographic support.

Our analysis delves into the security aspects of DNSSEC cryptographic standards, which are mentioned in the notes column of the corresponding table. For example, this standard supports digital signatures and hash functions. Due to the use of specific cryptography, there is a chance that some algorithms are outdated or vulnerable. In this way, we identify potential vulnerabilities and suggest corresponding

Table 24
PKI standardization bodies and standards.

Objective	Standards	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
Specifications	ISO/IEC 9594-8 [146] (ITU-T X.509)	2020[b]	Framework for public-Key and attribute certificates	• Defines frameworks for PKI and PMI • Also defines directory schema information allowing PKI and PMI-related data to be stored in a directory • Support: Asymmetric digital signatures (i.e., RSA, DSA, ECDSA) Symmetric encryption (AES-128/192/256) Hash and MAC functions (SHA-1, SHA-2, SHA-3, HMAC)	• Refer to Table 12 for digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	ISO/IEC 15945 [147]	2018[a]	Certificate management (security techniques for defining TTP services that facilitate the implementation of digital signatures)	• ISO/IEC 15945 [147] has additionally been embraced in the capacity of an ITU-T Recommendation, designated as X.843 • Support: Asymmetric digital signature Hash functions	• Refer to Table 12 for digital signature • Refer to Table 10 for hash algorithms	
	ISO 21188 [148]	2018	Establishing a policy framework and practices for PKI in the financial services sector	• It revised the ISO 15782-1 & 15782-2 • Support: Asymmetric key exchange and digital signature Symmetric encryption MAC function	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 11 for MAC recommendations	
	ANSI X9.62 [149]	1998	PKC with ECDSA For The Financial Services Industry	• Defines a technique for generating and validating digital signatures • Support: Asymmetric digital signature (ECDSA) Hash functions (SHA-1, SHA-2)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for hash algorithms	
	ANSI X9.63 [150]	2001	Utilizing ECC in PKC for the Financial Services Sector: Facilitating Secure Key Agreements and Key Transports	• Defines Key Agreement and key transport mechanism • Support: Asymmetric key agreement/transport (using ECC) Hash and MAC functions (SHA-1, HMAC)	• Refer to Table 13 and Table 12 for asymmetric cryptography • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	PKCS #7 [151]	1998	ASN.1-driven approach to structuring cryptographic messages	• DER or PEM format • Support: Asymmetric encryption (RSA) Symmetric encryption (DES) Hash function (MD5)	• Refer to Table 8 for asymmetric encryption • Refer to Table 6 for symmetric encryption • Refer to Table 10 for hash algorithms	

"–" represents Not Applicable/Available.

[a] Standard was last reviewed and confirmed in the mentioned year.

[b] Standard was last approved/revised in the mentioned year.

recommendations, which are already mentioned in the previous tables. Thus, we refer to those existing tables in the recommendation field.

By offering recommendations, we ensure the effective and secure implementation of DNSSEC, keeping pace with best practices, emerging threats, and advancements in cryptographic techniques. With Table 27 as a reference, users and implementers can make informed decisions to protect against forged DNS data, ultimately improving the security of Internet users and applications.

10.3.3. SAML standards and recommendations

SAML represents the Security Assertion Markup Language, an open standard widely adopted that facilitates the interchange of authentication and authorization information among diverse entities, especially in web applications. With SAML, users only need to authenticate once with an Identity Provider (IdP) and can then access multiple services or applications without repeatedly entering their credentials. This feature is called Single-Sign-On (SSO) [213].

The OASIS (Organization for the Advancement of Structured Information Standards) consortium is responsible for developing and maintaining the SAML standard. Although OASIS is the primary organization for developing SAML standards, other groups, such as IETF, also contribute to advancing SAML-related technology. Tables 28 and 29 illustrate the SAML IETF and OASIS standards, respectively. We reference both classical and post-quantum recommendations based on the cryptographic support of the corresponding standards. We thoroughly analyze the security aspects of these standards and offer relevant recommendations to enhance their implementation.

Regarding post-quantum cryptography, the use of digital signatures for authentication and integrity verification in SAML is significantly impacted. To maintain long-term security for SAML-dependent systems, it is necessary to transition to post-quantum signature algorithms in SAML implementations.

Although there is no specific hybrid version of SAML, it uses cryptographic algorithms that can replace quantum-safe ones. SAML utilizes message digest algorithms such as SHA-1, SHA-256, or SHA-512 to create hash values for digital signatures and message integrity checks. Post-quantum recommendations for the Hash function can be found in Table 10. SAML supports various asymmetric key algorithms for digital signatures and exchange, including RSA and DSA. Table 8 presents post-quantum recommendations for asymmetric cryptography. SAML utilizes key transport algorithms to securely transmit symmetric keys for message encryption, typically involving asymmetric encryption algorithms such as RSA or Diffie–Hellman key exchange. Table 13 provides post-quantum recommendations for key exchange.

10.3.4. Oauth standards and recommendations

OAuth (Open Authorization) is a protocol that helps protect the security of user or application resources. It allows users to give limited access to their data or functionality to other applications without sharing their login information [214]. To ensure compatibility and consistency across different systems, OAuth has been standardized through efforts of the IETF, which has defined various RFCs for OAuth standardization. The standardization of OAuth by IETF is shown in Table 30, listing both classical and post-quantum recommendations based on the cryptographic support of each standard. We thoroughly analyze the security aspects of these standards and offer relevant recommendations to enhance their implementation.

Table 25
IETF standards for cryptography in Public Key Infrastructure (PKI).

Algorithm	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
RSA	4055 [152]	2005	Enhancing RSA cryptography with supplementary algorithms and identifiers for integration within the PKIX profile	• Updates RFC 3279 and Updated by RFC 5756 • Support: Asymmetric digital signature (RSASSA-PSS) and key transport (RSAES-OAEP) Hash functions (SHA-1, SHA-2)	• Refer to Table 12 and Table 13 for asymmetric digital signature and key transport, respectively • Refer to Table 10 for hash algorithms
	5756 [86]	2010	Enhancements for RSAES-OAEP and RSASSA-PSS algorithm configuration	• Updates the standards for algorithm parameters within the subjectPublicKeyInfo field of an X.509 certificate • Support: Asymmetric digital signature (RSASSA-PSS) and key transport (RSAES-OAEP)	• Refer to Table 12 and Table 13 for asymmetric digital signature and key transport, respectively
	8017 [54]	2016	RSA Cryptography Specifications	• Recommendations including cryptographic primitives, encryption schemes, signature schemes, etc. • Support: Asymmetric encryption (RSA, RSAES-OAEP) and digital signature (RSASSA-PSS, RSASSA-PKCS1-v1_5) Hash functions (MD5, SHA-1, SHA-2)	• Refer to Table 8 and Table 12 for asymmetric encryption and digital signature, respectively • Refer to Table 10 for hash algorithms
	8018 [153]	2017	Password-Based Cryptography Specification	• Recommendations encompass essential key derivation functions, encryption methodologies, message authentication protocols, etc. • Support: Asymmetric encryption (RSA) Symmetric encryptions (DES/3DES-CBC, AES-CBC-128/192/256 RSA) Hash and MAC functions (MD5, SHA-1, SHA-2, HMAC)	• Refer to Table 8 for asymmetric encryption • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
ECC	5753 [154]	2010	Use of ECC Algorithms in CMS	• Obsoletes RFC 3278 • Support: Asymmetric cryptography (ECDSA, DH, RSA) Symmetric encryption (3DES-CBC, AES-CBC-128/192/512) Hash and MAC functions (SHA-1, SHA-2, HMAC)	• Refer to Table 13 and Table 12 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	5480 [155]	2009	Defines the syntax and semantics for the Subject Public Key Information field in certificates that enable ECC support	• Updates RFC 3279 and Updated by RFC 8813 • Support: Asymmetric cryptography (ECDSA, ECDH, ECMQV) Hash functions (SHA-1, SHA-2)	• Refer to Table 13 and Table 12 for asymmetric cryptography • Refer to Table 10 for hash algorithms

(*continued on next page*)

At present, the OAuth protocol does not have any quantum-safe cryptography measures. Despite not having a hybrid option, the protocol uses cryptographic algorithms that can potentially be replaced by quantum-safe alternatives. Like other cryptographic protocols, OAuth uses cryptographic algorithms such as symmetric encryption, asymmetric encryption, and digital signatures. Tables B.37, 8, and 12, containing post-quantum recommendations for symmetric/asymmetric cryptography and digital signature recommendations, respectively. OAuth 2.0 is the most commonly used version of OAuth. It employs secure communication protocols such as TLS [215] to ensure the confidentiality and integrity of data transmission. Table 31 presents a hybrid version of TLS that can be used in place of the classical version to improve safety in the context of post-quantum security.

11. Standards and recommendations for communication protocols

This section analyzes the cryptographic standards used in widely adopted network communication and security protocols such as TLS, IPsec, SSH, FTP, and S/MIME. Our goals are to (1) evaluate the cryptographic mechanisms and algorithms currently employed and (2) provide security recommendations addressing both classical and post-quantum threats. By assessing protocols current security posture and considering the potential impact of quantum computing, our goal is to provide actionable guidance to strengthen their resilience and ensure the continued protection of sensitive information.

We present both classical and post-quantum recommendations for these protocols (Tables 31–36), based on their cryptographic support as defined in Section 1.4. The analysis covers security aspects of the standards and offers recommendations to improve their implementation. For detailed guidelines on table columns, cryptographic recommendations, and evaluation criteria, see Sections 1.3 and 1.4.

11.1. TLS standards and recommendations

Table 31 presents a comprehensive overview of Transport Layer Security (TLS) cryptographic standards, which are widely used to ensure secure communication across networks. As part of our examination

Table 25 (*continued*).

Algorithm	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
Multiple Algorithms[a]	8692 [156]	2019	Enhancing Algorithm Identifiers for RSASSA-PSS and ECDSA utilizing SHAKEs for Internet X.509 PKI	• Describes the guidelines governing the utilization of the SHAKE function family • Support: Asymmetric digital signature (RSASSA-PSS, ECDSA) Hash functions (SHA-3 (SHAKE-128/256))	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for hash algorithms
	5758 [157]	2010	Enhanced Algorithms and Identifiers for DSA and ECDSA within the Internet X.509 PKI	• Define algorithm identifiers and ASN.1 encoding rules • Support: Asymmetric digital signature (DSA, ECDSA) Hash functions (SHA-2)	
	3279 [158]	2002	Specifies algorithm identifiers and ASN.1 encoding formats for digital signatures and subject public keys in the Internet X.509 PKI	• Algorithms and Identifiers for the PKIX profile • Updated by RFC 4055 [152], RFC 4491 [159], RFC 5480 [155], RFC 5758 [157], RFC 8692 [156] • Support: Asymmetric digital signature (RSA, DSA, ECDSA) and key exchange (DH) Hash functions (MD5, SHA-1)	• Refer to Table 12 and Table 13 for asymmetric digital signature and key exchange, respectively • Refer to Table 10 for hash algorithms
	6955 [160]	2013	Crafted with the intent of offering a Proof-of-Possession for the private key rather than serving as a general-purpose signing algorithm	• Obsoletes RFC 2875 and outlines a pair of techniques for generating an integrity check value from a DH key pair, along with a method for deriving an integrity check value from an EC key pair • Support: Asymmetric digital signature (ECDSA) and key agreement (DH, ECDH) Hash and MAC functions (SHA-1, SHA-2, HMAC)	• Refer to Table 12 and Table 13 for asymmetric digital signature and key agreement, respectively • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
MAC	9045 [161]	2021	Revisions have been made to the cryptographic algorithm prerequisites for the Password-Based MAC in the Internet X.509 PKI CRMF	• PKI CRMF specified in RFC 4211 • Support: Symmetric encryption (DES/3DES, AES-GMAC-128) Hash and MAC functions (SHA-1, SHA-256, HMAC, GMAC)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
HASH	5754 [162]	2010	Outlines the guidelines for utilizing SHA-2 Algorithms within CMS	• Provides 'SMIMECapabilities' attribute values for each algorithm • Support: Asymmetric digital signature (RSA, DSA, ECDSA) Hash and MAC functions (SHA-2, HMAC)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	8702 [163]	2020	Outlines the guidelines for incorporating the SHAKE family of hash functions into the CMS	Hash functions used with the RSASSA-PSS and ECDSA • Support: Asymmetric digital signature (RSASSA-PSS, ECDSA) Hash and MAC functions (SHA-3 (SHAKE-128/256), KMAC)	

Note: Abbreviations are defined as follows: MS: Modulus Size, KS: Key Size, HL: Hash Length.

[a] It comprises a set of security and cryptography standards that employ a combination of algorithms.

of the cryptographic standards of TLS, we analyze its security aspects and provide relevant recommendations. Standards that share the same cryptographic recommendations are consolidated and categorized according to the cryptographic mechanisms they employ. Additionally, the table includes the hybrid draft standard of TLS, which enhances security in the post-quantum cryptography era by employing multiple key exchange algorithms simultaneously.

11.2. Ipsec standards and recommendations

Tables 32 and 33 provide a comprehensive overview of the IPsec/IKE cryptographic standards, which are widely utilized for secure communication purposes. IPsec/IKE is commonly employed to establish secure Virtual Private Networks (VPNs) and protect data transmission over the Internet. The table offers valuable information such as the year of publication, intended purpose, specific notes, and cryptographic support for each standard. Our exploration of these standards includes a thorough analysis of their security aspects, and we provide relevant recommendations based on our findings. To simplify the presentation, we have grouped standards with similar cryptographic recommendations based on the specific mechanisms they employ. Additionally, the table includes the IKE standard for post-quantum security, which incorporates Mixing Pre-shared Keys to provide resistance against quantum computers.

11.3. SSH standards and recommendations

Table 34 presents a comprehensive overview of Secure Shell (SSH) cryptographic standards, a widely used network protocol that provides secure remote login, file transfer, and command execution between network devices. As we explore SSH cryptographic standards, we extensively analyze their security aspects and provide relevant

Table 26
IETF cryptographic standards/drafts for well-known Kerberos protocols.

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
Kerberos	2712 [164]	1999	Propose new cipher suites in TLS for Kerberos-based authentication	• Enabling mutual authentication and secure communication • Support: Asymmetric Cryptography for Authentication and key exchange (RSA, DH) Symmetric encryption (DES, 3DES) Hash and MAC functions (MD5, SHA) TLS	• Refer to Table 13 and Table 23 for asymmetric Key exchange and authentication, respectively • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 31 for TLS	
	3962 [165]	2005	Specify the integration AES algorithm into the Kerberos 5 cryptosystem suite	• Support: Symmetric encryption (AES-CTS-128/256) Hash and MAC functions (SHA-1, HMAC) Authentication	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 23 for Authentication	
	4120 [145]	2005	The Kerberos Network Authentication Service (V5)	• Support: Symmetric encryption (DES, AES-CTS-128/256) Hash and MAC functions (MD5, SHA-1, HMAC) Authentication		
	8009 [166]	2016	Specify encryption and checksum types for Kerberos 5 using AES in CTS mode and HMAC with SHA-2	• Support: Symmetric encryption (AES-CTS-128/256) Hash and MAC functions (SHA-2-256/384, HMAC) Authentication		
	5349 [167]	2008	ECC Support for PKC for Initial Authentication in Kerberos (PKINIT)	• Describes the use of ECC in PKINIT for secure public key-based authentication and key exchange • Support: Asymmetric digital signature and key exchange (ECDSA) Hash functions (SHA-1, SHA-256/384/512) Authentication and PKI	• Refer to Table 12 and Table 13 for asymmetric digital signature and Key exchange, respectively • Refer to Table 10 for Hash recommendations • Refer to Table 23 for authentication • Refer to Table 24 for PKI recommendations	
	6251 [168]	2011	Specify the integration of the Kerberos V5 protocol with TLS	• Defines Kerberos V5 protocol integration with TLS to enhance security • Support: Authentication, PKI, and TLS	• Refer to Table 23 for Authentication • Table 24 for PKI recommendations • Refer to Table 31 for TLS	
	6803 [169]	2012	Specify camellia encryption for Kerberos 5	• Defines encryption and checksum types using Camellia block cipher and CMAC algorithm • Support: Symmetric encryption (Camellia-CTS-128/256) Hash and MAC (SHA-1, HMAC, CMAC)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	7751 [170]	2016	Introduce a Kerberos authorization data container with multiple MACs to enhance authentication	• Supporting multiple MACs • Support: MAC and Authentication	• Refer to Table 11 for MAC recommendations • Refer to Table 23 for Authentication	
	8636 [171]	2019	PKC for Initial Authentication in Kerberos (PKINIT) Algorithm Agility	• Enhances PKINIT standard by removing algorithm dependencies, enabling flexibility, and future-proofing against vulnerabilities • Support: Asymmetric key exchange (DH) Symmetric encryption (AES-CTS-256) Hash and MAC functions (SHA-1, SHA-256/384/512) Authentication and PKI	• Refer to Table 13 for asymmetric Key exchange • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 23 for authentication • Refer to Table 24 for PKI recommendations	

recommendations to ensure the confidentiality and integrity of data transmission. We categorize standards that share the same cryptographic recommendations, grouping them according to the specific cryptographic mechanisms they use. Additionally, the table includes the hybrid draft standard of SSH, which combines classical ECDH key exchange with PQ KEM to provide PQ hybrid key exchange methods. This comprehensive overview of SSH cryptographic standards aims to help researchers, developers, and security practitioners in understanding and implementing robust security measures in SSH communications.

Table 27
IETF cryptographic standards/drafts for well-known DNSSEC protocols.

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
DNSSEC	4033 [172]	2005	DNS Security Introduction and Requirements	• Ensures data source authentication and content integrity within the Domain Name System • Support: Digital signature Hash functions Authentication	• Refer to Table 12 for digital signature • Refer to Table 10 for Hash recommendations • Refer to Table 23 for authentication	
	4398 [173]	2006	Storing cryptographic certificates and revocation lists in DNS	• CERT resource records enable authentication of cryptographic public keys in DNS • Support: Asymmetric digital signatures Hash functions PKI	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations • Refer to Table 24 for PKI recommendations	
	5155 [174]	2008	DNSSEC Hashed Authenticated Denial of Existence	• NSEC3 resource record in DNSSEC enhances denial of existence authentication and safeguards against zone enumeration and facilitates expansion of delegation-centric zones • Support: Asymmetric digital signatures (DSA, RSA) Hash functions (SHA-1)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations	
	6781 [175]	2012	Provide operational guidelines for zone administrators deploying DNSSEC	• Covering key management, signature generation, and related policies • Support: Asymmetric digital signatures and key exchange (RSA) Hash functions (MD5, SHA-1, SHA-256) Authentication and TLS	• Refer to Table 12 and Table 13 for asymmetric digital signature and Key exchange • Refer to Table 10 for Hash recommendations • Refer to Table 23 for Authentication • Refer to Table 31 for TLS	
	7091 [176]	2013	Digital Signature Algorithm is specified by GOST R 34.10-2012	• Offers comprehensive guidance on the generation and authentication of digital signatures using GOST R 34.10-2012 • Support: Digital signature (GOST R 34.10-2012) Hash functions	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations	
	8080 [177]	2017	Provides guidelines for specifying EdDSA keys and signatures in DNSSEC	• Focuses on the utilization of EdDSA with the two specified curves (Ed25519 and Ed448) for DNSSEC implementation • Support: Digital signature (EdDSA (Ed25519 and Ed448))	• Refer to Table 12 for digital signature	

(*continued on next page*)

11.4. FTP standards and recommendations

Table 35 presents a comprehensive overview of three widely used protocols for secure file transfer: FTP, FTPS, and SFTP. FTP, which stands for File Transfer Protocol, is a conventional network protocol used to facilitate the transfer of files between a computer network's client and server. Its purpose is to facilitate the efficient exchange of files, allowing users to upload, download, and manage files remotely. On the other hand, FTPS, or FTP Secure, is an extension of FTP that adds an extra layer of security through the use of SSL/TLS encryption. This protocol enhances the security of file transfers by encrypting the data exchanged between the client and the server, ensuring that sensitive information remains protected from unauthorized access.

SFTP, which stands for Secure File Transfer Protocol, is an entirely different protocol from FTP and FTPS. Unlike its counterparts, SFTP is not an extension, but rather a standalone protocol that operates over SSH (Secure Shell) connections. SFTP provides secure file transfer capabilities combined with the strong authentication and encryption features of SSH, making it an excellent choice for secure file transfers in a wide range of environments.

In Table 35, we thoroughly examine the security aspects of the FTP cryptographic standards to identify their strengths and vulnerabilities, allowing us to provide relevant and actionable recommendations. To enhance readability and comprehension, we have consolidated the standards based on the specific cryptographic recommendations they employ, allowing for a systematic evaluation process.

11.5. S/MIME standards and recommendations

S/MIME, which stands for Secure/Multipurpose Internet Mail Extensions, is a commonly used standard that guarantees safety in email communication. It involves a variety of protocols and formats that allow encryption, signing, and authentication of the email message. The S/MIME protocol is built on the Cryptographic Message Syntax (CMS) created by the IETF. It uses asymmetric encryption, digital signatures, and certificates to secure the privacy, accuracy, and legitimacy of email content [278].

S/MIME is not directly standardized by ISO and ANSI. The responsibility for standardizing S/MIME lies with the IETF, which achieves this through the creation of RFCs. These RFCs outline core specifications, including encryption, digital signatures, and certificate handling. They establish standards and protocols for the implementation of S/MIME, serving as authoritative guides for its implementation and

Table 27 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
	6605 [178]	2012	Elaborates on the technique of defining ECDSA keys and signatures in DNS Security (DNSSEC) using SHA-2 hashes	• Focuses on the utilization of EdDSA with the two specified curves (Ed25519 and Ed448) for DNSSEC implementation • Support: Asymmetric digital signature (ECDSA (P-256, P-384) Hash Functions (SHA-256/384)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations
	5933 [179]	2010	Utilizing GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC	• Outlines the process of creating digital signatures and hash functions using GOST algorithms • Support: Digital signature (ECC-GOST (GOST R 34.10-2001)) Hash functions (GOST R 34.11-94)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations
	5702 [180]	2009	Incorporating SHA-2 algorithms in conjunction with RSA within DNSKEY and RRSIG resource records for DNSSEC	• Provides instructions for generating DNSKEY and RRSIG resource records for DNS Security Extensions • Support: Asymmetric digital signature (RSA) Hash functions (SHA-256/512)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations
	3110 [181]	2001	RSA/SHA-1 signatures and RSA keys used in DNS	• Incorporates advancements in hashing for DNS signatures and deprecates the weaker mechanism • Support: Asymmetric digital signature (RSA) Hash functions (SHA-1)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations
	draft-makarenko-gost2012-dnssec-05 [182]	2023	Provide guidelines for generating digital signatures and hash functions via GOST R 34.10-2012 and GOST R 34.11-2012 algorithms within DNSSEC	• Focuses on the usage of these algorithms for DNSKEY, RRSIG, and DS resource records in DNSSEC. Expired in July 2024 • Support: Asymmetric digital signature (GOST R 34.10-2012) Hash and MAC functions (GOST R 34.11-2012, HMAC)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	draft-dnsop-dnssec-extension-pkix-01 [183]	2023	Mechanism to enhance DNSSEC protocol, ensuring the integrity of DNS messages using PKIX certificates independently	• Protocol is extended to address vulnerabilities in DNS messages, providing integrity assurance. Expired in September 2023 • Support: Digital signature PKI	• Refer to Table 12 for digital signature • Refer to Table 24 for PKI recommendations
	draft-cuiling-dnsop-sm2-alg-05 [184]	2023	Provide guidelines for specifying SM2 Digital Signature Algorithm keys and signatures in DNSSEC	• SM3 is used as the hash algorithm for signatures in DNSSEC. Expired in September 2023 • Support: Digital signature (SM2) Hash functions (SM3)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations.

understanding. The S/MIME standardization RFCs are listed in Table 36.

The security of S/MIME relies on MIME and CMS. CMS is a syntax that encapsulates data for protection, and its parameters influence many of S/MIME's security properties. Fortunately, the CMS parameters are customizable, including the choice of the algorithm used. This flexibility facilitates the transition of protocols such as TLS, SSL, and RSA to quantum-safe cryptography [297].

When it comes to S/MIME, post-quantum cryptography has an impact on encryption and digital signatures. This means that post-quantum encryption algorithms can replace current ones, such as RSA or ECC. Typically, S/MIME relies on the widely-used RSA encryption algorithm. However, to ensure secure encryption against quantum computers, it should be replaced with quantum-safe algorithms in a hybrid approach [298].

By using a combination of classical cryptographic algorithms and post-quantum algorithms, a post-quantum hybrid S/MIME system can offer protection against both classical and quantum-based attacks. The recommendations for using hybrid TLS/SSL in S/MIME are presented in Table 31, and in Table 12, there are hybrid recommendations for using digital signatures in S/MIME.

11.6. Summary

This section offers a comprehensive overview of cryptographic standards for different protocols and provides cryptographic recommendations based on the supported cryptographic mechanisms. Its primary goal is to assist researchers, developers, and security practitioners in gaining a clear understanding of these standards and in effectively implementing robust security measures. By providing detailed information and insights, the section equips its intended audience with the knowledge necessary to navigate the complexities of cryptographic protocols and ensure the implementation of robust security practices.

Table 28
IETF cryptographic standards/drafts for SAML.

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
SAML	8409 [185]	2019	The Entity Category (SAML) Attribute Types	• Describes two SAML entity attributes: one to indicate an entity's membership in a category, and another to signal that an entity supports or interoperates with entities in such categories. • Support:Digital signature	• Refer to Table 12 for digital signatures	
	7833 [186]	2016	A RADIUS Attribute, Binding, Profiles, Name Identifier Format, and Confirmation Methods for the (SAML)	• Describes the use of the SAML with RADIUS in the context of the Application Bridging for Federated Access Beyond web (ABFAB) architecture • Support: TLS	• Refer to Table 31 for TLS	
	7522 [187]	2015	SAML 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants	• Defines the use of a SAML 2.0 Bearer Assertion as a means for requesting an OAuth 2.0 access token as well as for client authentication • Support: Asymmetric digital signature (RSA), Hash function (SHA-256), X.509	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations • Refer to Table 31 for TLS • Refer to Table 24 for PKI recommendations	
	6595 [188]	2015	A Simple Authentication and Security Layer (SASL) and GSS-API Mechanism for the SAML	• Specifies a SASL mechanism and a GSS-API mechanism for SAML 2.0 that allows the integration of existing SAML. Identity Providers with applications using SASL and GSS-API • Support: TLS, X.509	• Refer to Table 31 for TLS • Refer to Table 24 for PKI recommendations	

Table 29
OASIS cryptographic standards/drafts for SAML.

Protocol	Standard	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
SAML	OASIS	2007	Metadata Profile for the OASIS Security Assertion Markup Language (SAML) V1.x [189]	• Defines a profile of the OASIS SAML V2.0 metadata specification for use in describing SAML V1.0 and V1.1 entities. • Support: Hash function	• Refer to Table 10 for Hash	
		2005	Assertions and Protocols for the OASIS SAML V2.0 [190]	• Defines the syntax and semantics for XML-encoded assertions about authentication, attributes, and authorization • Support: TLS/SSL, Asymmetric, Hash function	• Refer to Table 31 for TLS/SSL • Refer to Table 8 for asymmetric cryptography • Refer to Table 10 for Hash recommendations	
		2011	SAML v2.0 Metadata Profile for Algorithm Support Version 1.0 [191]	• Defines metadata extension elements to enable entities to describe the XML Signature [XMLSig] algorithms they support • Support: Symmetric cryptography, Asymmetric cryptography, HASH function	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations	
		2010	SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0 [192]	• The SAML V2.0 Holder-of-Key Web Browser SSO Profile allows for transport of holder-of-key assertions by standard HTTP user agents with no modification of client software and maximum compatibility with existing deployments • Support: TLS, SSL, X.509	• Refer to Table 31 for TLS/SSL • Refer to Table 24 for PKI recommendations	

12. Discussion

Post-Quantum Cryptography (PQC) is designed to be secure against both classical and quantum computers. PQC algorithms rely on mathematical problems that are considered challenging for quantum computers to solve. Efforts to develop PQC standards are ongoing and several organizations are actively involved in this area. For example, NIST has launched a competition to develop PQC standards. The competition is now in its final stage, and several post-quantum cryptography algorithms have been selected as finalists and standardized, as discussed in Section 1.2. However, PQC is not the only solution to the quantum threat. Classical cryptography can also be made more secure against quantum attacks by using larger key sizes or different algorithms. Many existing cryptographic standards, such as AES and SHA-3, have been analyzed to determine their resistance to quantum attacks, and recommendations have been made to increase their security. Therefore, in this work, we have presented both classical and post-quantum recommendations for block ciphers, hash and MAC functions, key establishment, digital signatures, some widely used cryptographic protocols, and so on. These recommendations can be used to protect sensitive information in the post-quantum era. The main goal of these recommendations is to ensure the ongoing security of cryptographic standards in response to advancements in quantum computing.

Table 30
IETF cryptographic standards for OAuth.

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
OAuth	9201 [193]	2022	Introducing Additional OAuth Parameters for Authentication and Authorization in Constrained Environments (ACE)	Introduces novel parameters and encodings intended to enhance the OAuth 2.0 token and introspection endpoints • Support: Symmetric key	• Refer to Table 6 for symmetric encryption	
	8705 [194]	2020	OAuth 2.0 Protocol for Mutual-TLS Client Authentication and Certificate-Bound Access Tokens	• Support: Asymmetric key (RSA), Hash function (SHA-256), TLS	• Refer to Table 8 and Table 13 for asymmetric cryptography • Refer to Table 10 for Hash recommendations • Refer to Table 31 for TLS	
	7662 [195]	2015	OAuth 2.0 Token Introspection	• Support: TLS, SSL	• Refer to Table 31 for TLS/SSL	
	7523 [196]	2015	Profile of JSON Web Token (JWT) usage for OAuth 2.0 Client Authentication and Authorization Grants	• Support: Hash function, TLS	• Refer to Table 10 for Hash recommendations • Refer to Table 31 for TLS	
	7628 [197]	2015	A collection of Simple Authentication and Security Layer (SASL) Mechanisms devised for integration with OAuth			
	7636 [198]	2015	Utilization of Proof Key for Code Exchange by Public Clients under OAuth			
	7521 [199]	2015	Framework for Assertion within OAuth 2.0 Client Authentication and Authorization Grants	• Support: Asymmetric digital signature (ECDSA P-256), Hash and MAC functions (SHA-256, MAC)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	7522 [187]	2015	Exploration of SAML 2.0's application as a profile for OAuth 2.0 Client Authentication and Authorization Grants	• Support: Asymmetric digital signature (RSA), Hash and MAC functions (SHA-256, MAC), TLS	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 31 for TLS	
	9101 [200]	2021	Enhancing OAuth 2.0 by securing authorization requests with JWT	• Support: Asymmetric encryption (DSA, RSA, GOST) and signature (ECDSA P-256), symmetric encryption (DES/3DES/GOST), Hash functions (MD5, SHA-1, GOST)	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 for hash algorithms	
	9200 [201]	2022	Authentication and Authorization within Constrained Environments leveraging the OAuth 2.0 Framework (ACE-OAuth)	• Support: Asymmetric digital signature, Hash and MAC functions, TLS	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 31 for TLS	
	6819 [202]	2013	Threat Model and Security Considerations pertaining to OAuth 2.0	• Support: Asymmetric Cryptography, Symmetric Keys, Hash and MAC functions, TLS	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 6 for symmetric encryption	
	7591 [203]	2015	Dynamic Client Registration Protocol for OAuth 2.0	• Support: Asymmetric digital signature (RS256), MAC function, TLS	• Refer to Table 12 for asymmetric digital signature • Refer to Table 11 for MAC recommendations • Refer to Table 31 for TLS	
	6750 [204]	2012	The OAuth 2.0 Authorization Framework: Bearer Token Usage	• Support: Asymmetric digital signature, MAC function, TLS		
	9068 [205]	2021	Profile of JWT meant for OAuth 2.0 Access Tokens	• Support: Asymmetric digital signature,	• Refer to Table 8 and Table 12 for asymmetric digital signature	
	6749 [206]	2012	The OAuth 2.0 Authorization Framework	• Support: MAC functions, TLS	• Refer to Table 11 for MAC recommendations • Refer to Table 31 for TLS	
	7009 [207]	2013	OAuth 2.0 Token Revocation	• Support: TLS	• Refer to Table 31 for TLS	
	7592 [208]	2015	Management Protocol for Dynamic Client Registration within OAuth 2.0			
	8252 [209]	2017	OAuth 2.0 Application Pattern for Native Apps			
	8414 [210]	2018	Metadata for OAuth 2.0 Authorization Servers			
	8628 [211]	2019	OAuth 2.0 Device Authorization Grant			
	8693 [212]	2020	OAuth 2.0 Token Exchange			

For instance, we have provided recommendations on the appropriate key and hash sizes for symmetric cryptography to maintain a quantum-resistant environment. We also recommend considering available post-quantum mechanisms for asymmetric cryptography. It is important to note that the development of post-quantum cryptography standards and the evaluation of classical standards against quantum

Table 31
IETF cryptographic standards/drafts for well-known network protocols (TLS/SSL).

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
TLS/SSL	5288 [216]	2008	AES-GCM Cipher Suites for TLS provide confidentiality and data origin authentication.	• Support: Asymmetric key exchange (RSA, DSA, DH) Symmetric encryption (AES-GCM (128/256)) Hash functions (SHA-2-256/384)	• Refer to Table 13 for asymmetric key exchange • Refer to Table 6 and Table B.37 for symmetric encryption and recommendations on classical and PQ modes • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	5487 [217]	2009	Enhancing security with TLS Pre-Shared Key (PSK) cipher suites featuring SHA-256/384 and AES GCM	• Support: Asymmetric key exchange (PSK, DHE_PSK, and RSA_PSK) Symmetric encryption (AES-GCM (128/256)) Hash and MAC functions (SHA-256/384, HMAC)	• Refer to Table 13 for asymmetric key exchange • Refer to Table 6 and Table B.37 for symmetric encryption and recommendations on classical and PQ modes • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	5489 [218]	2009	ECDHE_PSK Cipher Suites for TLS	• Support: Asymmetric key exchange (ECDHE_PSK) Symmetric encryption (3DES, AES-CBC (128/256)) Hash and MAC functions (SHA-1, SHA-256/384, HMAC)		
	5932 [219]	2010	Specify a collection of cipher suites for TLS that support the Camellia encryption algorithm	• Support: Asymmetric key exchange (RSA, DHE_RSA, DH_RSA, DHE_DSS, DH_DSS) Symmetric encryption (Camellia-CBC (128/256)) Hash and MAC functions (SHA-1 , SHA-256, HMAC)		
	6655 [220]	2012	AES-CCM Cipher Suites for TLS	• Support: Asymmetric key exchange (PSK, RSA, DHE_RSA) Symmetric encryption (AES-CCM (128/256)) Hash and MAC functions (SHA-256, HMAC)		
	7251 [221]	2014	Illustrating AES-CCM integration in TLS and define cipher suites using ECC	• Support: Asymmetric key exchange (ECDHE_ECDSA) Symmetric encryption (AES-CCM (128/256)) Hash and MAC functions (SHA-256/384/512, HMAC)		
	8446 [222]	2018	Specifies TLS Protocol Version 1.3	• Support: Asymmetric key exchange (PSK, RSA, DH, ECDHE (secp256r1, X25519)) and digital signature (RSASSA-PSS, RSASSA-PKCS1-v1_5, ECDSA) Symmetric encryption (AES-CCM/GCM (128/256)) Hash and MAC functions (SHA-1, SHA-256/384/512, HMAC)	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature, respectively • Refer to Table 6 and Table B.37 for symmetric encryption and recommendations on classical and PQ modes • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	

(*continued on next page*)

attacks is an ongoing process. As quantum computing technology continues to evolve, new attacks may emerge, requiring the development of new defenses. Therefore, ongoing research and development in quantum-resistant cryptography is crucial to safeguard sensitive information in the post-quantum era. To address this challenge, we suggest adopting crypto-agile practices. Crypto-agility [299] refers to the ability of a cryptographic system to switch quickly and seamlessly between different cryptographic algorithms, protocols, and standards as they evolve. This adaptability is essential because, as quantum computing progresses, algorithms that are currently considered secure may become vulnerable to quantum-based attacks, while new quantum-resistant algorithms will emerge. By designing systems to be crypto-agile, organizations can ensure they are prepared to integrate new algorithms without overhauling their entire infrastructure. This forward-thinking approach future-proofs security systems, allowing them to remain resilient in the face of evolving threats.

As a concrete illustration of crypto-agility, Section 3.3.3 presents a phased migration for financial-grade TLS from RSA-2048 key exchange to a hybrid RSA+ Kyber1024 handshake. The plan preserves backward compatibility, enables controlled canary rollouts, and limits operational change to load-balancer configuration, HSM policy, and certificate workflows rather than a complete redesign PKI redesign. The measured latency and bandwidth impacts remain within the online banking budgets, and Kyber's compact artifacts fit existing X.509 v3 structures and

Table 31 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
	5289 [223]	2008	Introducing cipher suites based on ECC featuring SHA-256/384 and AES-GCM	• Support: Asymmetric key exchange (ECDHE) and digital signature (ECDSA, RSA) Symmetric encryption (AES-GCM/CBC-128/256) Hash and MAC functions (SHA-256/384, HMAC)	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	6101 [224]	2011	Specifies the SSL 3.0 protocol	• Support: Asymmetric key exchange (RSA, DH) and digital signature (RSASSA-PKCS1, DSS) Symmetric encryption (DES, 3DES) Hash and MAC functions (SHA, MD5, MAC)	
	9325 [225]	2022	Offering best practices for the secure use of TLS/DTLS	• Support: Asymmetric key exchange (ECDHE) and digital signature (ECDSA, RSA) Symmetric encryption (AES-GCM-128/256) Hash and MAC functions (SHA-256/384, HMAC)	
	8422 [226]	2018	Describe key exchange algorithms based on ECC for the TLS	• Specifies the use of ECDHE key agreement in a TLS handshake • Support: Asymmetric key exchange (ECDHE) and digital signature (ECDSA, EdDSA, RSA) Symmetric encryption (AES-CBC/GCM-128/256) Hash functions (SHA-256/384)	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature • Refer to Table 6 and Table B.37 for symmetric encryption and recommendations on classical and PQ modes • Refer to Table 10 for Hash recommendations
	9367 [227]	2023	GOST Cipher Suites for TLS 1.3	• Specifies the utilization of GOST algorithms to establish cipher suites, signature schemes, and key exchange methods • Support: Asymmetric key exchange (ECDHE, PSK) and digital signature (GOST R 34.10-2012) Symmetric encryption (GOST R 34.12-2015) Hash and MAC functions (GOST R 34.11-2012, HMAC)	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	8734 [228]	2020	ECC Brainpool Curves for TLSv1.3	• Incorporates essential protocol mechanisms to enable the utilization of ECC Brainpool curves within TLS 1.3 • Support: Asymmetric key exchange (ECDHE) and digital signature (ECDSA) Hash functions (SHA-256/384/512)	• Refer to Table 13 and Table 12 for asymmetric key exchange and digital signature • Refer to Table 10 for Hash recommendations
	7027 [229]	2013	Specifies ECC Brainpool Curves for TLS	• Provides authentication and key exchange • Support: Asymmetric cryptography for key exchange (ECC Brainpool curves)	• Refer to Table 13 for asymmetric key exchange
	6091 [230]	2011	Define TLS authentication use	• Also establishes a registry for non-X.509 certificate types • Support: Asymmetric key exchange (DHE_DSS, DHE_RSA)	• Refer to Table 13 for asymmetric key exchange
	7366 [231]	2014	Propose a method to facilitate the negotiation of the encrypt-then-MAC security mechanism within TLS/DTLS	• Addressing the security vulnerabilities associated with the MAC-then-encrypt • Support: Symmetric encryption MAC function (HMAC)	• Refer to Table 6 for symmetric encryption • Refer to Table 11 for MAC recommendations

(*continued on next page*)

certificate-transparency processes. This hybrid posture provides an immediate rollback path as standards evolve and helps satisfy regulatory expectations for crypto-agility in payment environments.

In summary, the threat quantum computing poses to cryptography is a significant challenge that requires continuous attention and research. PQC and classical standards resistant to quantum attacks are being

Table 31 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
	9147 [232]	2022	Specify version 1.3 of the DTLS protocol	• Enabling secure communication over the Internet while preventing eavesdropping, tampering, and message forgery • Support: Symmetric encryption (AES-GCM/CCM-128/256) Hash-function (SHA-256)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations	
	7905 [43]	2016	ChaCha20-Poly1305 Cipher Suites for TLS	• Outline the utilization of the ChaCha stream cipher and Poly1305 authenticator in the TLS and DTLS protocols • Support: Asymmetric key exchange (DHE_RSA, ECDHE_RSA, ECDHE_ECDSA) Symmetric encryption (CHACHA20) Hash functions (SHA-256, POLY1305)	• Refer to Table 8 and Table 13 for asymmetric encryption and key exchange, respectively • Refer to Table 10 for Hash recommendations	
TLS-Hybrid	draft-ietf-tls-hybrid-design-10 [113]	2023	Introduce hybrid key exchange in TLS 1.3	• Utilizing multiple key exchange algorithms simultaneously for enhanced security in the post-quantum cryptography era (Draft is Expired in October 2024) • Support: Asymmetric key exchange (ECDHE (x25519, secp256r1) with kyber (512, 768) Hash functions (SHA-3)	• Refer to Table 13 for asymmetric key exchange • Refer to Table 10 for Hash recommendations	

developed, but the field is constantly evolving. Organizations must stay up-to-date with the latest advancements in quantum-resistant cryptography to ensure the security of their digital communications and data. Furthermore, we recommend implementing "crypto-agile" practices for existing cryptographic systems. This involves designing systems that can easily adapt to new cryptographic standards, recommendations, and algorithms as they are developed, ensuring ongoing security in the era of quantum computing.

13. Conclusion

The advent of quantum computing poses a significant threat to the security of traditional cryptographic algorithms, necessitating an urgent shift toward a quantum-resistant environment. In this paper, we provide a comprehensive evaluation of cryptographic standards across various domains, including both symmetric and asymmetric algorithms, and propose quantum-resistant recommendations to facilitate a secure transition for existing systems. Our analysis covers essential cryptographic primitives, such as block and stream ciphers, hash functions, digital signatures, and key establishment protocols. We also review global cryptographic standards and security recommendations from various countries to assess the current state of readiness for quantum threats.

Additionally, we examine the quantum readiness of cryptographic standards and provide essential Quantum-Resistant/Post-Quantum (PQ) recommendations for authentication. This includes entity-based authentication standards, public key infrastructure for certificate-based authentication, and specific PQ recommendations for widely used authentication protocols such as Kerberos, DNSSEC, SAML, and OAuth. We further explore the standards for communication protocols, including TLS, IPsec, SSH, FTP, and S/MIME, offering the necessary PQ recommendations for these protocols. These standards are crucial for securing data transmission and maintaining integrity across diverse applications, particularly as we transition to a quantum-resistant environment. Moreover, we assess hybrid cryptographic standards and emerging drafts that combine classical and quantum-resistant mechanisms to ensure backward compatibility and minimize disruption during the transition.

The proposed quantum-resistant recommendations offer immediate benefits by strengthening cryptographic defenses against both classical and quantum-based attacks. Organizations that adopt these PQ strategies, particularly in authentication and communication protocols, will better protect sensitive data and communications during the transition. This approach mitigates the risks associated with quantum vulnerabilities without requiring a complete redesign of existing systems, ensuring continuity. Furthermore, these recommendations provide forward-looking strategies to help organizations future-proof their security infrastructure as quantum technologies evolve.

The adoption of crypto-agile systems will be pivotal in navigating the uncertainties of the post-quantum era. Crypto-agile systems, designed to adapt to evolving cryptographic standards, will empower organizations to swiftly respond to future cryptographic challenges. Future research should continue to refine quantum-resistant cryptographic techniques, optimize hybrid implementations, and ensure seamless integration of these solutions into global standards and widely adopted protocols. By embracing these recommendations and fostering cryptographic agility, the global cryptographic community can pave the way for a quantum-resistant future, ensuring the long-term security of digital systems in the quantum age.

CRediT authorship contribution statement

Vikas Chouhan: Writing – review & editing, Writing – original draft, Validation, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Mohammed Aldarwbi:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Investigation, Formal analysis, Conceptualization. **Somayeh Sadeghi:** Writing – review & editing, Writing – original draft, Validation, Investigation, Formal analysis. **Ali Ghorbani:** Supervision, Resources, Project administration. **Aaron Chow:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis. **Robby Burko:** Validation, Supervision, Project administration, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 32
IETF cryptographic standards/drafts for well-known network protocols (IPsec/IKE).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
IKE	draft-ietf-ipsecme-ikev2-auth-announce-03 [233]	2023	Establish a mechanism in the IKEv2 that enables implementations to communicate the list of supported authentication methods	• Expired in October 2023 • Support: Asymmetric digital signature (RSA, DSS, ECDSA-SHA-256 (P-256 curve), ECDSA-SHA-384 (P-384 curve)), ECDSA-SHA-512 (P-521 curve) Hash functions (SHA-256/384)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations
	7296 [234]	2014	Describe IKEv2	• Support: Asymmetric Digital Signature (RSASSA-PKCS1-v1_5) Diffie–Hellman exchange Symmetric encryption (3DES, IDEA, AES-128/192/256) Hash (MD5, SHA-1) and MAC (HMAC, CMAC) functions PKI	• Refer to Table 12 and Table 13 for asymmetric digital signature and Key exchange, respectively • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 24 for PKI recommendations
	9370 [235]	2023	Describe the extension of the IKEv2 to allow multiple key exchanges	• Support: Asymmetric key exchange (DH,ECDH,PQ_KEM) Symmetric encryption (AES) Hash (SHA-2) and MAC (HMAC) functions	• Refer to Table 13 for Key exchange • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	7427 [236]	2015	Signature authentication in the IKEv2	• Introduces negotiation for the signature hash algorithm • Support: Asymmetric Digital Signature (RSA, DSA, ECDSA, RSASSA-PSS,ECGDSA, ElGamal) Asymmetric encryption (RSA) Hash (SHA-1, SHA-2, SHA-3) functions PKI	• Refer to Table 12 for asymmetric digital signature • Refer to Table 8 for asymmetric encryption • Refer to Table 10 for Hash recommendations • Refer to Table 24 for PKI recommendations
	6989 [237]	2013	Additional Diffie–Hellman Tests for the IKEv2	• Ensuring the secure operation of the IKEv2 specifically with elliptic curve groups • Support: Asymmetric Key exchange (DH, ECDH)	• Refer to Table 13 for key exchange
	4754 [238]	2007	IKE and IKEv2 Authentication Using the ECDSA	• Emphasizes that the addition of ECDSA does not require any modifications to the existing IKE • Support: Asymmetric digital signature (ECDSA-256, ECDSA-384, ECDSA-521) Hash functions (SHA-256/384/512)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations

(*continued on next page*)

Appendix A. Standardization organizations

Standardization organizations play a crucial role in establishing uniformity and quality benchmarks across industries worldwide, facilitating cooperation among stakeholders to develop and maintain standards that enhance efficiency, interoperability, and consumer safety. Below are the following standardization organizations that contribute to shaping global/national norms and practices across various sectors.

A.1. International standardization organizations

(i) *International Organization for Standardization (ISO)*
The standardization efforts of ISO[21] are carried out by multiple Technical Committees (TCs) that focus on specific fields of study. These committees assign responsibilities to Subcommittees (SCs), which further distribute the workload among multiple Working Groups (WGs). The development of cryptographic standards within ISO is primarily handled by two technical committees: TC68 and JTC1, the latter being a collaborative effort between ISO and the IEC. JTC1 has several subcommittees, including JTC1/SC17, JTC1/SC27, and JTC1/SC37, which all focus on different aspects of cryptographic standards. JTC1/SC17 is responsible for developing international standards for personal identification and security cards, while the main focus of JTC1/SC27 is on innovation related to information technology security, such as cryptography, intrusion detection, and incident management. Finally, JTC1/SC37 is responsible for developing international standards for biometric techniques, such as fingerprint recognition and facial recognition.
The technical committee, TC68, focused on the financial services industry, comprises several subcommittees that specialize in using cryptography to protect financial information. These subcommittees primarily interpret the general cryptography standards established by JTC1 and adapt them for use in the financial domain.
ISO standards are assigned standard numbers that are used to refer to them. For instance, ISO 8730 specifies the requirements for message authentication codes utilized in the financial sector. When ISO and the IEC publish a standard jointly via the JTC,

[21] https://www.iso.org

Table 32 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
	4307 [239]	2005	Define a set of mandatory-to-implement cryptographic algorithms for the IKEv2	• Support: Symmetric encryption (3DES-CBC, AES-128-CBC, AES_CTR, AES128_CBC, AES_XCBC_96) Hash and MAC functions (HMAC_MD5, HMAC_MD5_96, HMAC-SHA-1, HMAC_SHA-1_96)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	4434 [240]	2006	Describe the AES-XCBC-PRF-128 algorithm for the IKE	• Support: Symmetric encryption (AES-XCBC-PRF-128) Hash and MAC functions (HMAC-SHA-1, MAC-PRF)	
	5930 [241]	2010	Utilize the AES-CTR with the IKEv2	• AES-CTR is employed by IKEv2 to enhance the security and confidentiality of data transmission • Support: Symmetric encryption (AES-CTR)	• Refer to Table 6 for symmetric encryption
	8420 [242]	2018	Utilization of the EdDSA within the IKEv2	• Providing secure digital signatures and ensuring the authenticity and integrity of the exchanged data • Support: Asymmetric digital signature (EdDSA)	• Refer to Table 12 for asymmetric digital signatures
IKE-PQ	8784 [243]	2020	Mixing Preshared Keys in the IKEv2 for Post-quantum Security	• Describe the extension of IKEv2 to provide resistance to quantum computers through the utilization of preshared keys • Support: Asymmetric cryptography Symmetric encryption	• Refer to Tables 8, 12, and 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption
	draft-smyslov-ipsecme-ikev2-qr-alt-07 [244]	2023	Alternative procedure for Mixing Preshared Keys in IKEv2 for PQ Security	• Provide protection against quantum computers for both IPsec traffic and the initial IKEv2 SA in scenarios. This draft Expired in October 2023 • Support: Pre-shared key mechanism	• Refer to Table 13 for key establishment mechanisms

it is recognized as an ISO/IEC standard. Notably, ISO 27001[22] is one of the most widely adopted international standards for Information Security Management Systems (ISMS). It offers organizations a systematic approach to managing sensitive company information, ensuring its security across people, processes, and IT systems by implementing a robust risk management process. ISO 27002[23] provides best practice recommendations on information security management for use by those responsible for initiating, implementing, or maintaining ISMS. ISO 27017[24] offers guidelines for information security controls applicable to the provision and use of cloud services, ensuring both cloud service providers and their customers have the same information security controls to mitigate risks in the cloud environment. ISO 27018[25] focuses on protecting personal data in the cloud, establishing control objectives, controls, and guidelines for implementing measures to protect Personally Identifiable Information (PII) in accordance with the privacy principles in ISO 29100 for the public cloud computing environment. These ISO standards are essential for organizations to establish robust information security management practices, especially in the era of increasing cyber threats and data breaches. In addition to standards, ISO issues technical reports, which are specifications for specific tasks rather than actual standards. Technical reports can be recognized by the letters "TR", as seen in the example of ISO/IEC TR 14516, which provides recommendations regarding the utilization of trusted third parties.

(ii) *International Electrotechnical Commission (IEC)*
The IEC[26] is in charge of producing standards for all types of electrical and electronic technologies. Regarding cryptography and security-related issues, the IEC's primary emphasis is on its collaboration with ISO through the joint technical committee, JTC1. This collaboration helps the IEC to achieve its objectives in this field.

(iii) *International Telecommunication Union (ITU)*
ITU[27] is an organization sponsored by the United Nations that aims to facilitate global telecommunications networks and services through government and corporate collaboration. ITU is divided into three main branches, namely ITU-R, ITU-D, and ITU-T, where ITU-R is responsible for radio communication, ITU-D focuses on the development of telecommunications services, and ITU-T creates telecommunications standards. One of the most important standards in ITU-T is the X series, which is concerned with data networks and emphasizes network design over cryptography. However, the X series covers significant topics such as public-key infrastructures and cryptographic network services. ITU-T standards are typically referred to as "recommendations".

(iv) *Internet Engineering Task Force (IETF)*
The Internet is the outcome of a collaborative effort between various entities such as governments, academia, and enterprises to establish a global communication network. To ensure the Internet functions efficiently and securely, it must be built on

[22] https://www.iso.org/standard/27001
[23] https://www.iso.org/standard/75652.html
[24] https://www.iso.org/standard/82878.html
[25] https://www.iso.org/standard/76559.html
[26] https://www.iec.ch
[27] https://www.itu.int

Table 33
(Cont.) IETF cryptographic standards/drafts for well-known network protocols (IPsec/IKE).

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
IPsec/IKE	3602 [245]	2003	Describes the utilization of the AES-CBC Cipher Algorithm for the IPsec Encapsulating Security Payload (ESP)	• Support: Symmetric encryption (AES-CBC-128/192/256) Hash and MAC functions (SHA-256/384/512, HMAC)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	3566 [246]	2003	Introduces the AES-XCBC-MAC-96 algorithm and specifies its use with IPsec	• Support: Symmetric encryption (AES-XCBC, AES-128) Hash and MAC functions (HMAC)		
	4494 [247]	2006	Specify the use of the AES-CMAC-96 algorithm for the IPsec	• Support: Symmetric encryption (AES-CMAC-96) Hash and MAC functions (MAC, CMAC)		
	4894 [248]	2007	Discuss the usage of hash functions in the IKEv1, IKEv2, and IPsec protocols	• Support: Symmetric encryption (AES-XCBC-PRF-128) Hash and MAC functions (SHA-256/384/512, HMACs) PKI	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 24 for PKI recommendations	
	4308 [249]	2005	Provide the specification for cryptographic suites used in IPsec	• Support: Symmetric encryption (TripleDES-CBC,AES-XCBC) Hash and MAC functions Diffie–Hellman	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 13 for key establishments	
	6379 [250]	2011	Proposes cryptographic user interface suites for IPsec	• Support: Symmetric encryption (AES-GCM/GMAC-128/256) Hash and MAC functions (HMAC-SHA-256/384) Diffie–Hellman		
	4106 [251]	2005	Utilization of GCM mode in IPsec Encapsulating Security Payload (ESP)	• Support: Symmetric encryption (AES-GCM-128/192/256)	• Refer to Table 6 for symmetric encryption	
	4312 [36]	2005	Describe the utilization of the Camellia algorithm for IPsec	• Support: Symmetric encryption (Camellia-128/192/256)		
	3526 [252]	2003	Define new Modular Exponential (MODP) Diffie–Hellman Groups for the IKE	• Introduces new Diffie–Hellman groups with varying bit lengths (2048, 3072, 4096, 6144, and 8192) • Support: Asymmetric Key exchange (Diffie–Hellman (2048 to 8192 bit MODP Group)) Symmetric encryption (AES-128/192/256)	• Refer to Table 13 for key exchange • Refer to Table 6 for symmetric encryption	

(*continued on next page*)

standardized communication protocols, which the IETF[28] designs. Apart from the IETF, the Internet Research Task Force (IRTF) is also responsible for exploring research issues relevant to the Internet in the long run. The IRTF is focused on exploring new and innovative solutions to some of the Internet's most complex and challenging issues, such as network security, scalability, and performance. Through these efforts, the IRTF helps to ensure that the Internet remains a vital and reliable resource for people around the world.

In addition to developing Internet standards, the IETF has also created a set of publication records called Requests for Comments (RFCs). These documents provide guidance, recommendations, and information on a wide range of topics related to the Internet, including protocols, procedures, and concepts. RFCs are published openly and are freely available to the public, making them an invaluable resource for developers, researchers, and anyone interested in the workings of the Internet.

(v) *Internet Assigned Numbers Authority (IANA)*

IANA[29] initially administered by Jon Postel in the late 1970s, IANA's functions are now operated by the Internet Corporation for Assigned Names and Numbers (ICANN). It manages the global allocation of IP addresses, domain names, protocol parameters, and other Internet resources. IANA's work ensures the stable and secure operation of the Internet's unique identifiers through its coordination and management of various registries, such as the Domain Name System (DNS) root zone and the Internet Protocol (IP) address space. Its responsibilities include overseeing the assignment of Internet number resources to Regional Internet Registries (RIRs) and maintaining registries for protocol parameters defined by the Internet Engineering Task Force (IETF).

A.2. National standardization organizations

The establishment of national standardization organizations is a common practice across many countries aimed at ensuring standardiza-

[28] https://www.ietf.org

[29] https://www.ietf.org/process/iana

Table 33 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
	1828 [253]	1995	Utilization of keyed MD5 with the IP Authentication Header	• Keyed MD5 is utilized to provide integrity and authentication • Support: Hash function (MD5)	• Refer to Table 10 for Hash recommendations	
	2085 [254]	1997	Describe a keyed-MD5 transform to be used with the IP Authentication Header	• Support: Hash and MAC functions (HMAC, MD5)	• Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	2403 [255]	1998	Utilization of the HMAC algorithm with MD5 as an authentication mechanism in revised IPSEC			
	2404 [256]	1998	Utilization of the HMAC algorithm using SHA-1 for authentication purposes within the updated IPSEC	• Support: Hash and MAC functions (HMAC, SHA-1)		
	2857 [257]	2000	Utilization of the HMAC approach using the RIPEMD-160 algorithm for authentication within the revised IPSEC	• Support: Hash and MAC functions (HMAC, RIPEMD-160)		
	4868 [258]	2007	Employing HMAC along with the SHA-256, SHA-384, and SHA-512 algorithms within the realm of IPsec	• Support: Hash and MAC functions (HMAC, SHA-256/384/512)		
	6380 [259]	2011	Specify the conventions and guidelines for using Suite B cryptography in (IPsec)	• Providing relevant details and aiding developers who opt to implement Suite B cryptography in IPsec • Support: Asymmetric Key exchange (ECDH (P-256 & P-384)) Asymmetric digital signature (ECDSA (P-256 & P-384)) Symmetric encryption (AES-128/256) Hash (SHA-256/384) and MAC (HMAC, GMAC-128, GCM-256) functions PKI	• Refer to Tables 12 and 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 24 for PKI recommendations	
	5529 [35]	2009	Utilization of the Camellia algorithm in various modes (CBC, CTR, and CCM) with IPsec	• Camellia algorithm is an optional-to-implement component in IKEv2 and ESP, offering robust security features for secure communication over the Internet • Support: Symmetric encryption (Camellia-CBC/CTR/CCM-128/192/256) MAC function	• Refer to Table 6 for symmetric encryption • Refer to Table 11 for MAC recommendations	

tion in various fields. These entities have the autonomy to produce their standards while also collaborating with international standardization bodies to contribute to the development of global standards. This dual role enables national standardization bodies to maintain a balance between domestic needs and global standards, promoting interoperability and enhancing efficiency in various sectors.

(i) *American National Standards Institute (ANSI)*
ANSI[30] functions as the coordinating organization for voluntary standardization initiatives in the United States (U.S.) and represents the U.S. in the ISO. In addition to its coordination role, ANSI is also responsible for developing its own standards, which cover a wide range of industries and technologies. Among the standardization bodies accredited by ANSI, X9[31] is the most eminent in cryptography. X9 is dedicated to creating, publishing, and advocating standards for the financial services industry and working in close collaboration with the ISO's TC68 committee. Thanks to its expertise and contributions, X9 has played a vital role in ensuring the security and integrity of financial transactions worldwide, and its standards are widely adopted by financial institutions and organizations.

(ii) *British Standards Institution (BSI)*
BSI[32] is responsible for managing standardization efforts in the United Kingdom and serves as the official member body representing the country in ISO. While BSI is not widely known for its technical cryptographic standards, it has made significant contributions to the field of management techniques. For instance, BSI has produced several influential standards, such as the BS 7799-2 standard, which provides guidelines and requirements for information security management systems. Overall, BSI's contributions to the field of standardization extend beyond cryptography and highlight the importance of organizational management and processes in ensuring security and efficiency in various sectors. Later, it was adopted by ISO (ISO/IEC 27001) in 2005.

(iii) *National Institute of Standards and Technology (NIST)*
NIST[33] is a significant standards organization in the United States that operates as the federal agency responsible for standardization within the technology administration of the U.S. Commerce Department. In addition to its wide-ranging secu-

[30] http://www.ansi.org
[31] https://x9.org
[32] https://www.bsigroup.com
[33] https://www.nist.gov

Table 34
IETF cryptographic standards/drafts for well-known network protocols (SSH).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
SSH	4251 [260]	2006	SSH Protocol Architecture	• Support: Symmetric encryption (3DES, AES) Asymmetric key exchange (DH) Hash and MAC functions (MAC, HMAC, SHA-1)	• Refer to Table 6 for symmetric encryption • Refer to Table 13 for Key exchange • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	6668 [261]	2012	SHA-2 Data Integrity Verification for the SSH	• Support: Hash and MAC functions (HMAC, SHA-2-256, SHA-2-512)	• Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	draft-gutmann -ssh-preauth-00 [262]	2022	A Pre-Authentication Mechanism for SSH. This draft Expired in June 2023.	• Support: Hash and MAC functions (HMAC, SHA-256)	
	9142 [263]	2022	Revisions to the suggested number of key exchange techniques within the SSH protocol	• Revision of SSH protocol's key exchange methods to address the need for stronger security • Support: Asymmetric key exchange (ECC, DH, ECDH, RSA-1024/2048) Symmetric encryption (3DES-cbc, AES-128/192/256) Hash functions (SHA-1, SHA-256/384/512)	• Refer to Table 6 for symmetric encryption • Refer to Table 13 for Key exchange • Refer to Table 10 for Hash recommendations
	6594 [264]	2012	Enhance the SSHFP DNS Resource Record by incorporating the SHA-256 algorithm with RSA, DSA, and ECDSA	• Support: Asymmetric digital signature (RSA, DSA, ECDSA) Hash-function (SHA-256)	• Refer to Table 12 for asymmetric digital signatures • Refer to Table 10 for Hash recommendations
	8332 [265]	2018	Introducing new public key algorithms using RSA keys alongside SHA-256 and SHA-512 in the SSH Protocol	Asymmetric digital signature (RSASSA-PKCS1-v1_5) Hash functions (SHA-256/512)	
	8709 [266]	2020	Outlines the utilization of the Ed25519 and Ed448 digital signature schemes in the SSH	Asymmetric digital signature (Ed25519 and Ed448) Hash functions (SHA-256)	
	8731 [267]	2020	Specifies the usage of Curve25519 and Curve448 key exchange algorithms in the SSH protocol	• Incorporating key exchange methods in SSH for enhanced security and cryptographic operations • Support: Asymmetric key exchange (ECDH, Curve25519, Curve448), key agreement (ECMQV), and digital signature (ECDSA) Hash functions (SHA-256/512)	• Refer to Table 12 and Table 13 for asymmetric digital signature and key establishment mechanisms, respectively • Refer to Table 10 for Hash recommendations
	8268 [268]	2017	Enhancement of the Modular Exponentiation (MODP) DH Key Exchange (KEX) Groups dedicated to SSH	• Rectification of an error pertaining to the verification of the Peer's DH Public Key • Support: Asymmetric key exchange (DH, ECDH, ECDSA) Hash functions (SHA-256/512)	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 10 for Hash recommendations

(*continued on next page*)

rity standards, NIST has also developed various measurement, testing, and evaluation protocols. Nonetheless, NIST is perhaps best known for its role in creating the AES encryption algorithm, which is broadly implemented. NIST issues its standards as Federal Information Processing Standards (FIPS), which are mandated for use in various U.S. government agencies and by many private organizations. NIST is currently in the process of developing standards for Post-Quantum (PQ) cryptography, which is designed to be resistant to attacks by quantum computers [8]. The goal is to ensure that the cryptographic systems used to secure our digital infrastructure remain secure even in the face of quantum computing advances.

A.3. Industrial standardization organizations

The production of cryptographic standards is not limited to governments and government-led organizations, as the business community also takes the initiative to establish standards in this field.

(i) *Third Generation Partnership Project (3GPP)*
The 3GPP[34] is a collaborative platform that unites multiple corporations and standardization organizations in developing third-generation cellular networks. Given that security and privacy are crucial aspects of these advanced networks, the 3GPP

[34] http://www.3gpp.org

Table 34 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
	6239 [269]	2011	Cryptographic Suites for SSH	• Incorporates ECDH key agreement, ECDSA, AES-GCM, SHA-2 family hashes (SHA-256 and SHA-384), and X.509 certificates • Support: Asymmetric key agreement (ECDH) and digital signature (ECDSA) Symmetric encryption (AES-GCM-128/256) Hash (SHA-256 and SHA-384) and MAC functions	• Refer to Table 12 and Table 13 for asymmetric digital signature and key establishment mechanisms, respectively • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	4252 [270]	2006	Define the SSH authentication protocol designed for secure remote login and network services	• Support: Asymmetric cryptography Hash functions	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 10 for Hash recommendations	
	4432 [271]	2006	RSA Key Exchange for the SSH Protocol	• Support: Asymmetric encryption (RSAES-OAEP (1024,2048)) Hash functions (SHA-1,SHA-256)		
	4344 [272]	2006	SSH Transport Layer Encryption Modes	• Provides guidelines for SSH implementation rekeying • Support: Symmetric encryption (3DES-CTR, AES-CTR-128/192/256) Hash and MAC functions (HMAC, SHA-1)	• Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	4462 [273]	2006	Outlines the usage of the Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange within the framework of the SSH Protocol	• Support: Asymmetric key exchange (DH) Hash-function (SHA-1)	• Refer to Table 13 for asymmetric key exchange • Refer to Table 10 for Hash recommendations	
SSH-Hybrid	draft-kampanakis-curdle-ssh-pq-ke-01[111]	2023	Defines PQ hybrid key exchange methods in the SSH	• It combines classical ECDH key exchange with PQ KEM. This draft Expired in October 2023 • Support: Asymmetric key exchange: Classical (ECDH (x25519)) and PQ (kyber-512/768/1024) Hash and MAC functions (SHA-256/384/512, HMAC)	• Refer to Table 13 for asymmetric key exchange • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	

has played a crucial role in standardizing both network security characteristics and cryptographic approaches to ensure the protection of sensitive information transmitted over these networks. Through this effort, the 3GPP has helped to establish a high level of security for mobile communications, which has become increasingly essential in our digital age.

(ii) *European Telecommunications Standard Institute (ETSI)*
ETSI[35] is a self-governing entity that consists of organizations that have a mutual objective of creating telecommunications standards. It is among the organizational collaborators that participate in coordinating the 3GPP project.

(iii) *Institute of Electrical and Electronics Engineers (IEEE)*
IEEE[36] is a professional organization of engineers that strives to facilitate the exchange of technical knowledge and information related to electrical and electronic engineering. It also plays a pivotal role in hosting conferences and publishing books and journals to promote the dissemination of information in these fields. From a cryptographic standpoint, there are two major standardization groups within the IEEE that are of particular interest. The first group, known as the 1363 group, concentrates on developing standards related to asymmetric cryptography. These standards cover a broad range of topics, including key management, digital signatures, and public key encryption. The second major standardization group is the 802 group, which is focused on developing standards for wireless local area networks. This group is responsible for producing the widely used 802.11 series of standards, which have revolutionized the way we connect to the Internet and communicate wirelessly.

(iv) *Standards for Efficient Cryptography Group (SECG)*
Elliptic Curve Cryptography (ECC) was introduced as a novel type of public-key cryptography based on complex mathematical structures in the early 1990s. A group of businesses spearheaded by Certicom was the first to recognize the potential of this innovative cryptographic approach in the corporate world. To facilitate interoperability and tackle practical concerns associated with ECC, these businesses established the SECG.[37] In the year 2000, two standards were released by the SECG. The first, SEC 1, standardized the application of ECC in cryptography. This standard laid out a framework for implementing elliptic curve cryptography in various settings such as digital signatures, key exchange, and encryption. The second standard, SEC 2, advocated specific parameters, such as curve equations and point representations, for utilizing ECC.

(v) *Public-Key Cryptography Standards (PKCSs)*
RSA Laboratories stands out as one of the few companies that have independently developed a comprehensive set of standards in addition to collections of businesses producing industrial

[35] http://www.etsi.org
[36] https://standards.ieee.org
[37] http://www.secg.org

Table 35
IETF cryptographic standards/drafts for well-known network protocols (FTP).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
FTP	2228 [274]	1997	FTP Security Extensions	• Specify Internet standards and solicit feedback for improvement • Support: Asymmetric keys (RSA) Hash functions Authentication	• Refer to Table 8 for asymmetric cryptography • Refer to Table 10 for Hash recommendations • Refer to Table 23 for Authentication
	2577 [275]	1999	To address security vulnerabilities in the FTP specification and provide recommendations for mitigating associated risks	• Highlights issues with proxy FTP and unlimited password attempts, offering suggestions for improving security in FTP implementations • Support: Authentication	• Refer to Table 23 for Authentication
SFTP	4253 [276]	2006	Describes SSH as a secure protocol for remote login and network services over an insecure network	• Defines SSH transport layer, encryption, authentication, and algorithms used for secure network communication • Support: Asymmetric key exchange (DH) and signature (RSASSA-PKCS1-v1_5, DSS) Symmetric encryption (3DES-CBC, AES-CBC-128/192/256) Hash and MAC functions (MD5, SHA-1, HMAC) Authentication and SSH	• Refer to Table 12 and Table 13 for asymmetric digital signature and Key exchange, respectively • Refer to Table 6 and Table B.37 for symmetric encryption and recommendations on classical and PQ modes • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 23 for Authentication • Refer to Table 34 for SSH
	6668 [261]	2012	Defines algorithm names and parameters for SHA-2 family hash algorithms for data integrity verification	• Introduces a novel data integrity algorithm for SSH, updating RFC 4253 • Support: Hash and MAC functions (HMAC, SHA-2-256, SHA-2-512) SSH	• Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 34 for SSH
FTPS	4217 [277]	2005	To outline a mechanism for implementing TLS-based security and authentication in FTP clients and servers	• Specifies the required extensions and parameters, discusses policy considerations, and promotes interoperability • Support: Authentication and TLS	• Refer to Table 23 for Authentication • Refer to Table 31 for TLS

standards. The PKCS began as simple standards for using the RSA encryption and signature schemes but later expanded to cover all aspects of asymmetric cryptography, including key management, cryptographic message syntax, and cryptographic token interfaces. The PKCS series has been widely adopted and has significantly impacted the development of public-key cryptography.

(vi) *CA/Browser Forum*
Formed in 2005, the CA/Browser Forum[38] comprises leading Certificate Authorities (CAs) and browser vendors, aiming to enhance the security and trustworthiness of SSL/TLS certificates used in web encryption. The forum establishes guidelines and best practices for CAs and browser vendors, developing baseline requirements and extended validation guidelines to ensure the integrity and authenticity of digital certificates. Its work addresses evolving security threats and industry needs, promoting the adoption of cryptographic best practices and standards-compliant certificate issuance and validation processes. Through collaboration and consensus-building, the CA/Browser Forum strives to maintain trust in the Public Key Infrastructure (PKI) ecosystem and protect users from fraudulent activities and security vulnerabilities.

(vii) *Organization for the Advancement of Structured Information Standards (OASIS)*
Founded in 1993 as a nonprofit consortium, OASIS[39] develops open standards for security, cloud computing, web services, and other areas to facilitate interoperability and promote industry collaboration. It hosts technical committees and working groups that define and refine specifications, such as the Security Assertion Markup Language (SAML) and the OpenDocument Format (ODF), through a transparent and consensus-driven process. OASIS provides a neutral and collaborative environment for stakeholders from diverse industries to collaborate on the development of standards that address common challenges and enable the seamless exchange of information and services across different platforms and systems.

(viii) *Post-Quantum Cryptography (PQC) Alliance and Coalition*
The emergence of the Post-Quantum Cryptography (PQC) Alliance[40] and Coalition[41] was prompted by the increasing threat posed by quantum computing. These organizations are dedicated to standardizing and promoting cryptographic algorithms and protocols resistant to quantum attacks. By collaborating to evaluate and endorse candidate algorithms, they contribute to the development of future-proof cryptographic standards. Engaging researchers, cryptographers, industry stakeholders, and standards bodies, they strive to advance the state of cryptography and ensure long-term security against emerging threats. Their work is crucial in preparing for the transition to quantum-safe cryptographic algorithms and mitigating risks to information security and privacy.
The landscape of Post-Quantum Cryptography (PQC) has witnessed the emergence of two significant entities: the PQC Alliance and the PQC Coalition. Their shared purpose is to address

[38] https://cabforum.org
[39] https://www.oasis-open.org/org
[40] https://pqca.org
[41] www.mitre.org/news-insights/news-release/post-quantum-cryptography-coalition-launches

Table 36
IETF cryptographic standards/drafts for S/MIME.

Protocol	RFC	Year	Purpose	Note	Recommendations	
					Classical (till 2030 & beyond)	PQ
S/MIME	8551 [279]	2019	S/MIME Version 4.0 Message Specification	• Outlines how to add cryptographic signature and encryption services to MIME data, including using the multipart/signed media type for S/MIME signed messages • Support: Asymmetric digital signature (RSA, ECDSA), encryption (RSA) and key establishment (DH, ECDH), Symmetric encryption (AES-GCM-128/256, ChaCha20), Hash and MAC functions (MD5, SHA-1, SHA-2, Poly1305, HMAC)	• Refer to Table 8, Table 12, and Table 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	2634 [280]	1999	Enhanced Security Services for S/MIME	• Support: Asymmetric digital signature (RSA,DSS), Hash function (SHA-1)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 for Hash recommendations	
	6210 [281]	2011	Hash Functions with Parameters in the CMS and S/MIME	• Support: Asymmetric digital signature (RSA), Hash and Mac functions (MD5, SHA-1, HMAC)	• Refer to Table 12 for asymmetric digital signature • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively	
	3657 [282]	2004	Utilizing the Camellia encryption algorithm with CMS	• Support: Symmetric encryption (Camellia-cbc-128/192/256, AES-128/192/256)	• Refer to Table 6 for symmetric encryption	
	9216 [283]	2022	S/MIME Example Keys and Certificates	• Defining a small set of X.509v3 certificates and keys • Support: Asymmetric cryptography (RSA, Ed25519, ECDH), symmetric encryption (AES-128), Hash and MAC functions (SHA-256, HMAC), PKI (X.509)	• Refer to Table 8, Table 12, Table 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively • Refer to Table 24 for PKI recommendations	
	9219 [284]	2022	Signature Verification	• Specifies an extension to "The JSON Meta Application Protocol (JMAP) for Mail" (RFC 8621) for returning the S/MIME signature verification status • Support: Digital Signature	• Refer to Table 12 for asymmetric digital signature	
	8823[285]	2021	Automatic Certificate Management Environment	• The ACME process for issuing S/MIME certificates to email users requires specific identifiers and challenges • Support: Hash functions (SHA-256), PKI (X.509)	• Refer to Table 10 for Hash recommendations • Refer to Table 24 for PKI recommendations	
	8550 [286]	2019	(S/MIME) Version 4.0 Certificate Handling	• Support: Asymmetric Encryption and signatures (RSA, ECDSA, DSA), Hash functions (SHA-1, SHA-256), PKI (X.509, PKIX)	• Refer to Table 8 and Table 12 for asymmetric cryptography • Refer to Table 10 for Hash recommendations • Refer to Table 24 for PKI recommendations	

(*continued on next page*)

the threat posed by quantum computing to cryptographic systems. Focusing on standardization and promotion of quantum-resistant cryptographic algorithms, they collaborate to evaluate and endorse candidate algorithms, contributing to the establishment of future-proof cryptographic standards. Their responsibility involves engaging various stakeholders to advance cryptography and ensure long-term security against emerging threats.

Comprised of members such as IBM Quantum, Microsoft, MITRE, PQShield, SandboxAQ, and the University of Waterloo, the PQC Coalition is dedicated to facilitating global adoption of PQC. Their efforts include advancing standards relevant to PQC migration, creating educational materials, producing open-source, production-quality code, and ensuring cryptographic agility. The Coalition aims to streamline organizational migration to PQC and provide comprehensive guidance for the community.

In February 2024, the Linux Foundation introduced the Post-Quantum Cryptography Alliance (PQCA) to drive the advancement and adoption of post-quantum cryptography. Supported by founding members like Amazon Web Services (AWS), Cisco, Google, and IBM, the PQCA aims to secure sensitive data and communications in the post-quantum era. Their objective includes providing production-ready libraries and packages and enabling cryptographic agility across the ecosystem. Engaging in various technical projects, such as software development for new post-quantum algorithms and hosting initiatives like the Open Quantum Safe project and the PQ Code Package Project, the PQCA welcomes organizations and individuals to participate and contribute to the advancement of post-quantum cryptography.

Appendix B. Standards for modes of operation

As shown in Table B.37, each operational mode is analyzed for its intended purpose (confidentiality, authentication, or both), standardization body, and year of publication. The 'Mode' column confirms support for all listed modes in their respective rows, while the 'Algorithm' column indicates support for corresponding modes listed in their respective rows. Furthermore, the 'Note' column provides specific information about each standard. For detailed guidelines on cryptographic recommendations, including the basic criteria we follow, please refer to Section 3.2.

B.1. Recommendations

Based on our recommendations presented in Table B.37, we conclude CBC, CFB, OFB, and CTR are post-quantum resistant if they come with a block cipher that is post-quantum safe such as AES-256, which means the security of these modes depends on the security of the block

Table 36 (*continued*).

Protocol	RFC	Year	Purpose	Note	Recommendations
					Classical (till 2030 & beyond) PQ
	8591 [287]	2019	SIP-Based Messaging with S/MIME	• Support: Asymmetric encryption (RSA) and key agreement (ECDH-curve25519), Symmetric encryption (AES-128), Hash function (SHA-256), PKI	• Refer to Table 8 and Table 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations • Refer to Table 24 for PKI recommendations
	8162 [288]	2017	Using Secure DNS to Associate Certificates with Domain Names for S/MIME	• Defines how to associate an S/MIME user's certificate with a domain name using secure DNS • Support: Hash functions (SHA-256, SHA-512), TLS, PKI	• Refer to Table 10 for Hash recommendations • Refer to Table 31 for TLS • Refer to Table 24 for PKI recommendations
	7508 [289]	2015	Securing Header Fields with S/MIME	• Explains how the S/MIME protocol can secure message headers, known as 'Secure Headers', defined in RFC 5322 • Support: TLS	• Refer to Table 31 for TLS
	6318 [290]	2011	Suite B in S/MIME	• Support: Asymmetric signature (ECDSA-P-256/384) and key agreement (ECDH-P-256/384), symmetric encryption (AES-CBC-128/256), Hash function (SHA-256/384)	• Refer to Table 12 and Table 13 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations
	6664 [291]	2012	S/MIME Capabilities for Public Key Definitions	• Support: Asymmetric signature (RSASSA-PSS), key transport (RSAES-OAEP) and key agreement (DH), Hash function (MD5, SHA-1, SHA-256)	• Refer to Table 8, Table 12, and Table 13 for asymmetric cryptography • Refer to Table 10 for Hash recommendations
	7107[292]	2014	Object Identifier Registry for the S/MIME Mail Security Working Group	• Support: Asymmetric cryptography (RSA, ECC), symmetric encryption (AES), Hash and Mac functions (MD5, HMAC)	• Refer to Table 8 for asymmetric cryptography • Refer to Table 6 for symmetric encryption • Refer to Table 10 and Table 11 for Hash and MAC recommendations, respectively
	4262 [293]	2005	X.509 Certificate Extension for S/MIME Capabilities	• Support: PKI	• Refer to Table 24 for PKI recommendations
	7281 [294]	2014	Authentication-Results Registration for S/MIME Signature Verification		
	3853 [295]	2004	S/MIME AES Requirement for the Session Initiation Protocol (SIP)	• Require the AES for S/MIME • Support: Asymmetric signature (RSA), Symmetric encryption (AES), Hash function (SHA-1), TLS	• Refer to Table 12 for asymmetric signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations • Refer to Table 31 for TLS
	3854 [296]	2004	Securing X.400 Content with S/MIME	• Describes a protocol for adding cryptographic signature and encryption services to X.400 content with S/MIME • Support: Asymmetric signature (RSA, DSA), Symmetric encryption (AES), HASH function (SHA-1)	• Refer to Table 12 for asymmetric signature • Refer to Table 6 for symmetric encryption • Refer to Table 10 for Hash recommendations

cipher. In addition, chosen block cipher should be PRF, and finally, each mode should be IND-qCPA qPRF to be post-quantum resistant. We also provide recommendations for each mode regarding classical and post-quantum cryptography. CBC, CFB, OFB, CTR, CMAC, CCM, GCM, KW, KWP, and TKW modes are considered post-quantum safe under certain conditions. However, it is essential to note that using 3DES block cipher in any mode will be prohibited[42] after 2023 (By NIST [32]). In contrast, encrypting confidential information using ECB, as indicated in the NIST National Vulnerability Database (NVD), is considered a severe security vulnerability.

Appendix C. Stream ciphers

C.1. Stream ciphers standards

We thoroughly examined the Stream Ciphers Standards and have summarized them in Table 7, detailing several key columns. For detailed guidelines on table columns and cryptographic recommendations, please refer to Section 3.2.

(i) *ChaCha20:* ChaCha20 is a reliable and versatile stream cipher algorithm widely used for secure encryption. It has gained popularity due to its simplicity, efficiency, and high level of security. The IETF has standardized ChaCha20, and various RFCs are available in Table 7 to cover different aspects of it.
Based on the evaluation presented by Bathe et al. [40], ChaCha20-256, in its current configuration of 20 rounds, cannot be considered quantum-resistant to a degree comparable to AES-256. The analysis demonstrates that ChaCha20 falls short of meeting the quantum security criteria established by NIST, primarily due to its lower complexity when subjected to Grover's quantum search algorithm. Specifically, ChaCha20-256 would require significantly fewer quantum resources to break compared to AES-256. According to the authors' estimations, ChaCha would need approximately 166 rounds (far beyond its standard implementation) to attain a quantum security level equivalent to AES-256. For clarity and ease of reference, this recommended configuration can be denoted as ChaCha166-256, indicating a version of ChaCha with a 256-bit key and 166 rounds. ChaCha20-256 can be considered tentatively quantum-safe and is recommended for non-critical applications [308].

[42] https://csrc.nist.gov/news/2023/nist-to-withdraw-sp-800-67-rev-2

Table B.37
Standards for modes of operation.

Mode	Standards	Year	Purpose	Algorithm	Note	Recommendations	
						Classical (till 2030 & beyond)	PQ (Available)
ECB	NIST SP 800-38E [300]	2010	Confidentiality Modes	AES	Consider disallowing ECB to encrypt secrets. (March 2023)	NIST proposes limiting ECB approval to instances explicitly allowed by other NIST standards or guidance.	
CBC					Security of CBC depends on the security of AES (as a PRP)	CBC-128/192/256	CBC-256 (IND-qCPA qPRF)
CFB					Camellia-256	CFB-128/192/256	CFB-256 (IND-qCPA qPRF)
OFB						OFB-128/192/256	OFB-256 (IND-qCPA qPRF)
CTR						CTR-128/192/256	CTR-256 (IND-qCPA qPRF)
XTS	NIST SP 800-38E [300]	2010			The key size for XTS is twice the security strength	128, 192, 256 bit	None
FF1	NIST SP 800-38G [301]	2013				None	None
FF3							
CMAC	NIST SP 800-38B [302]	2005	Authentication mode		CMAC is a variant of CBC-MAC, the security properties of CMAC are similar to those of CBC	128, 256 bit	CMAC-256 (IND-qCPA qPRF)
CCM	NIST SP 800-38C [303]	2004	Combined modes for confidentiality and authentication		It is considered to be secure when used with a secure block cipher, such as AES. The size of the MAC also affects the security of CCM mode. A larger MAC size provides greater security	128, 192, 256 bit	256 bits
GCM	NIST SP 800-38D [304] IEEE 802.11 [305]	2007			Suitable for files less than 64 GiB The size of the MAC is fixed at 128 bits. The recommended authentication tag size for GCM mode is 128 bits, which provides a high level of security against known attacks	128, 192, 256 bit	
KW	NIST SP 800-38G [301]	2013			To wrap a 256-bit key using AES Key Wrap, you would need to use AES-256 for the encryption steps	128, 192, 256 bit	
KWP							
TKW							
TECB-I	NIST SP 800-20* [306] NIST SP 800-38F [307]	2012	Confidentiality Modes	3DES (ANSI X9.52)		Disallowed after 2023	
KW			Combined modes for confidentiality and authentication				
TKW							

(*) Withdrawn on September 26, 2018.

Table D.38
Public-key algorithms broken by Shor's algorithm.

Algorithm	Function	Security based on	Specification
RSA (PKCS#1, OAEP, PSS)	Key Establishment, Digital Signatures	Integer Factorization	PKCS#1, FIPS 186
Diffie–Hellman (DH)	Key Establishment	Discrete Logarithm (finite field)	NIST SP 800-56A/B/C, RFC 3526
Elliptic Curve Diffie–Hellman (ECDH)	Key Establishment	Discrete Logarithm (elliptic curve)	NIST SP 800-56A, FIPS 186
MQV, ECMQV	Key Establishment	Discrete Logarithm (elliptic curve)	NIST SP 800-56A
X25519, X448	Key Establishment	Discrete Logarithm (elliptic curve)	RFC 7748
DSA	Digital Signatures	Discrete Logarithm (finite field)	FIPS 186
ElGamal	Encryption	Discrete Logarithm (finite field)	IEEE P1363
ECDSA	Digital Signatures	Discrete Logarithm (elliptic curve)	FIPS 186-5
Ed25519, Ed448 (EdDSA)	Digital Signatures	Discrete Logarithm (elliptic curve)	RFC 8032
BLS Signatures	Digital Signatures	Discrete Logarithm (pairing-friendly curves)	draft-irtf-cfrg-bls-signature
ECIES, EC-ElGamal	Encryption/Key Encapsulation	Discrete Logarithm (elliptic curve)	ISO/IEC 18033-2

(ii) *RC4:* The RC4 algorithm is a popular symmetric stream cipher used to encrypt data in many cryptographic systems and protocols. However, it is important to be aware that RC4 is no longer considered secure for most applications due to vulnerabilities in its design that have been discovered over time [309]. Table 7 provides details on the standardization of RC4 by

Table D.39
Symmetric algorithms affected by Grover's algorithm.

Category	Examples	Mitigation/Recommendation
Block Ciphers	AES-128, SM4, SEED	Use AES-256 to restore ≥128-bit post-quantum security
Stream Ciphers	ChaCha20, Salsa20	Use AES-256 in CTR, OFB or CFB mode, with a recommendation to use CTR mode
Hash Functions (collision resistance)	SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-512, SM3	Use SHA-512 or SHA3-512 to maintain ≥128-bit post-quantum collision resistance
Hash Functions (preimage resistance)	SHA-256, SHA-512, SHA3-family	SHA-256 provides 128-bit PQ security; SHA-512 provides 256-bit PQ security
Message Authentication	HMAC-SHA256, HMAC-SHA512, Poly1305	Use HMAC-SHA512 or 256-bit keys to maintain ≥128-bit post-quantum security
Authenticated Encryption	AES-128-GCM, AES-256-GCM, ChaCha20–Poly1305	Use AES-256-GCM or ChaCha20–Poly1305 with 256-bit keys; 128-bit tags remain secure

Table D.40
Recommended post-quantum replacements for vulnerable classical algorithms.

Replace this	With (PQC algorithm)	Notes/Remarks
RSA, DH, ECDH, MQV, X25519	ML-KEM-768/1024 (Kyber)	NIST FIPS 203; lattice-based KEM; hybrid mode (e.g., X25519+ML-KEM-768) recommended during transition
RSA, ECDSA, EdDSA, DSA	ML-DSA-65/87 (Dilithium), SLH-DSA-128/256 (SPHINCS+)	NIST FIPS 204, 205; ML-DSA balances size/speed; SLH-DSA is stateless hash-based; FN-DSA (Falcon) under consideration

IETF, NIST, IEEE, and ISO. However, it is no longer recommended by any of these organizations. For encryption purposes, it is recommended to use more secure alternatives, such as the AES.

(iii) *SNOW 2.0:* The Snow stream cipher was first created for cellular communication standards like 3G and 4G. Its purpose is to offer efficient and secure encryption for wireless communication. ISO/IEC 18033 includes SNOW 2.0 stream cipher algorithms standards and recommendations [310].

(iv) *Trivium:* Trivium is a stream cipher method that is able to balance speed and hardware usage, while still being efficient when implemented in software. Trivium is a hardware-efficient cipher that is easy to use and performs well. However, it has been analyzed and found to have weaknesses. To ensure secure communication and data protection, it is recommended to use other modern and thoroughly tested stream ciphers, such as those found in the eSTREAM portfolio or the AES [311]. As a lightweight stream cipher, ISO has some standardization and recommendations for it in ISO/IEC 29192-3:2012 [48].

(v) *Enocoro:* Enocoro is a family of pseudorandom number generators (stream ciphers). It comprises two algorithms, Enocoro-80 and Enocoro-128v1.1, which have key lengths of 80 bits and 128 bits, respectively [312]. ISO/IEC 29192-3:2012 [48] provides standardization and recommendations for Enocoro.

(vi) *Grain:* There are two types of Grain ciphers: an 80-bit and a 128-bit variant labeled as Grain and Grain-128, respectively. These designs utilize two shift registers and a nonlinear output function, and are intended for hardware environments with limited gate count, power consumption, and memory. Additionally, expanding the hardware can make the ciphers faster, but this comes at a cost [313]. Grain-128AEAD is a highly compatible choice for the Internet of Things (IoT) and embedded systems due to its significant advantages. Its previous versions have also proven to be relevant in industrial applications. NIST and ISO provide recommendations for handling Grain in their respective publications, NISTIR 8369 and ISO/IEC 29167-13:2015 [50].

(vii) *AES-CTR:* AES-CTR is a stream cipher that utilizes the AES block cipher. It XORs the key stream generated by AES encryption of sequential counter block values to encrypt and decrypt data. AES-CTR is a stream cipher mode that is easy to implement, parallelize, and pipelined. It also supports key stream precomputation and has a smaller implementation size compared to other AES modes.

Appendix D. Algorithms vulnerable to quantum attacks

Shor's algorithm efficiently solves two mathematical problems that are fundamental to modern public-key cryptography: integer factorization and the discrete logarithm problem (in both finite fields and elliptic curve groups). As a result, any cryptographic scheme that depends on these computational hardness assumptions will become completely insecure once large-scale, fault-tolerant quantum computers are realized. Table D.38 summarizes the major affected algorithms and their underlying security foundations.

Once a sufficiently powerful quantum computer becomes available, the above schemes will provide no security guarantees whatsoever. Past ciphertexts and signatures generated using these algorithms can be retrospectively compromised under a harvest-now, decrypt-later (HNDL) threat model, in which adversaries collect encrypted data today to decrypt once quantum capabilities mature. Organizations managing long-lived sensitive data, such as medical records, financial transactions, or classified government communications, should prioritize the immediate transition away from these algorithms.

While Shor's algorithm completely compromises public-key cryptosystems, Grover's algorithm provides a quadratic speedup for exhaustive key search, effectively reducing the security strength of symmetric primitives by up to half in the worst case. For example, a 128-bit key provides at most 64-bit quantum security against Grover's attack, whereas a 256-bit key retains up to 128-bit post-quantum security. Table D.39 summarizes the effects on symmetric algorithms and the corresponding recommended countermeasures.

D.1. PQC replacements

To mitigate quantum threats, NIST has standardized PQC algorithms specifically designed to resist both classical and quantum attacks. These algorithms are based on mathematical problems believed to be hard for quantum computers, including lattice problems, hash-based signatures and code-based cryptography. During the transition period, hybrid

modes that combine classical and PQC algorithms are strongly recommended to ensure security against both current and future threats. Table D.40 provides a practical migration roadmap from vulnerable classical schemes to their quantum-resistant counterparts.

Data availability

No data was used for the research described in the article.

References

[1] H.-Y. Kwon, I. Bajuna, M.-K. Lee, Compact hybrid signature for secure transition to post-quantum era, IEEE Access (2024).

[2] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. 41 (2) (1999) 303–332.

[3] I. Kong, M. Janssen, N. Bharosa, Realizing quantum-safe information sharing: Implementation and adoption challenges and policy recommendations for quantum-safe transitions, Gov. Inf. Q. 41 (1) (2024) 101884, http://dx.doi.org/10.1016/j.giq.2023.101884, URL https://www.sciencedirect.com/science/article/pii/S0740624X23000849.

[4] NIST, Post-quantum cryptography standardization, 2024, NIST, Available: https://csrc.nist.gov/pqc-standardization.

[5] NIST, Post-quantum cryptography round 4 submissions, 2024, NIST, Available: https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions.

[6] C. Boutin, NIST announces first four quantum-resistant cryptographic algorithms, Natl. Inst. Stand. Technol. (2022).

[7] M. Mosca, Cybersecurity in an era with quantum computers: Will we be ready? IEEE Secur. Priv. 16 (5) (2018) 38–41.

[8] Post-quantum cryptography, 2017, https://csrc.nist.gov/projects/post-quantum-cryptography.

[9] National Institute of Standards and Technology (NIST), Mappings of Migration to PQC Project Capabilities to Risk Framework Documents, Tech. Rep. NIST CSWP 48 (Initial Public Draft), NIST, 2025, URL https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.48.ipd.pdf.

[10] Executive Office of the President, Strengthening and promoting innovation in the Nation's Cybersecurity, 2025, URL https://www.federalregister.gov/documents/2025/01/17/2025-01470/strengthening-and-promoting-innovation-in-the-nations-cybersecurity, Published Jan 17, 2025, Executive Order 14144, Federal Register 90(12):6755–6771.

[11] E. Barker, Recommendation for Key Management: Part 1 (Revised)(May 2020 edition) – General, vol. 800, NIST Special Publication, 2020, URL https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final.

[12] National Institute of Standards and Technology, Post-quantum cryptography security (evaluation criteria), 2017, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria).

[13] NIST, FIPS 203: Cryptographic Algorithms and Key Sizes for Post-Quantum Cryptography, 2024, URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf, (Online; Accessed 30 October 2025).

[14] NIST, FIPS 204: CRYSTALS-Dilithium, 2024, URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf, (Online; Accessed 30 October 2025).

[15] NIST, FIPS 205: Falcon, 2024, URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.205.pdf. (Online; Accessed 30 October 2025).

[16] evolutionQ Inc. and Global Risk Institute, Quantum Threat Timeline Report 2024, Tech. Rep., evolutionQ Inc. under license by Global Risk Institute, Toronto, Ontario, Canada, 2024, URL https://www.globalriskinstitute.org/publication/2024-quantum-threat-timeline-report/.

[17] Australian Signals Directorate, Cryptographic roadmap – Preparing for post-quantum cryptography, 2025, Version 1.0, URL https://www.cyber.gov.au/resources-business-and-government/essential-cybersecurity/ism/cybersecurity-guidelines/guidelines-cryptography.

[18] National Security Agency, Commercial National Security Algorithm (CNSA) suite 2.0 and quantum-resistant algorithm transition strategy, 2022, Tech. guidance memo, URL https://media.defense.gov/2022/Sep/07/2003075988/-1/-1/0/CSA_CNSA-2.0_FACT_SHEET.PDF.

[19] EU NIS Cooperation Group, Joint statement on the transition to quantum-resistant cryptography, 2024, URL https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/PQC-joint-statement.pdf?__blob=publicationFile&v=3.

[20] D. Moody, et al., Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process, Nist interagency/internal report (ir) 8547, National Institute of Standards and Technology (NIST), 2020, http://dx.doi.org/10.6028/NIST.IR.8547.

[21] Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI), ANSSI views on the post-quantum cryptography transition: 2023 follow-up, 2023, URL https://www.ssi.gouv.fr/uploads/2023/11/anssi-pqc-transition-2023.pdf.

[22] UK National Cyber Security Centre, Post-quantum cryptography migration timelines and guidance, 2024, URL https://www.ncsc.gov.uk/guidance/pqc-migration-timelines.

[23] Z. Huang, S. Sun, Synthesizing quantum circuits of AES with lower t-depth and less qubits, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2022, pp. 614–644.

[24] M. Grassl, B. Langenberg, M. Roetteler, R. Steinwandt, Applying Grover's algorithm to AES: quantum resource estimates, in: Post-Quantum Cryptography, Springer, 2016, pp. 29–43.

[25] NIST, Post-Quantum Cryptography (PQC): Security (Evaluation Criteria), 2021, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)#FN5. (Online; Accessed 10 February 2022).

[26] T.M. Fernandez-Carames, P. Fraga-Lamas, Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks, IEEE Access 8 (2020) 21091–21116.

[27] T.M. Fernández-Caramés, From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things, IEEE Internet Things J. 7 (7) (2019) 6457–6480.

[28] M. Amy, M. Grassl, B. Langenberg, M. Roetteler, Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3, in: Advances in Cryptology – ASIACRYPT 2016, in: Lecture Notes in Computer Science, Springer, 2016, pp. 317–337, http://dx.doi.org/10.1007/978-3-662-53890-6_12.

[29] NIST, Advanced Encryption Standard (AES), NIST Special Publication, 2001, URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf.

[30] ISO Central Secretary, Information Technology-Security Techniques-Encryption Algorithms-Part 3: Block Ciphers, Standard ISO/IEC 18033-3:2010, International Organization for Standardization, 2010, URL https://www.iso.org/standard/54531.html.

[31] ISO Central Secretary, Banking and Related Financial Services-Key Wrap Using AES, Standard ISO/IEC 20038:2017, International Organization for Standardization, 2017, URL https://www.iso.org/standard/64400.html.

[32] N.M. Elaine Barker, Recommendation for the Triple Data Encryption Algorithm (TDEA) block Cipher, NIST Spec. Publ. 800 (2012) 67.

[33] ISO Central Secretary, Information Technology-Security Techniques-Security Requirements for Cryptographic Modules, Standard ISO/IEC 19790:2012, International Organization for Standardization, 2012, URL https://www.iso.org/standard/52906.html.

[34] M. Matsui, S. Moriai, J. Nakajima, A Description of the Camellia Encryption Algorithm, Request for Comments RFC 3713, RFC Editor, 2004, http://dx.doi.org/10.17487/RFC3713, URL https://www.rfc-editor.org/info/rfc3713.

[35] A. Kato, M. Kanda, S. Kanno, Modes of Operation for Camellia for Use with IPsec, RFC 5529 Request for Comments, RFC Editor, 2009, http://dx.doi.org/10.17487/RFC5529, URL https://www.rfc-editor.org/info/rfc5529.

[36] A. Kato, S. Moriai, M. Kanda, The Camellia Cipher Algorithm and Its Use With IPsec, RFC 4312 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4312, URL https://www.rfc-editor.org/info/rfc4312.

[37] N. Mouha, M. Dworkin, Report on the Block Cipher Modes of Operation in the NIST SP 800-38 Series, Tech. Rep., National Institute of Standards and Technology, 2023.

[38] P. Rogaway, Evaluation of some blockcipher modes of operation, 630, 2011, Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan.

[39] 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, in: Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7, Springer, 2016, pp. 44–63.

[40] B. Bathe, R. Anand, S. Dutta, Evaluation of Grover's algorithm toward quantum cryptanalysis on ChaCha: B. Bathe et al., Quantum Inf. Process. 20 (12) (2021) 394.

[41] Y. Nir, A. Langley, ChaCha20 and Poly1305 for IETF Protocols, RFC 8439 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8439, URL https://www.rfc-editor.org/info/rfc8439.

[42] R. Housley, Using ChaCha20-Poly1305 Authenticated Encryption in the Cryptographic Message Syntax (CMS), RFC 8103 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8103, URL https://www.rfc-editor.org/info/rfc8103.

[43] A. Langley, W.-T. Chang, N. Mavrogiannopoulos, J. Strombergson, S. Josefsson, ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS), RFC 7905 Request for Comments, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC7905, URL https://www.rfc-editor.org/info/rfc7905.

[44] Y. Nir, ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec, RFC 7634 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7634, URL https://www.rfc-editor.org/info/rfc7634.

[45] J. Strombergson, S. Josefsson, Test Vectors for the Stream Cipher RC4, RFC 6229 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6229, URL https://www.rfc-editor.org/info/rfc6229.

[46] ISO Central Secretary, Information Technology-Security Techniques-Encryption Algorithms-Part 4: Stream Ciphers, Standard ISO/IEC 18033-4:2011, International Organization for Standardization, 2011, URL https://www.iso.org/standard/54532.html.

[47] IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 6: Medium access control (MAC) security enhancements, 2004, pp. 1–190, http://dx.doi.org/10.1109/IEEESTD.2004.94585, IEEE Std 802.11i-2004.

[48] ISO Central Secretary, Information Technology-Security Techniques-Lightweight Cryptography-Part 3: Stream Ciphers, Standard ISO/IEC 29192-3:2012, International Organization for Standardization, 2012, URL https://www.iso.org/standard/56426.html.

[49] K. McKay, L. Bassham, M. Sönmez Turan, N. Mouha, Report on Lightweight Cryptography, Tech. Rep., National Institute of Standards and Technology, 2017, URL https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf.

[50] ISO Central Secretary, Information Technology-Automatic Identification and Data Capture Techniques-Part 13: Crypto Suite Grain-128A Security Services for Air Interface Communications, Standard ISO/IEC 29167-13:2015, International Organization for Standardization, 2015, URL https://www.iso.org/standard/60682.html.

[51] M.S. Turan, K. McKay, D. Chang, C. Calik, L. Bassham, J. Kang, J. Kelsey, et al., Status report on the second round of the NIST lightweight cryptography standardization process, Natl. Inst. Stand. Technol. Intern. Rep. 8369 (10.6028) (2021).

[52] M. Boesgaard, M. Vesterager, E. Zenner, A Description of the Rabbit Stream Cipher Algorithm, RFC 4503 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4503, URL https://www.rfc-editor.org/info/rfc4503.

[53] R. Housley, Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP), RFC 3686 Request for Comments, RFC Editor, 2004, http://dx.doi.org/10.17487/RFC3686, URL https://www.rfc-editor.org/info/rfc3686.

[54] K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch, PKCS #1: RSA Cryptography Specifications Version 2.2, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC8017, URL https://www.rfc-editor.org/info/rfc8017.

[55] ISO Central Secretary, IT Security Techniques-Digital Signatures with Appendix-Part 3: Discrete Logarithm Based Mechanisms, Standard ISO/IEC 14888-3:2018, International Organization for Standardization, 2018, URL https://www.iso.org/standard/76382.html.

[56] NIST, PQC standardization process: Announcing four candidates to be standardized, plus fourth round candidates, 2022, https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4.

[57] E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, Tech. Rep., 2018.

[58] S. Sadeghi, V. Chouhan, M. Aldarwbi, A. Ghorbani, A. Chow, R. Burko, Securing financial sector applications in the quantum era: A comprehensive evaluation of NIST's recommended algorithms through use-case analysis, Expert Syst. Appl. (2025) 128243.

[59] D. Adrian, B. Beck, D. Benjamin, D. O'Brien, Advancing our amazing bet on asymmetric cryptography, 2024, Reports ~4% median handshake latency increase on desktop due to ClientHello split, https://blog.chromium.org/2024/05/advancing-our-amazing-bet-on-asymmetric.html.

[60] P. Kampanakis, W. Childs-Klein, The impact of data-heavy, post-quantum TLS 1.3 on the time-to-last-byte of real-world connections, in: MADWeb 2024 (Co-Located with NDSS), 2024, URL https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/the-impact-of-data-heavy-post-quantum.pdf, TTLB increase < 5% on stable high-bw; < 15% on low-bw with 50 KiB+ payload.

[61] M. Sosnowski, et al., The performance of post-quantum TLS 1.3, in: ACM CCS, 2023, URL https://dl.acm.org/doi/10.1145/3624354.3630585.

[62] D.E.E. 3rd, P. Jones, US Secure Hash Algorithm 1 (SHA1), RFC 3174 Request for Comments, RFC Editor, 2001, http://dx.doi.org/10.17487/RFC3174, URL https://www.rfc-editor.org/info/rfc3174.

[63] T. Hansen, D.E.E. 3rd, US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF), RFC 6234 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6234, URL https://www.rfc-editor.org/info/rfc6234.

[64] NIST, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, NIST Special Publication, 2015, URL https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf.

[65] NIST, Secure Hash Standard (SHS), NIST Special Publication, 2015, URL https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.202.pdf.

[66] M.-J.O. Saarinen, J.-P. Aumasson, The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC), RFC 7693 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7693, URL https://www.rfc-editor.org/info/rfc7693.

[67] S. Shen, X. Lee, R.H. Tse, W.W. Kit, P. Yang, The SM3 Cryptographic Hash Function, Internet-Draft draft-sca-cfrg-sm3-01, Internet Engineering Task Force, 2018, URL https://datatracker.ietf.org/doc/draft-sca-cfrg-sm3/01/, Work in Progress.

[68] ISO Central Secretary, Information Technology-Security Techniques-Hash-Functions-Part 4: Hash-Functions Using Modular Arithmetic, Standard ISO/IEC 10118-4:1998, International Organization for Standardization, 1998, URL https://www.iso.org/standard/25429.html.

[69] ISO Central Secretary, IT Security Techniques-Hash-Functions-Part 3: Dedicated Hash-Functions, Standard ISO/IEC 10118-3:2018, International Organization for Standardization, 2018, URL https://www.iso.org/standard/67116.html.

[70] E.B. Barker, A.L. Roginsky, SP 800-131a. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, National Institute of Standards & Technology, 2011.

[71] ISO Central Secretary, Information Security-Cryptographic Techniques Based on Elliptic Curves-Part 5: Elliptic Curve Generation, Standard ISO/IEC 15946-5:2017, International Organization for Standardization, 2017, URL https://www.iso.org/standard/80241.html.

[72] ISO Central Secretary, Information Technology-Security Techniques-Security Requirements for Cryptographic Modules, Standard ISO/IEC 19790:2012, International Organization for Standardization, 2012, URL https://www.iso.org/standard/52906.html.

[73] ISO Central Secretary, Identification Cards-Integrated Circuit Card Programming Interfaces-Part 3: Application Interface, Standard ISO/IEC 24727-3:2008, International Organization for Standardization, 2008, URL https://www.iso.org/standard/43809.html.

[74] ISO Central Secretary, Standard ISO/IEC 29192-6:2019, International Organization for Standardization, 2019, URL https://https://www.iso.org/standard/71116.html.

[75] ISO Central Secretary, Information Security-Message Authentication Codes (Macs)-Part 2: Mechanisms Using a Dedicated Hash-Function, Standard ISO/IEC 9797-2:2021, International Organization for Standardization, 2021, URL https://www.iso.org/standard/75296.html.

[76] ISO Central Secretary, Information Technology-Security Techniques-Message Authentication Codes (Macs)-Part 1: Mechanisms Using a Block Cipher, Standard ISO/IEC 9797-1:2011, International Organization for Standardization, 2011, URL https://www.iso.org/standard/50375.html.

[77] K.R. Glenn, P.-C. Cheng, Test Cases for HMAC-MD5 and HMAC-SHA-1, RFC 2202 Request for Comments, RFC Editor, 1997, http://dx.doi.org/10.17487/RFC2202, URL https://www.rfc-editor.org/info/rfc2202.

[78] D.S.E. Deering, J. McCann, J. Mogul, Path MTU Discovery for IP Version 6, RFC 1981 Request for Comments, RFC Editor, 1996, http://dx.doi.org/10.17487/RFC1981, URL https://www.rfc-editor.org/info/rfc1981.

[79] S. Turner, Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms, RFC 6151 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6151, URL https://www.rfc-editor.org/info/rfc6151.

[80] J. Kelsey, S.-j. Chang, R. Perlner, Nist Special Publication 800-185: Sha-3 Derived Functions: Cshake, KMAC, Tuplehash and Parallelhash, Tech. Rep, National Institute of Standards and Technology, Gaithersburg, MD, 2016.

[81] ANSIx9.62 public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ECDSA), 2005, ANSI.

[82] IEEE standard specifications for public-key cryptography-amendment 1: Additional techniques, 2004, pp. 1–167, http://dx.doi.org/10.1109/IEEESTD.2004.94612, IEEE Std 1363a-2004 (Amendment To IEEE Std 1363-2000).

[83] NIST, Digital Signature Standard (DSS), NIST Special Publication, 2023, URL https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf.

[84] ISO Central Secretary, Information Technology-Security Techniques-Digital Signature Schemes Giving Message Recovery-Part 3: discrete Logarithm Based Mechanisms, Standard ISO/IEC 9796-3:2006, International Organization for Standardization, 2006, URL https://www.iso.org/standard/42228.html.

[85] ISO Central Secretary, Information Technology-Security Techniques-Digital Signatures with Appendix-Part 1: General, Standard ISO/IEC 14888-1:2008, International Organization for Standardization, and International Electrotechnical Commission, 2008, URL https://www.iso.org/standard/44226.html.

[86] T. Polk, S. Turner, R. Housley, D.R.L. Brown, K. Yiu, Updates for RSAES-OAEP and RSASSA-PSS Algorithm Parameters, RFC 5756 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5756, URL https://www.rfc-editor.org/info/rfc5756.

[87] E. Barker, A. Roginsky, NIST special publication 800–131a revision 2: Transitioning the use of cryptographic algorithms and key lengths, 2019.

[88] ANSI X9.30.1 public key cryptography for the financial services industry: Part 1: the Digital Signature Algorithm (DSA), 1997, ANSI.

[89] IEEE standard specifications for public-key cryptography, 2000, pp. 1–228, http://dx.doi.org/10.1109/IEEESTD.2000.92292, IEEE Std 1363-2000.

[90] ISO Central Secretary, Information Technology, Standard ISO/IEC JTC1, International Organization for Standardization, URL https://www.iec.ch/dyn/www/f?p=103:7:411214269599852::::FSP_ORG_ID,FSP_LANG_ID:3387,25.

[91] NIST, Digital signature standard (DSS), 2013, NIST Special Publication, URL https://csrc.nist.gov/pubs/fips/186-4/final.

[92] S. Blake, ANSIx9. 63 overview: Key agreement and key transport using elliptic curve cryptography, 2013, ANSI.

[93] ANSIx9.42 public key cryptography for the financial services industry: Agreement of symmetric keys using discrete logarithm cryptography, 2013, ANSI.

[94] E. Rescorla, Diffie-Hellman Key Agreement Method, RFC 2631 Request for Comments, RFC Editor, 1999, http://dx.doi.org/10.17487/RFC2631, URL https://www.rfc-editor.org/info/rfc2631.
[95] E. Barker, L. Chen, S. Keller, A. Roginsky, A. Vassilev, R. Davis, Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, Tech. Rep., National Institute of Standards and Technology, 2018.
[96] A. Langley, M. Hamburg, S. Turner, Elliptic Curves for Security, RFC 7748 Request for Comments, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC7748, URL https://www.rfc-editor.org/info/rfc7748.
[97] J. Herzog, R. Khazan, Use of Static-Static Elliptic Curve Diffie-Hellman Key Agreement in Cryptographic Message Syntax, RFC 6278 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6278, URL https://www.rfc-editor.org/info/rfc6278.
[98] ISO/IEC 18033-1:2021 information security-encryption algorithms-part 1: General, 2021, ISO.
[99] IEEE standard specifications for public-key cryptography-amendment 1: Additional techniques, 2004, pp. 1–167, http://dx.doi.org/10.1109/IEEESTD.2004.94612, IEEE Std 1363a-2004 (Amendment To IEEE Std 1363-2000).
[100] E. Barker, L. Chen, A. Roginsky, A. Vassilev, R. Davis, S. Simon, SP 800-56B Rev. 1: Recommendation for Pair-Wise Key-Establishment Using Integer Factorization Cryptography, Tech. Rep., National Institute of Standards and Technology, 2018.
[101] E. Barker, A. Roginsky, SP 800-131A Rev.2 Transitioning the Use of Cryptographic Algorithms and Key Lengths, Tech. Rep., National Institute of Standards and Technology, 2018.
[102] F. NIST, 171: Key Management Using ANSI X9. 17, Apr. 27, 1992, Retrieved from Internet: http://securityv.isu.edu/isl/fips171.html, pp. 1–26.
[103] ANSI X9.24-1-2017 retail financial services symmetric key management Part 1: Using symmetric techniques (contains corrigendum), 2017, ANSI.
[104] E.B. Barker, L. Chen, A.R. Regenscheid, M.E. Smid, Sp 800-56b. recommendation for pair-wise key establishment schemes using integer factorization cryptography, 2009, National Institute of Standards & Technology.
[105] E.B. Barker, A.L. Roginsky, Sp 800-131a. transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths, 2011, National Institute of Standards & Technology.
[106] W. Castryck, T. Decru, An efficient key recovery attack on SIDH, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2023, pp. 423–447.
[107] M. Ounsworth, A. Wussler, S. Kousidis, Combiner Function for Hybrid Key Encapsulation Mechanisms (Hybrid KEMs), Internet-Draft draft-ounsworth-cfrg-kem-combiners-04, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-ounsworth-cfrg-kem-combiners/04/, Work in Progress.
[108] R. Barnes, K. Bhargavan, B. Lipp, C.A. Wood, Hybrid Public Key Encryption, RFC 9180 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9180, URL https://www.rfc-editor.org/info/rfc9180.
[109] B. Westerbaan, C.A. Wood, X25519Kyber768Draft00 Hybrid Post-Quantum KEM for HPKE, Internet-Draft draft-westerbaan-cfrg-hpke-xyber768d00-02, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-westerbaan-cfrg-hpke-xyber768d00/02/, Work in Progress.
[110] D. Harkins, Deterministic Nonce-less Hybrid Public Key Encryption, Internet-Draft draft-irtf-cfrg-dnhpke-04, Internet Engineering Task Force, 2024, URL https://datatracker.ietf.org/doc/draft-irtf-cfrg-dnhpke/04/, Work in Progress.
[111] P. Kampanakis, D. Stebila, T. Hansen, Post-quantum Hybrid Key Exchange in SSH, Internet-Draft draft-kampanakis-curdle-ssh-pq-ke-01, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-kampanakis-curdle-ssh-pq-ke/01/, Work in Progress.
[112] B. Westerbaan, D. Stebila, X25519Kyber768Draft00 Hybrid Post-Quantum Key Agreement, Internet-Draft draft-tls-westerbaan-xyber768d00-03, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-tls-westerbaan-xyber768d00/03/, Work in Progress.
[113] D. Stebila, S. Fluhrer, S. Gueron, Hybrid Key Exchange in TLS 1.3, (Internet-Draft draft-ietf-tls-hybrid-design-10) Internet Engineering Task Force, 2024, URL https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/10/, Work in Progress.
[114] M.S. Turan, M.S. Turan, K. McKay, D. Chang, L.E. Bassham, J. Kang, N.D. Waller, J.M. Kelsey, D. Hong, Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process, US Department of Commerce, National Institute of Standards and Technology, 2023.
[115] M.S. Turan, K.A. McKay, J. Kang, J. Kelsey, D. Chang, Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions, NIST Special Publication 800-232, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, 2025, http://dx.doi.org/10.6028/NIST.SP.800-232, URL https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-232.pdf.
[116] C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer, Ascon v1. 2, 5 (6), 2016, p. 7, Submission to the CAESAR Competition.
[117] NIST, Lightweight cryptography, 2022, https://csrc.nist.gov/Projects/lightweight-cryptography/finalists.
[118] B. Mennink, Elephant v2, 2021.
[119] S. Banik, S.K. Pandey, T. Peyrin, Y. Sasaki, S.M. Sim, Y. Todo, GIFT: A small present: Towards reaching the limit of lightweight encryption, in: Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, Springer, 2017, pp. 321–345.
[120] M. Hell, T. Johansson, A. Maximov, W. Meier, J. Sönnerup, H. Yoshida, Grain-128aeadv2-a lightweight AEAD stream cipher, 2021, Submission to NIST LWC Project.
[121] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, T. Unterluggauer, ISAP v2. 0, 2020.
[122] J. Guo, T. Peyrin, A. Poschmann, The PHOTON family of lightweight hash functions, in: Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings 31, Springer, 2011, pp. 222–239.
[123] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, S.M. Sim, The SKINNY family of block ciphers and its low-latency variant MANTIS, in: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36, Springer, 2016, pp. 123–153.
[124] C. Beierle, A. Biryukov, L.C. dos Santos, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov, Q. Wang, A. Biryukov, Schwaemm and esch: lightweight authenticated encryption and hashing using the sparkle permutation family, 2, 2019, NIST Round.
[125] H. Wu, T. Huang, JAMBU lightweight authenticated encryption mode and AES-JAMBU, 2014, CAESAR competition proposal.
[126] J. Daemen, S. Hoffert, M. Peeters, G.V. Assche, R.V. Keer, Xoodyak, a lightweight cryptographic scheme, 2020.
[127] K. Jang, G. Song, H. Kim, H. Kwon, H. Kim, H. Seo, Efficient implementation of PRESENT and GIFT on quantum computers, Appl. Sci. 11 (11) (2021) 4776, http://dx.doi.org/10.3390/app11114776.
[128] R. Anand, A. Maitra, S. Maitra, C.S. Mukherjee, S. Mukhopadhyay, Quantum resource estimation for FSR based symmetric ciphers and related Grover's attacks, in: Progress in Cryptology – INDOCRYPT 2021, Lecture Notes in Computer Science, vol. 13143, Springer, Cham, 2021, pp. 179–198, http://dx.doi.org/10.1007/978-3-030-92518-5_9.
[129] C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer, Ascon v1.2: Lightweight authenticated encryption and hashing, J. Cryptology 34 (3) (2021) 33, http://dx.doi.org/10.1007/s00145-021-09398-9.
[130] A. Jagielski, K. Kanciak, Grover on SPARKLE: Quantum resource estimation for a NIST LWC call finalist, Quantum Inf. Comput. 22 (13–14) (2022) 1132–1143, http://dx.doi.org/10.26421/QIC22.13-14-3.
[131] National Institute of Standards and Technology, Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process, Tech. Rep. NIST IR 8454, National Institute of Standards and Technology, Gaithersburg, MD, 2023.
[132] W.-K. Lee, K. Jang, J. Han, J. Park, J.H. Roh, H. Kim, J. Seo, Efficient implementation of lightweight hash functions on GPU and quantum computers for IoT applications, IEEE Access 10 (2022) 59655–59674, http://dx.doi.org/10.1109/ACCESS.2022.3179755, URL https://ieeexplore.ieee.org/document/9799864.
[133] G. Brassard, P. Høyer, An exact quantum polynomial-time algorithm for Simon's problem, in: Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems, ISTCS'97, IEEE, 1997, pp. 12–23, http://dx.doi.org/10.1109/ISTCS.1997.595153.
[134] S. Aaronson, D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A 70 (5) (2004) 052328, http://dx.doi.org/10.1103/PhysRevA.70.052328.
[135] ISO Central Secretary, Information Technology-Security Techniques-Entity Authentication-Part 1: General, Standard ISO/IEC 9798-1:2010, International Organization for Standardization, 2010, URL https://www.iso.org/standard/53634.html.
[136] ISO Central Secretary, IT Security Techniques-Entity Authentication-Part 2: Mechanisms Using Authenticated Encryption, Standard ISO/IEC 9798-2:2019, International Organization for Standardization, 2019, URL https://www.iso.org/standard/67114.html.
[137] ISO Central Secretary, IT Security Techniques-Entity Authentication-Part 3: Mechanisms Using Digital Signature Techniques, Standard ISO/IEC 9798-3:2019, International Organization for Standardization, 2019, URL https://www.iso.org/standard/67115.html.
[138] ISO Central Secretary, Information Technology-Security Techniques-Entity Authentication-Part 4: Mechanisms Using a Cryptographic Check Function, Standard ISO/IEC 9798-4:1999, International Organization for Standardization, 1999, URL https://www.iso.org/standard/31488.html.
[139] ISO Central Secretary, Information technology-Security techniques-Entity authentication-Part 5: Mechanisms Using Zero-Knowledge Techniques, Standard ISO/IEC 9798-5:2009, International Organization for Standardization, 2009, URL https://www.iso.org/standard/50456.html.
[140] ISO Central Secretary, Information Technology-Security Techniques-Entity Authentication-Part 6: mechanisms Using Manual Data Transfer, Standard ISO/IEC 9798-6:2010, International Organization for Standardization, 2010, URL https://www.iso.org/standard/54529.html.

[141] NIST, Entity Authentication Using Public Key Cryptography, NIST Special Publication, 1997, URL https://csrc.nist.gov/pubs/fips/196/final.

[142] C.W. Kaufman, DASS-Distributed Authentication Security Service, RFC 1507 Request for Comments, RFC Editor, 1993, http://dx.doi.org/10.17487/RFC1507, URL https://www.rfc-editor.org/info/rfc1507.

[143] R. Atkinson, N.M. Haller, On Internet Authentication, RFC 1704 Request for Comments, RFC Editor, 1994, http://dx.doi.org/10.17487/RFC1704, URL https://www.rfc-editor.org/info/rfc1704.

[144] N.M. Haller, The S/KEY One-Time Password System, RFC 1760 Request for Comments, RFC Editor, 1995, http://dx.doi.org/10.17487/RFC1760, URL https://www.rfc-editor.org/info/rfc1760.

[145] D.C. Neuman, S. Hartman, K. Raeburn, T. Yu, The Kerberos Network Authentication Service (V5), RFC 4120 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4120, URL https://www.rfc-editor.org/info/rfc4120.

[146] ISO Central Secretary, Information Technology – Open Systems Interconnection – the Directory: Public-Key and Attribute Certificate Frameworks, Standard ISO/IEC 9594-8, International Organization for Standardization, 2020, URL https://www.iso.org/standard/80325.html.

[147] ISO Central Secretary, Information Technology-Security Techniques-Specification of TTP Services to Support the Application of Digital Signatures, Standard ISO/IEC 15945:2002, International Organization for Standardization, 2002, URL https://www.iso.org/standard/29578.html.

[148] ISO Central Secretary, Public Key Infrastructure for Financial Services-Practices and Policy Framework, Standard ISO/IEC 21188:2018, International Organization for Standardization, 2018, URL https://www.iso.org/standard/63134.html.

[149] American Bankers Association, et al., ANSI X9. 62: The Elliptic Curve Digital Signature Algorithm, Tech. Rep., Technical report. American Bankers Association, 1999.

[150] American Bankers Association, et al., ANSI X9. 63 elliptic curve key agreement and key transport protocols.[on-line], 1999.

[151] B. Kaliski, PKCS #7: Cryptographic Message Syntax Version 1.5, RFC 2315 Request for Comments, RFC Editor, 1998, http://dx.doi.org/10.17487/RFC2315, URL https://www.rfc-editor.org/info/rfc2315.

[152] R. Housley, J. Schaad, B. Kaliski, Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 4055 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4055, URL https://www.rfc-editor.org/info/rfc4055.

[153] K. Moriarty, B. Kaliski, A. Rusch, PKCS #5: Password-Based Cryptography Specification Version 2.1, RFC 8018 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8018, URL https://www.rfc-editor.org/info/rfc8018.

[154] S. Turner, D.R.L. Brown, Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), RFC 5753 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5753, URL https://www.rfc-editor.org/info/rfc5753.

[155] T. Polk, R. Housley, S. Turner, D.R.L. Brown, K. Yiu, Elliptic Curve Cryptography Subject Public Key Information, Request for Comments, (5480) RFC Editor, 2009, http://dx.doi.org/10.17487/RFC5480, URL https://www.rfc-editor.org/info/rfc5480.

[156] P. Kampanakis, Q. Dang, Internet X.509 Public Key Infrastructure: Additional Algorithm Identifiers for RSASSA-PSS and ECDSA Using SHAKEs, RFC 8692 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8692, URL https://www.rfc-editor.org/info/rfc8692.

[157] D.R.L. Brown, Q. Dang, T. Polk, S. Santesson, K. Moriarty, Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA, RFC 5758 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5758, URL https://www.rfc-editor.org/info/rfc5758.

[158] R. Housley, T. Polk, L.E.B. III, Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 3279 Request for Comments, (3279) RFC Editor, 2002, http://dx.doi.org/10.17487/RFC3279, URL https://www.rfc-editor.org/info/rfc3279.

[159] D. Shefanovski, S. Leontiev, Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC 4491 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4491, URL https://www.rfc-editor.org/info/rfc4491.

[160] J. Schaad, H. Prafullchandra, Diffie-Hellman Proof-of-Possession Algorithms, RFC 6955 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC6955, URL https://www.rfc-editor.org/info/rfc6955.

[161] R. Housley, Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF), RFC 9045 Request for Comments, RFC Editor, 2021, http://dx.doi.org/10.17487/RFC9045, URL https://www.rfc-editor.org/info/rfc9045.

[162] S. Turner, Using SHA2 Algorithms with Cryptographic Message Syntax, RFC 5754 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5754, URL https://www.rfc-editor.org/info/rfc5754.

[163] P. Kampanakis, Q. Dang, Use of the SHAKE One-Way Hash Functions in the Cryptographic Message Syntax (CMS), RFC 8702 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8702, URL https://www.rfc-editor.org/info/rfc8702.

[164] A. Medvinsky, M. Hur, Addition of Kerberos Cipher Suites to Transport Layer Security (TLS), RFC 2712 Request for Comments, RFC Editor, 1999, http://dx.doi.org/10.17487/RFC2712, URL https://www.rfc-editor.org/info/rfc2712.

[165] K. Raeburn, Advanced Encryption Standard (AES) Encryption for Kerberos 5, RFC 3962 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC3962, URL https://www.rfc-editor.org/info/rfc3962.

[166] M.J. Jenkins, M. Peck, K.W. Burgin, AES Encryption with HMAC-SHA2 for Kerberos 5, RFC 8009 Request for Comments, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC8009, URL https://www.rfc-editor.org/info/rfc8009.

[167] K. Jaganathan, K. Lauter, L. Zhu, Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT), RFC 5349 Request for Comments, RFC Editor, 2008, http://dx.doi.org/10.17487/RFC5349, URL https://www.rfc-editor.org/info/rfc5349.

[168] S. Josefsson, Using Kerberos Version 5 over the Transport Layer Security (TLS) Protocol, RFC 6251 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6251, URL https://www.rfc-editor.org/info/rfc6251.

[169] G. Hudson, Camellia Encryption for Kerberos 5, RFC 6803 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6803, URL https://www.rfc-editor.org/info/rfc6803.

[170] S. Sorce, T. Yu, Kerberos Authorization Data Container Authenticated by Multiple Message Authentication Codes (MACs), RFC 7751 Request for Comments, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC7751, URL https://www.rfc-editor.org/info/rfc7751.

[171] L. Astrand, L. Zhu, M. Cullen, M. Cullen, G. Hudson, Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) Algorithm Agility, RFC 8636 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8636, URL https://www.rfc-editor.org/info/rfc8636.

[172] S. Rose, M. Larson, D. Massey, R. Austein, R. Arends, DNS Security Introduction and Requirements, RFC 4033 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4033, URL https://www.rfc-editor.org/info/rfc4033.

[173] S. Josefsson, Storing Certificates in the Domain Name System (DNS), RFC 4398 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4398, URL https://www.rfc-editor.org/info/rfc4398.

[174] R. Arends, G. Sisson, D. Blacka, B. Laurie, DNS Security (DNSSEC) Hashed Authenticated Denial of Existence, RFC 5155 Request for Comments, RFC Editor, 2008, http://dx.doi.org/10.17487/RFC5155, URL https://www.rfc-editor.org/info/rfc5155.

[175] O. Kolkman, M. Mekking, R.M. Gieben, DNSSEC Operational Practices, Version 2, RFC 6781 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6781, URL https://www.rfc-editor.org/info/rfc6781.

[176] V. Dolmatov, A. Degtyarev, GOST R 34.10-2012: Digital Signature Algorithm, RFC 7091 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC7091, URL https://www.rfc-editor.org/info/rfc7091.

[177] O. Surý, R. Edmonds, Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC, RFC 8080 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8080, URL https://www.rfc-editor.org/info/rfc8080.

[178] P.E. Hoffman, W. Wijngaards, Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC, RFC 6605 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6605, URL https://www.rfc-editor.org/info/rfc6605.

[179] V. Dolmatov, I. Ustinov, A. Chuprina, Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC, RFC 5933 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5933, URL https://www.rfc-editor.org/info/rfc5933.

[180] J. Jansen, Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC, RFC 5702 Request for Comments, (5702) RFC Editor, 2009, http://dx.doi.org/10.17487/RFC5702, URL https://www.rfc-editor.org/info/rfc5702.

[181] D.E.E. 3rd, RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS), RFC 3110 Request for Comments, RFC Editor, 2001, http://dx.doi.org/10.17487/RFC3110, URL https://www.rfc-editor.org/info/rfc3110.

[182] B. Makarenko, V. Dolmatov, Use of GOST 2012 Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC, Internet-Draft draft-makarenko-gost2012-dnssec-05, Internet Engineering Task Force, 2024, URL https://datatracker.ietf.org/doc/draft-makarenko-gost2012-dnssec/05/, Work in Progress.

[183] H. Lee, T. Kwon, DNSSEC Extension by Using PKIX Certificates, Internet-Draft draft-dnsop-dnssec-extension-pkix-01, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-dnsop-dnssec-extension-pkix/01/, Work in Progress.

[184] C. Zhang, Y. Liu, F. Leng, Q. Zhao, Z. He, SM2 Digital Signature Algorithm for DNSSEC, Internet-Draft draft-cuiling-dnsop-sm2-alg-05, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-cuiling-dnsop-sm2-alg/05/, Work in Progress.

[185] I. Young, L. Johansson, S. Cantor, The Entity Category Security Assertion Markup Language (SAML) Attribute Types, RFC 8409 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8409, URL https://www.rfc-editor.org/info/rfc8409.

[186] J. Howlett, S. Hartman, A. Pérez-Méndez, A RADIUS Attribute, Binding, Profiles, Name Identifier Format, and Confirmation Methods for the Security Assertion Markup Language (SAML), RFC 7833 Request for Comments, RFC Editor, 2016, http://dx.doi.org/10.17487/RFC7833, URL https://www.rfc-editor.org/info/rfc7833.

[187] B. Campbell, C. Mortimore, M.B. Jones, Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants, RFC 7522 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7522, URL https://www.rfc-editor.org/info/rfc7522.

[188] K. Wierenga, E. Lear, S. Josefsson, A Simple Authentication and Security Layer (SASL) and GSS-API Mechanism for the Security Assertion Markup Language (SAML), RFC 6595 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6595, URL https://www.rfc-editor.org/info/rfc6595.

[189] G. Whitehead, et al., Metadata Profile for the OASIS Security Assertion Markup Language (SAML) V1.x. OASIS SSTC, 2005, URL http://www.oasis-open.org/committees/security/.

[190] S. Cantor, et al., Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS SSTC, 2005, URL http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[191] OASIS Committee Specification 01, SAML v2.0 Metadata Profile for Algorithm Support Version 1.0,, 2011, URL http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-v1.0-cs01.odt.

[192] W. Draft, SAML V2. 0 holder-of-key web browser SSO profile version 1.0, 2008.

[193] L. Seitz, Additional OAuth Parameters for Authentication and Authorization for Constrained Environments (ACE), RFC 9201 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9201, URL https://www.rfc-editor.org/info/rfc9201.

[194] B. Campbell, J. Bradley, N. Sakimura, T. Lodderstedt, OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens, RFC 8705 Request for Comments, (8705) RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8705, URL https://www.rfc-editor.org/info/rfc8705.

[195] J. Richer, OAuth 2.0 Token Introspection, RFC 7662 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7662, URL https://www.rfc-editor.org/info/rfc7662.

[196] M.B. Jones, B. Campbell, C. Mortimore, JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants, RFC 7523 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7523, URL https://www.rfc-editor.org/info/rfc7523.

[197] W. Mills, T. Showalter, H. Tschofenig, A Set of Simple Authentication and Security Layer (SASL) Mechanisms for OAuth, RFC 7628 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7628, URL https://www.rfc-editor.org/info/rfc7628.

[198] N. Sakimura, J. Bradley, N. Agarwal, Proof Key for Code Exchange by OAuth Public Clients, RFC 7636 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7636, URL https://www.rfc-editor.org/info/rfc7636.

[199] B. Campbell, C. Mortimore, M.B. Jones, Y.Y. Goland, Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants, RFC 7521 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7521, URL https://www.rfc-editor.org/info/rfc7521.

[200] N. Sakimura, J. Bradley, M.B. Jones, The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR), RFC 9101 Request for Comments, RFC Editor, 2021, http://dx.doi.org/10.17487/RFC9101, URL https://www.rfc-editor.org/info/rfc9101.

[201] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, H. Tschofenig, Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth), RFC 9200 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9200, URL https://www.rfc-editor.org/info/rfc9200.

[202] T. Lodderstedt, M. McGloin, P. Hunt, OAuth 2.0 Threat Model and Security Considerations, RFC 6819 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC6819, URL https://www.rfc-editor.org/info/rfc6819.

[203] J. Richer, M.B. Jones, J. Bradley, M. Machulak, P. Hunt, OAuth 2.0 Dynamic Client Registration Protocol, RFC 7591 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7591, URL https://www.rfc-editor.org/info/rfc7591.

[204] M.B. Jones, D. Hardt, The OAuth 2.0 Authorization Framework: Bearer Token Usage, RFC 6750 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6750, URL https://www.rfc-editor.org/info/rfc6750.

[205] V. Bertocci, JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens, RFC 9068 Request for Comments, RFC Editor, 2021, http://dx.doi.org/10.17487/RFC9068, URL https://www.rfc-editor.org/info/rfc9068.

[206] D. Hardt, The OAuth 2.0 Authorization Framework, RFC 6749 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6749, URL https://www.rfc-editor.org/info/rfc6749.

[207] T. Lodderstedt, S. Dronia, M. Scurtescu, OAuth 2.0 Token Revocation, RFC 7009 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC7009, URL https://www.rfc-editor.org/info/rfc7009.

[208] J. Richer, M.B. Jones, J. Bradley, M. Machulak, OAuth 2.0 Dynamic Client Registration Management Protocol, RFC 7592 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7592, URL https://www.rfc-editor.org/info/rfc7592.

[209] W. Denniss, J. Bradley, OAuth 2.0 for Native Apps, RFC 8252 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8252, URL https://www.rfc-editor.org/info/rfc8252.

[210] M.B. Jones, N. Sakimura, J. Bradley, OAuth 2.0 Authorization Server Metadata, RFC Editor Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8414, URL https://www.rfc-editor.org/info/rfc8414.

[211] W. Denniss, J. Bradley, M.B. Jones, H. Tschofenig, OAuth 2.0 Device Authorization Grant, RFC 8628 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8628, URL https://www.rfc-editor.org/info/rfc8628.

[212] M.B. Jones, A. Nadalin, B. Campbell, J. Bradley, C. Mortimore, OAuth 2.0 Token Exchange, RFC 8693 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8693, URL https://www.rfc-editor.org/info/rfc8693.

[213] Y. Wilson, A. Hingnikar, SAML 2, in: Solving Identity Management in Modern Applications: Demystifying OAuth 2, OpenID Connect, and SAML 2, Springer, 2022, pp. 127–141.

[214] D. Hardt, The Oauth 2.0 Authorization Framework, Tech. Rep., 2012.

[215] A.A. Giron, Migrating applications to post-quantum cryptography: Beyond algorithm replacement, 2023, Cryptology ePrint Archive.

[216] J.A. Salowey, D. McGrew, A. Choudhury, AES Galois Counter Mode (GCM) Cipher Suites for TLS, RFC 5288 Request for Comments, RFC Editor, 2008, http://dx.doi.org/10.17487/RFC5288, URL https://www.rfc-editor.org/info/rfc5288.

[217] M. Badra, Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode, RFC 5487 Request for Comments, RFC Editor, 2009, http://dx.doi.org/10.17487/RFC5487, URL https://www.rfc-editor.org/info/rfc5487.

[218] I. Hajjeh, M. Badra, ECDHE_PSK Cipher Suites for Transport Layer Security (TLS), RFC 5489 Request for Comments, RFC Editor, 2009, http://dx.doi.org/10.17487/RFC5489, URL https://www.rfc-editor.org/info/rfc5489.

[219] M.K. A. Kato, S. Kanno, Camellia Cipher Suites for TLS, RFC 5932 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5932, URL https://www.rfc-editor.org/info/rfc5932.

[220] D. McGrew, D. Bailey, AES-CCM Cipher Suites for Transport Layer Security (TLS), RFC 6655 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6655, URL https://www.rfc-editor.org/info/rfc6655.

[221] D. McGrew, D. Bailey, M. Campagna, R. Dugal, AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, RFC 7251 Request for Comments, RFC Editor, 2014, http://dx.doi.org/10.17487/RFC7251, URL https://www.rfc-editor.org/info/rfc7251.

[222] E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8446, URL https://www.rfc-editor.org/info/rfc8446.

[223] E. Rescorla, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), RFC 5289 Request for Comments, RFC Editor, 2008, http://dx.doi.org/10.17487/RFC5289, URL https://www.rfc-editor.org/info/rfc5289.

[224] A.O. Freier, P. Karlton, P.C. Kocher, The Secure Sockets Layer (SSL) Protocol Version 3.0, RFC 6101 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6101, URL https://www.rfc-editor.org/info/rfc6101.

[225] Y. Sheffer, P. Saint-Andre, T. Fossati, Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), RFC 9325 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9325, URL https://www.rfc-editor.org/info/rfc9325.

[226] Y. Nir, S. Josefsson, M. Pégourié-Gonnard, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier, RFC 8422 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8422, URL https://www.rfc-editor.org/info/rfc8422.

[227] S.V. Smyshlyaev, E. Alekseev, E. Griboedova, A. Babueva, L. Nikiforova, GOST Cipher Suites for Transport Layer Security (TLS) Protocol Version 1.3, RFC 9367 Request for Comments, RFC Editor, 2023, http://dx.doi.org/10.17487/RFC9367, URL https://www.rfc-editor.org/info/rfc9367.

[228] L. Bruckert, J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3, RFC 8734 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8734, URL https://www.rfc-editor.org/info/rfc8734.

[229] J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), RFC 7027 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC7027, URL https://www.rfc-editor.org/info/rfc7027.

[230] N. Mavrogiannopoulos, D.K. Gillmor, Using OpenPGP Keys for Transport Layer Security (TLS) Authentication, RFC 6091 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6091, URL https://www.rfc-editor.org/info/rfc6091.

[231] P. Gutmann, Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), (7366) RFC Editor, 2014, http://dx.doi.org/10.17487/RFC7366, URL https://www.rfc-editor.org/info/rfc7366.

[232] E. Rescorla, H. Tschofenig, N. Modadugu, The Datagram Transport Layer Security (DTLS) Protocol Version 1.3, RFC 9147 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9147, URL https://www.rfc-editor.org/info/rfc9147.

[233] V. Smyslov, Announcing Supported Authentication Methods in IKEv2, Internet-Draft Draft-Ietf-Ipsecme-Ikev2-Auth-Announce-03, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-auth-announce/03/, Work in Progress.
[234] C. Kaufman, P.E. Hoffman, Y. Nir, P. Eronen, T. Kivinen, Internet Key Exchange Protocol Version 2 (IKEv2), RFC 7296 Request for Comments, RFC Editor, 2014, http://dx.doi.org/10.17487/RFC7296, URL https://www.rfc-editor.org/info/rfc7296.
[235] C. Tjhai, M. Tomlinson, G. Bartlett, S. Fluhrer, D.V. Geest, O. Garcia-Morchon, V. Smyslov, Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2), RFC 9370 Request for Comments, RFC Editor, 2023, http://dx.doi.org/10.17487/RFC9370, URL https://www.rfc-editor.org/info/rfc9370.
[236] T. Kivinen, J. Snyder, Signature Authentication in the Internet Key Exchange Version 2 (IKEv2), RFC 7427 Request for Comments, (7427) RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7427, URL https://www.rfc-editor.org/info/rfc7427.
[237] Y. Sheffer, S. Fluhrer, Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2), RFC 6989 Request for Comments, RFC Editor, 2013, http://dx.doi.org/10.17487/RFC6989, URL https://www.rfc-editor.org/info/rfc6989.
[238] D.E. Fu, J. Solinas, IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA), RFC 4754 Request for Comments, (4754) RFC Editor, 2007, http://dx.doi.org/10.17487/RFC4754, URL https://www.rfc-editor.org/info/rfc4754.
[239] J.I. Schiller, Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2), RFC 4307 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4307, URL https://www.rfc-editor.org/info/rfc4307.
[240] P.E. Hoffman, The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE), RFC 4434 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4434, URL https://www.rfc-editor.org/info/rfc4434.
[241] S. murthy, S. Shen, Y. Mao, Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 02 (IKEv2) Protocol, RFC 5930 Request for Comments, RFC Editor, 2010, http://dx.doi.org/10.17487/RFC5930, URL https://www.rfc-editor.org/info/rfc5930.
[242] Y. Nir, Using the Edwards-Curve Digital Signature Algorithm (EdDSA) in the Internet Key Exchange Protocol Version 2 (IKEv2), RFC 8420 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8420, URL https://www.rfc-editor.org/info/rfc8420.
[243] S. Fluhrer, P. Kampanakis, D. McGrew, V. Smyslov, Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security, RFC 8784 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8784, URL https://www.rfc-editor.org/info/rfc8784.
[244] V. Smyslov, Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security, Internet-Draft draft-smyslov-ipsecme-ikev2-qr-alt-08, Internet Engineering Task Force, 2023, URL https://datatracker.ietf.org/doc/draft-smyslov-ipsecme-ikev2-qr-alt/08/, Work in Progress.
[245] S. Frankel, K.R. Glenn, S.G. Kelly, The AES-CBC Cipher Algorithm and Its Use with IPsec, RFC 3602 Request for Comments, RFC Editor, 2003, http://dx.doi.org/10.17487/RFC3602, URL https://www.rfc-editor.org/info/rfc3602.
[246] S. Frankel, H.C. Herbert, The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec, RFC 3566 Request for Comments, RFC Editor, 2003, http://dx.doi.org/10.17487/RFC3566, URL https://www.rfc-editor.org/info/rfc3566.
[247] R. Poovendran, J. Song, J. Lee, The AES-CMAC-96 Algorithm and Its Use with IPsec, RFC 4494 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4494, URL https://www.rfc-editor.org/info/rfc4494.
[248] P.E. Hoffman, Use of Hash Algorithms in Internet Key Exchange (IKE) and IPsec, RFC 4894 Request for Comments, RFC Editor, 2007, http://dx.doi.org/10.17487/RFC4894, URL https://www.rfc-editor.org/info/rfc4894.
[249] P.E. Hoffman, Cryptographic Suites for IPsec, RFC 4308 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4308, URL https://www.rfc-editor.org/info/rfc4308.
[250] L. Law, J. Solinas, Suite B Cryptographic Suites for IPsec, RFC 6379 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6379, URL https://www.rfc-editor.org/info/rfc6379.
[251] J. Viega, D. McGrew, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP), Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4106, URL https://www.rfc-editor.org/info/rfc4106.
[252] M. Kojo, T. Kivinen, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), RFC 3526 Request for Comments, RFC Editor, 2003, http://dx.doi.org/10.17487/RFC3526, URL https://www.rfc-editor.org/info/rfc3526.
[253] P.E. Metzger, W.A. Simpson, IP Authentication Using Keyed MD5, RFC 1828 Request for Comments, RFC Editor, 1995, http://dx.doi.org/10.17487/RFC1828, URL https://www.rfc-editor.org/info/rfc1828.
[254] M. Oehler, K.R. Glenn, HMAC-MD5 IP Authentication with Replay Prevention, RFC 2085 Request for Comments, RFC Editor, 1997, http://dx.doi.org/10.17487/RFC2085, URL https://www.rfc-editor.org/info/rfc2085.
[255] K.R. Glenn, C.R. Madson, The Use of HMAC-MD5-96 within ESP and AH, RFC 2403 Request for Comments, RFC Editor, 1998, http://dx.doi.org/10.17487/RFC2403, URL https://www.rfc-editor.org/info/rfc2403.
[256] K.R. Glenn, C.R. Madson, The Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404 Request for Comments, RFC Editor, 1998, http://dx.doi.org/10.17487/RFC2404, URL https://www.rfc-editor.org/info/rfc2404.
[257] A.D. Keromytis, N. Provos, The Use of HMAC-RIPEMD-160-96 within ESP and AH, RFC 2857 Request for Comments, RFC Editor, 2000, http://dx.doi.org/10.17487/RFC2857, URL https://www.rfc-editor.org/info/rfc2857.
[258] S. Frankel, S.G. Kelly, Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec, RFC 4868 Request for Comments, RFC Editor, 2007, http://dx.doi.org/10.17487/RFC4868, URL https://www.rfc-editor.org/info/rfc4868.
[259] K. Burgin, M. Peck, Suite B Profile for Internet Protocol Security (IPsec), RFC 6380 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6380, URL https://www.rfc-editor.org/info/rfc6380.
[260] C.M. Lonvick, T. Ylonen, The Secure Shell (SSH) Protocol Architecture, RFC 4251 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4251, URL https://www.rfc-editor.org/info/rfc4251.
[261] M.D. Baushke, d. bider, SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol, RFC 6668 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6668, URL https://www.rfc-editor.org/info/rfc6668.
[262] P. Gutmann, A Pre-Authentication Mechanism for SSH, Internet-Draft draft-gutmann-ssh-preauth-00, Internet Engineering Task Force, 2022, URL https://datatracker.ietf.org/doc/draft-gutmann-ssh-preauth/00/, Work in Progress.
[263] M.D. Baushke, Key Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH), RFC 9142 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9142, URL https://www.rfc-editor.org/info/rfc9142.
[264] O. Surý, Use of the SHA-256 Algorithm with RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA) in SSHFP Resource Records, RFC 6594 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6594, URL https://www.rfc-editor.org/info/rfc6594.
[265] d. bider, Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol, RFC 8332 Request for Comments, RFC Editor, 2018, http://dx.doi.org/10.17487/RFC8332, URL https://www.rfc-editor.org/info/rfc8332.
[266] B. Harris, L. Velvindron, Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol, RFC 8709 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8709, URL https://www.rfc-editor.org/info/rfc8709.
[267] A. Adamantiadis, S. Josefsson, M.D. Baushke, Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448, RFC 8731 Request for Comments, RFC Editor, 2020, http://dx.doi.org/10.17487/RFC8731, URL https://www.rfc-editor.org/info/rfc8731.
[268] M.D. Baushke, More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH), RFC 8268 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8268, URL https://www.rfc-editor.org/info/rfc8268.
[269] K. Igoe, Suite B Cryptographic Suites for Secure Shell (SSH), RFC 6239 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6239, URL https://www.rfc-editor.org/info/rfc6239.
[270] C.M. Lonvick, T. Ylonen, The Secure Shell (SSH) Authentication Protocol, RFC 4252 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4252, URL https://www.rfc-editor.org/info/rfc4252.
[271] B. Harris, RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol, RFC 4252 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4432, URL https://www.rfc-editor.org/info/rfc4432.
[272] C. Namprempre, T. Kohno, M. Bellare, The Secure Shell (SSH) Transport Layer Encryption Modes, RFC 4344 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4344, URL https://www.rfc-editor.org/info/rfc4344.
[273] J.A. Salowey, V. Welch, J. Hutzelman, J. Galbraith, Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol, RFC 4462 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4462, URL https://www.rfc-editor.org/info/rfc4462.
[274] S.J. Lunt, FTP Security Extensions, RFC 2228 Request for Comments, RFC Editor, 1997, http://dx.doi.org/10.17487/RFC2228, URL https://www.rfc-editor.org/info/rfc2228.
[275] M. Allman, S. Ostermann, FTP Security Considerations, Request for Comments, RFC Editor, 1999, http://dx.doi.org/10.17487/RFC2577, URL https://www.rfc-editor.org/info/rfc2577.
[276] C.M. Lonvick, T. Ylonen, The Secure Shell (SSH) Transport Layer Protocol, RFC 4253 Request for Comments, RFC Editor, 2006, http://dx.doi.org/10.17487/RFC4253, URL https://www.rfc-editor.org/info/rfc4253.
[277] P. Ford-Hutchinson, Securing FTP with TLS, RFC 4217 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4217, URL https://www.rfc-editor.org/info/rfc4217.
[278] J. Schaad, B. Ramsdell, S. Turner, Secure/multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification, Tech. Rep., 2019.
[279] J. Schaad, B.C. Ramsdell, S. Turner, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification, RFC 8551 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8551, URL https://www.rfc-editor.org/info/rfc8551.

[280] J. Schaad, Enhanced Security Services for S/MIME, Internet-Draft draft-ietf-smime-rfc2634-update-00, Internet Engineering Task Force, 2004, URL https://datatracker.ietf.org/doc/draft-ietf-smime-rfc2634-update/00/, Work in Progress.
[281] J. Schaad, Experiment: Hash Functions with Parameters in the Cryptographic Message Syntax (CMS) and S/MIME, RFC 6210 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6210, URL https://www.rfc-editor.org/info/rfc6210.
[282] A. Kato, S. Moriai, Use of the Camellia Encryption Algorithm in Cryptographic Message Syntax (CMS), RFC 3657 Request for Comments, RFC Editor, 2004, http://dx.doi.org/10.17487/RFC3657, URL https://www.rfc-editor.org/info/rfc3657.
[283] D.K. Gillmor, S/MIME Example Keys and Certificates, RFC 9216 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9216, URL https://www.rfc-editor.org/info/rfc9216.
[284] A. Melnikov, S/MIME Signature Verification Extension to the JSON Meta Application Protocol (JMAP), RFC 9219 Request for Comments, RFC Editor, 2022, http://dx.doi.org/10.17487/RFC9219, URL https://www.rfc-editor.org/info/rfc9219.
[285] A. Melnikov, Extensions to Automatic Certificate Management Environment for End-User S/MIME Certificates, RFC 8823 Request for Comments, RFC Editor, 2021, http://dx.doi.org/10.17487/RFC8823, URL https://www.rfc-editor.org/info/rfc8823.
[286] J. Schaad, B.C. Ramsdell, S. Turner, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling, RFC 8550 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8550, URL https://www.rfc-editor.org/info/rfc8550.
[287] B. Campbell, R. Housley, SIP-Based Messaging with S/MIME, RFC 8591 Request for Comments, RFC Editor, 2019, http://dx.doi.org/10.17487/RFC8591, URL https://www.rfc-editor.org/info/rfc8591.
[288] P.E. Hoffman, J. Schlyter, Using Secure DNS to Associate Certificates with Domain Names for S/MIME, RFC 8162 Request for Comments, RFC Editor, 2017, http://dx.doi.org/10.17487/RFC8162, URL https://www.rfc-editor.org/info/rfc8162.
[289] L. Cailleux, C. Bonatti, Securing Header Fields with S/MIME, RFC 7508 Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/RFC7508, URL https://www.rfc-editor.org/info/rfc7508.
[290] J. Solinas, R. Housley, Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME), RFC 6318 Request for Comments, RFC Editor, 2011, http://dx.doi.org/10.17487/RFC6318, URL https://www.rfc-editor.org/info/rfc6318.
[291] J. Schaad, S/MIME Capabilities for Public Key Definitions, RFC 6664 Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6664, URL https://www.rfc-editor.org/info/rfc6664.
[292] R. Housley, Object Identifier Registry for the S/MIME Mail Security Working Group, RFC 7107 Request for Comments, RFC Editor, 2014, http://dx.doi.org/10.17487/RFC7107, URL https://www.rfc-editor.org/info/rfc7107.
[293] S. Santesson, X.509 Certificate Extension for Secure/Multipurpose Internet Mail Extensions (S/MIME) Capabilities, RFC 4262 Request for Comments, RFC Editor, 2005, http://dx.doi.org/10.17487/RFC4262, URL https://www.rfc-editor.org/info/rfc4262.
[294] A. Melnikov, Authentication-Results Registration for S/MIME Signature Verification, RFC 7281 Request for Comments, RFC Editor, 2014, http://dx.doi.org/10.17487/RFC7281, URL https://www.rfc-editor.org/info/rfc7281.
[295] J. Peterson, S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP), RFC 3853 Request for Comments, RFC Editor, 2004, http://dx.doi.org/10.17487/RFC3853, URL https://www.rfc-editor.org/info/rfc3853.
[296] P.E. Hoffman, C. Bonatti, A. Eggen, Securing X.400 Content with Secure/Multipurpose Internet Mail Extensions (S/MIME), RFC 3854 Request for Comments, RFC Editor, 2004, http://dx.doi.org/10.17487/RFC3854, URL https://www.rfc-editor.org/info/rfc3854.
[297] European Telecommunications Standards Institute, Quantum safe cryptography and security, 2015, URL https://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf.
[298] N. Bindel, U. Herath, M. McKague, D. Stebila, Transitioning to a quantum-resistant public key infrastructure, in: International Workshop on Post-Quantum Cryptography, Springer, 2017, pp. 384–405.
[299] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, T. Grasmeyer, On the state of crypto-agility, 2023, URL https://eprint.iacr.org/2023/487, https://eprint.iacr.org/2023/487, Cryptology ePrint Archive, Paper 2023/487.
[300] M.J. Dworkin, SP 800-38e. Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, National Institute of Standards & Technology, 2010.
[301] M. Dworkin, et al., Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption, vol. 800, NIST Special Publication, 2016, p. 38G.
[302] M.J. Dworkin, SP 800-38B. Recommendation For Block Cipher Modes of Operation: The CMAC Mode for Authentication, National Institute of Standards & Technology, 2005.
[303] M.J. Dworkin, SP 800-38c. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, National Institute of Standards & Technology, 2004.
[304] M.J. Dworkin, SP 800-38d. Recommendation for Block Cipher Modes of Operation: Galois/counter Mode (GCM) and GMAC, National Institute of Standards & Technology, 2007.
[305] B. Crow, I. Widjaja, J. Kim, P. Sakai, IEEE 802.11 Wireless Local Area networks, IEEE Commun. Mag. 35 (9) (1997) 116–126, http://dx.doi.org/10.1109/35.620533.
[306] S.K. (NIST), Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures, vol. 800, NIST Special Publication, 2012, p. 20.
[307] M. Dworkin, NIST Special Publication 800-38F December 2012, vol. 800, NIST Special Publication, 2012, p. 38F.
[308] NCC Group, Post-Quantum Cryptography Overview, Tech. Rep., NCC Group, 2016, URL https://www.nccgroup.com/media/swdbffg3/_ncc-group-cryptographic-services.pdf, Section: Post-quantum algorithm overview. States that current symmetric ciphers with 256-bit keys, such as AES-256 and ChaCha20, are believed to be quantum-resistant.
[309] G. Paul, S. Maitra, RC4 Stream Cipher and Its Variants, CRC Press, 2011.
[310] P. Ekdahl, T. Johansson, A new version of the stream cipher SNOW, in: Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002 St. John's, Newfoundland, Canada, August 15–16, 2002 Revised Papers 9, Springer, 2003, pp. 47–61.
[311] C. De Canniere, B. Preneel, Trivium, in: New Stream Cipher Designs: The ESTREAM Finalists, Springer, 2008, pp. 244–266.
[312] L. Jiao, Y. Hao, D. Feng, Stream cipher designs: a review, Sci. China Inf. Sci. 63 (2020) 1–25.
[313] M. Hell, T. Johansson, W. Meier, J. Sönnerup, H. Yoshida, An AEAD variant of the grain stream cipher, in: International Conference on Codes, Cryptology, and Information Security, Springer, 2019, pp. 55–71.