Black Sea Journal of Engineering and Science
doi: 10.34248/bsengineering.1670109

Open Access Journal
e-ISSN: 2619 – 8991

Research Article
Volume 8 - Issue 6: 1841-1855 / November 2025

EVALUATION OF PASSWORD HASHING COMPETITION FINALISTS: PERFORMANCE, SECURITY, COMPLIANCE MAPPING, AND POST-QUANTUM READINESS

Erdem ULUTAS[1]*, Baris CELIKTAS[1]

[1]Işık University, Institute of Graduate Students, Department of Computer Science Engineering, 34398, İstanbul, Türkiye

Abstract: Password hashes and key derivation functions (KDFs) are central to authentication and cryptographic security schemes crafted to defend user credentials from brute-force attacks and unauthorized access. Password hashing algorithms, for example PBKDF2, bcrypt, or scrypt, are very popular today, but are lacking in the face of modern hardware acceleration, parallel processing, and advanced cryptanalytic attacks. To contest these shortcomings, the Password Hashing Competition (PHC) was started in 2013 and had 22 candidates for functions for hashing passwords. After thorough evaluation, 9 finalists were selected based on how secure, fast, memory-friendly, flexible, and efficient these functions were. This study evaluates the nine PHC finalists—Argon2, battcrypt, Catena, Lyra2, MAKWA, Parallel, POMELO, Pufferfish, and yescrypt—through survey findings and performance benchmarks. We have evaluated these functions from an architectural standpoint and studied their security features, memory hardness, performance trade-off, and practical usage. We also compare these finalists with traditional password hashing functions to highlight their advantages and limitations. We also investigate the post-quantum assumption for password hashing – the effectiveness of these functions against quantum assaults, their position in a new cryptography set, and the role of peppering as an additional security measure. In addition, we perform a comprehensive compliance mapping of the PHC finalists against major global standards and regulations such as NIST SP 800-63B, OWASP ASVS, PCI DSS, GDPR, KVKK, and ISO/IEC 27001, highlighting their practical suitability for secure deployment in regulated environments. Finally, we provide usage recommendations for these functions for web authentication, KDFs, and embedded platforms. This paper serves as a reference for researchers, developers, and security engineers, while also introducing a compliance-aware, post-quantum-ready framework that bridges cryptographic design with regulatory and deployment needs.

Keywords: Password hashing, Key derivation, Security, Performance, Quantum resistance, Compliance

***Corresponding author:** Işık University, Institute of Graduate Students, Department of Computer Science Engineering, 34398, İstanbul, Türkiye
E mail: 23sibe5003@isik.edu.tr (E. ULUTAS)
Erdem ULUTAS https://orcid.org/0009-0007-4263-5396
Baris CELIKTAS https://orcid.org/0000-0003-2865-6370

Received: April 04, 2025
Accepted: September 26, 2025
Published: November 15, 2025

Cite as: Ulutas E, Celiktas B. 2025. Evaluation of password hashing competition finalists: performance, security, compliance mapping, and post-quantum readiness. BSJ Eng Sci, 8(6): 1841-1855.

1. Introduction

Password hashing forms a critical component of modern security systems, where it plays a fundamental role in authentication, session management and storing secrets. In authentication systems, hashed passwords provide a layer of protection even if the database is compromised (Yao and Yin, 2005).

Despite improvements in password hashing, several vulnerabilities persist, including brute-force, dictionary, and rainbow table attacks (Luo et al., 2021). While widely used, cryptographic hash functions like MD5, SHA-1, and SHA-256 are designed to be fast for integrity verification, which paradoxically makes them attractive targets for attackers who can exploit their speed in large-scale password guessing attacks (Andrade et al., 2016). Moreover, their lack of integrated salting mechanisms and memory hardness makes them susceptible to precomputed attacks (Álvarez and Zamora, 2017). Incorporating a pepper into password storage enhances security by mitigating the risk of offline attacks, as it remains unknown to attackers even if the password database is compromised (Blocki and Sridhar, 2016). Side-channel attacks, such as timing attacks, pose additional risks by leaking sensitive cryptographic operations through observable system behaviors (Alwen et al., 2018).

KDFs also convert weak user passwords into strong cryptographic keys, enhancing encryption and secure communication protocols (Grassi et al., 2023). Current KDFs, such as PBKDF2, bcrypt, and scrypt are vulnerable to hardware-level optimizations that reduce the computational cost of password cracking. PBKDF2, for instance, is not memory-hard and thus is vulnerable to hardware-accelerated brute-force attacks (Alwen et al., 2018). While bcrypt does support different computational overheads, it is nonetheless subject to field-programmable gate array (FPGA) optimizations (Luo et al., 2021). Even scrypt, which offers immunity from parallel attacks, is still susceptible to certain memory-based attacks (Choe et al., 2019).

To address these limitations, the PHC was launched in

2013 as a global initiative to discover more resilient password hashing schemes. A total of 22 candidates were submitted and assessed based on criteria such as security, memory hardness, performance, and implementation flexibility (Aumasson, 2013). Nine finalists—Argon2, battcrypt, Catena, Lyra2, MAKWA, Parallel, POMELO, Pufferfish, and yescrypt—emerged from the process, offering significant advancements in brute-force resistance and practical deployability (Hatzivasilis et al., 2015).

This study provides an extensive review and benchmarking study of the PHC finalists against their architectural designs, security aspects, and applicability in real-world scenarios. Furthermore, it compares these functions against well-known password hashing algorithms, detailing their weaknesses and strengths. Post-quantum aspects are briefly considered and discussed further in subsequent sections. Finally, this study makes concrete recommendations for various security applications, including web authentication, cryptographic key derivation, and embedded system security.

As advancements in quantum computing continue to challenge cryptographic security, this study extends the analysis of PHC finalists by discussing their resilience against quantum threats.

The key contributions of this study are as follows:

- A comparative evaluation of the nine PHC finalists in terms of security, performance, compliance, and real-world applicability.
- An analysis of the robustness of these functions against advanced attack vectors, including graphics processing unit (GPU) parallelization, side-channel attacks, and hardware optimizations.
- A comprehensive compliance mapping of the PHC finalists against global standards and regulations such as NIST SP 800-63B, OWASP ASVS, PCI DSS, GDPR, KVKK, and ISO/IEC 27001.
- A discussion on the potential impact of quantum computing on password hashing, with an assessment of the quantum resistance of each PHC finalist.
- Practical guidance for developers, researchers, and organizations on selecting the most suitable password hashing function based on specific security and performance requirements.

This paper is organized as follows. Section 2 reviews existing related works on KDFs, identifying their strengths and limitations. Section 3 provides essential background and preliminary concepts, including password hashing, KDFs, salting, peppering, and memory-hard functions (MHFs). Section 4 outlines relevant cryptographic standards and regulatory frameworks that influence the implementation of secure password storage. Section 5 presents the core results of this study, including performance benchmarking, security analysis, and compliance evaluation of PHC finalists. Finally, this work is concluded in Section 6 with a summary of the main conclusions and recommendations for further study.

2. Literature Review

This section ensures a comprehensive review of existing KDF research, integrating insights from prior studies to establish the foundation for evaluating PHC finalists.

2.1. Overview of KDF Research

The research on key KDFs has evolved significantly to address security challenges in various domains, including password authentication, blockchain security (Wang et al., 2018), and mobile encryption (Lu et al., 2016). To better illustrate the current state of the literature, Table 1 summarizes key works across various application areas of KDFs, emphasizing their findings and limitations. The studies explore various aspects of KDFs, including user-centric key management, cryptographic security, resistance to side-channel attacks, and applications in blockchain-based security.

A recurring theme in KDF research is the trade-off between security and performance, particularly in constrained environments such as mobile or embedded systems. For instance, studies on memory-hard KDFs highlight the necessity of increased computational cost to mitigate brute-force and near-data processing attacks (Choe et al., 2019). However, these methods often pose implementation challenges, such as high resource consumption, making them unsuitable for constrained environments like mobile devices or cloud computing platforms. In addition, PRNG-based KDFs have also been studied, though their efficiency and real-world viability remain uncertain due to a lack of implementation-focused studies (McGinthy and Michaels, 2019).

Additionally, security evaluations in blockchain-based hashing schemes and password-based cryptographic functions indicate gaps in real-world deployment scenarios. Some studies have proposed improvements to authentication schemes using anonymity features and cryptographic primitives, such as PBKDF2, to enhance user privacy and security (Saad et al., 2016). However, these solutions require further empirical validation and large-scale implementation to assess their robustness against advanced threats. Overall, while significant progress has been made in KDF research, future studies should focus on optimizing performance while ensuring security across diverse application environments.

Beyond individual algorithm-focused studies, several survey and comparison works have shaped the broader understanding of password hashing functions. For example, Hatzivasilis et al. (2015) conducted a survey and benchmark analysis of PBKDF2, bcrypt, and scrypt, while Forler et al. (2015) and Wetzels (2016) provided overviews of PHC candidates with emphasis on their resistance against emerging attacks.

Hatzivasilis (2017) further assessed the state of password-hashing research, and Alwen et al. (2018) analyzed the theoretical memory-hardness of data-independent constructions.

Table 1. Summary of previous works on KDF

Reference	Focus Area	Key Insight	Limitation
Shen et al. (2025)	NIST KDF standards analysis	Critically examines the security of NIST's KDF constructions (SP 800-108), particularly HMAC/CMAC/KMAC-based derivations, identifying subtle structural assumptions and potential misuse risks.	Focused primarily on theoretical and structural vulnerabilities; lacks empirical validation or recommendations for mitigation in deployment.
Backendal et al. (2025)	Password-Based Cryptography	Analyzed the security of KDFs when salts are absent, showing how unsalted derivations are vulnerable to large-scale precomputation and rainbow table attacks.	Applicability limited, since most practical standards mandate salting; primarily a theoretical security analysis.
Tran et al. (2024)	User-Centric Key Management	Proposed a cloud encryption scheme using a KDF for key derivation.	Limited evaluation on large-scale cloud environments.
Clark and Seamons (2022)	Password and Cryptographic Security	Explored the theoretical limits of password-based authentication schemes.	Focused on theoretical aspects rather than practical implementations.
Kodwani et al. (2021)	Password-Based Cryptography	Evaluated security aspects of KDFs in password-based authentication.	Did not include performance benchmarks for different KDF implementations.
Lata and Bansal (2021)	Side-Channel Attack Resistance	Proposed secure KDFs resistant to timing-based side-channel attacks.	Limited discussion on real-world deployment challenges.
Luo et al. (2021)	Blockchain-Based Cybersecurity	Introduced a memory-hard KDF for blockchain-based security applications.	High computational cost for resource-constrained devices.
Choe et al. (2019)	Memory-Hard KDFs	Analyzed scrypt's vulnerability to near-data processing attacks.	Did not propose countermeasures for the identified vulnerabilities.
McGinthy and Michaels (2019)	PRNG-Based KDFs	Investigated improvements in key derivation using pseudo-random number generators.	Focused on theoretical security rather than implementation efficiency.
Alwen et al. (2018)	Memory-Hard KDFs	Examined the memory-hardness of password hashing schemes.	Lacked empirical testing on various hardware architectures.
Wang et al. (2018)	Blockchain Hashing Security	Studied the security criteria of hash functions used in blockchain environments.	Did not evaluate KDFs outside the blockchain context.
Álvarez and Zamora (2016)	Spritz-Based KDFs	Investigated using the Spritz cipher as a password-based KDF.	Limited performance evaluation on modern hardware.
Lu et al. (2016)	Mobile Encryption Security	Studied efficient storage encryption for mobile devices using KDFs.	Did not compare KDF performance across different mobile platforms.
Saad et al. (2016)	Secure Authentication	Proposed an anonymous authentication scheme using a password-based KDF.	Lacked large-scale implementation results.
Forler et al. (2015)	Password Hashing Competition	Evaluated the candidates of the PHC, including Argon2.	Focused on competition outcomes rather than real-world adoption.
Hatzivasilis et al. (2015)	Password Hashing Security	Conducted a survey and benchmark analysis of KDFs used in password hashing.	Did not include post-quantum cryptographic considerations.
Aumasson (2013)	Future of Password Hashing	Discussed advancements in password hashing techniques and KDFs.	Predominantly theoretical analysis without implementation details.
Yao and Yin (2005)	Key Derivation Security	Provided foundational security analysis for password-based KDFs.	Did not anticipate modern hardware acceleration attacks.

McGinthy and Michaels (2019) compared PRNG-based KDFs with HKDF, highlighting trade-offs in efficiency and adoption. These studies deliver valuable insights but generally focus on classical algorithms or isolated PHC candidates. In contrast, our work extends this line of research by offering a consolidated evaluation of all PHC finalists, enriched with compliance mapping against major standards and post-quantum security considerations.

2.2. Limitations and Challenges in Existing KDFs (PHC Finalists)

The PHC finalists represent a selection of the most robust and secure password hashing and KDF schemes available. However, despite their advancements in security, each of these finalists has inherent limitations and challenges that affect their performance, applicability, and overall effectiveness. The primary concerns involve memory usage, computational complexity, security validation, and adaptability in various environments. Table 2 summarizes limitations and challenges in existing PHC finalist KDFs.

One of the major challenges faced by PHC finalists is the balance between security and efficiency. Memory-intensive schemes such as Argon2, Lyra2, and POMELO can utilize gigabytes of memory, making them highly resistant to GPU and ASIC attacks but less feasible for resource-constrained environments (Luo et al., 2021). In contrast, schemes like MAKWA, Parallel, and yescrypt require minimal memory but depend on computational hardness, which can be exploited by specialized hardware to some extent (Luo et al., 2021). Moreover, some schemes, such as POMELO, are not designed as KDFs and thus do not provide the same level of security for key derivation applications (Hatzivasilis et al., 2015).

A further limitation involves the degree of formal security validation available for certain PHC finalists. While Catena and Lyra2 have undergone extensive security analysis, others such as battcrypt, Parallel, and yescrypt do not have formal security proofs to support their claims. Pufferfish, another finalist, has not been fully validated for its security properties, which introduces uncertainty regarding its reliability in adversarial scenarios. Additionally, POMELO has been found to generate lower randomness in its outputs, making it unsuitable as a strong KDF (Hatzivasilis et al., 2015).

Overall, while PHC finalists represent a significant advancement in password hashing and KDFs, each of them has specific trade-offs with respect to memory usage, computational overhead, and security guarantee. The ideal choice of a KDF must be in line with contextual restrictions and objectives of the target system—either minimizing resource usage, side-channel attack resistance, or post-quantum resistance. Future developments in this field should focus on optimizing security while ensuring efficiency and adaptability in various deployment scenarios.

Building on prior surveys that benchmarked PHC finalists primarily on performance and security (e.g., Hatzivasilis et al., 2015), our study advances the discussion by introducing three novel contributions. First, it provides a compliance-aware evaluation aligned with OWASP ASVS, NIST SP 800-63B, GDPR, and PCI DSS. Second, it examines post-quantum resilience under Grover's and Shor's algorithms, offering deployment recommendations under quantum threat models. Third, it delivers a visual decision matrix that bridges theoretical properties with applied security engineering. Collectively, these contributions position this work as a forward-looking reference that integrates standards, post-quantum considerations, and practical decision-making guidance.

Table 2. Limitations and challenges in existing KDFs (PHC Finalists)

KDF	Limitation	Challenge
Argon2	High memory consumption when configured for security	Requires careful tuning of parameters to balance security and performance
battcrypt	Simplified version of scrypt, lacks full hash upgrade independent from the user (HUIU) support	Designed primarily for server-side applications, limiting versatility
Catena	Moderate memory consumption compared to scrypt	Needs careful parameter selection to avoid efficiency bottlenecks
Lyra2	Large memory footprint	Potential efficiency concerns in resource-constrained environments
MAKWA	Computationally expensive due to large-number arithmetic	Complex design, requires offline hash upgrade for security
Parallel	Computational hardness but no RAM-hardness	Not as thoroughly analyzed for security as some other candidates
POMELO	Does not function as a secure KDF due to lower randomness	May not be suitable for applications requiring strong key derivation
Pufferfish	Security properties not fully validated	Slightly less efficient compared to other memory-hard KDFs
yescrypt	Based on scrypt but security proofs are incomplete	Balancing memory-hardness and computational performance

3. Preliminaries

The design and evaluation of password hashing schemes cannot be fully understood without examining the underlying cryptographic features that determine their effectiveness. This section introduces the foundational concepts—hashing, key derivation functions (KDFs), salting, peppering, memory-hard functions (MHFs), and post-quantum considerations—while clarifying why these features are essential and how they are applied in subsequent analysis of PHC finalists.

3.1. Password Hashing Overview

Password hashing transforms a plaintext password into a fixed-length, irreversible output (Aumasson, 2013). Unlike encryption, hashing is one-way and cannot be inverted, making it suitable for password storage. Its importance lies in mitigating credential exposure: even if a database is leaked, the attacker faces the computational challenge of recovering the original passwords (Lu et al., 2016).

In authentication systems, stored password hashes are compared with newly hashed user input, rather than retrieving the original password. Encryption, however, is more suitable for protecting retrievable data, such as stored user information (OWASP Password Storage Cheat Sheet, 2025). Later sections evaluate PHC finalists in terms of their hashing design choices, particularly resistance to brute-force and preimage attacks.

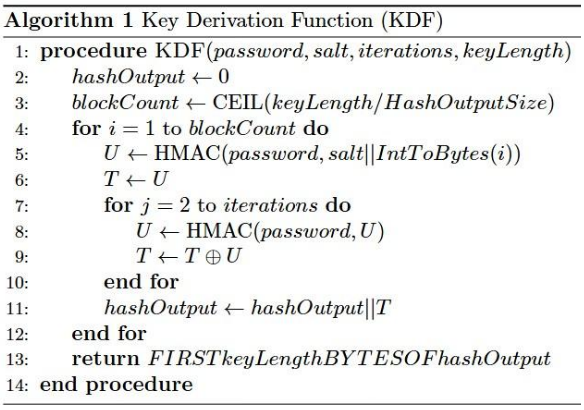
3.2. Key Derivation Functions (KDFs)

KDFs strengthen weak, user-chosen passwords into cryptographic keys by applying repeated hashing, salting, and sometimes memory-hardness. Widely used KDFs including PBKDF2 (RFC 6070, 2011) and bcrypt (Provos and Mazieres, 1999), which rely on multiple iterations but lacking memory hardness. Argon2 (RFC 9106, 2021) and Lyra2 (Simplicio et al., 2014) includes memory-hardness to strengthen security.

The work factor in KDFs determines the number of iterations applied to a password. Increasing the work factor enhances security by making brute-force and dictionary attacks computationally expensive (Clark and Seamons, 2022). However, choosing an excessively high work factor can degrade system performance, potentially leading to denial-of-service (DoS) risks if authentication requests consume excessive server resources (Hatzivasilis et al., 2015).

To address evolving security threats, modern KDFs, such as Argon2 and yescrypt, incorporate adjustable work factors. These settings allow systems to gradually increase the computational cost over time. Rehashing passwords upon authentication ensures compatibility while maintaining security (Clark and Seamons, 2022). It is recommended to periodically reevaluate the work factors to counter advances in computational power (OWASP Password Storage Cheat Sheet, 2025).

To illustrate this process, Figure 1 shows a simplified pseudocode representation of a generic KDF algorithm, outlining key stretching with salt, iteration loops, and optional peppering.



Algorithm 1 Key Derivation Function (KDF)

```
procedure KDF($password, salt, iterations, keyLength$)
    $hashOutput \leftarrow 0$
    $blockCount \leftarrow$ CEIL($keyLength/HashOutputSize$)
    for $i = 1$ to $blockCount$ do
        $U \leftarrow$ HMAC($password, salt||IntToBytes(i)$)
        $T \leftarrow U$
        for $j = 2$ to $iterations$ do
            $U \leftarrow$ HMAC($password, U$)
            $T \leftarrow T \oplus U$
        end for
        $hashOutput \leftarrow hashOutput||T$
    end for
    return $FIRSTkeyLengthBYTESOFhashOutput$
end procedure
```

Figure 1. Pseudocode implementation of a generic KDF.

It demonstrates how KDFs operate internally to transform user passwords into secure cryptographic keys. This visual model complements the digest outputs shown in Table 3, which reflect how the results of such processes are encoded for storage and verification. Several of the KDFs lack publicly documented digest formats due to their limited adoption and absence from formal standardization processes.

Table 3. KDF digests (Anonymous, 2025)

KDF	Digest
Argon	$argon2id$v=19$m=65536,t=2,p=1$gZiV/M1gPc22E$lH1En5eHEPvWWzApCTetd3Xl65ytiM4W99bRjFpbM
Battcrypt	Common/Standardized digest format not widely adopted
Catena	$catena$dragonfly$10$m=65536,t=2,p=1$ileG836J$pmSICYS8Nh8utulAeb5CztaWXtczijq0ZoJZqqHsL1T
Lyra2	$lyra2$1$m=65536,t=2,r=4$nIks49z6$CR2hYiziZcmBPOV56JisutRGt2txcS6iSHLLhrKvg6b
MAKWA	$makwa$2048,t=10000$2Ztq41qm$k9yoBQDf7NW9wVi4Q4saesgJyN386uWH7P3VbeMCfQU
Parallel	Common/Standardized digest format not widely adopted
POMELO	Common/Standardized digest format not widely adopted
Pufferfish	Common/Standardized digest format not widely adopted
yescrypt	$y$j9T$F5Jx5fExrKuPp53xLKQ..1$tVtFMx0Aj05nVcJZIUjztwvFLQncJjGsMPR6wV748pN

3.3. Enhancing Security with Salting and Peppering

In order to guarantee that identical passwords yield different hash values, salting adds a unique random value to each password prior to hashing. This mitigates rainbow table attacks by preventing precomputed hash lookups (Bellovin and Merritt, 1993). Figure 2 illustrates the process of applying a salt to a user password before hashing, ensuring unique hash outputs even for identical passwords. Salting is universally recommended in standards such as NIST SP 800-63B and OWASP ASVS. The presence of mandatory salting support in PHC finalists is thus directly evaluated in our work.

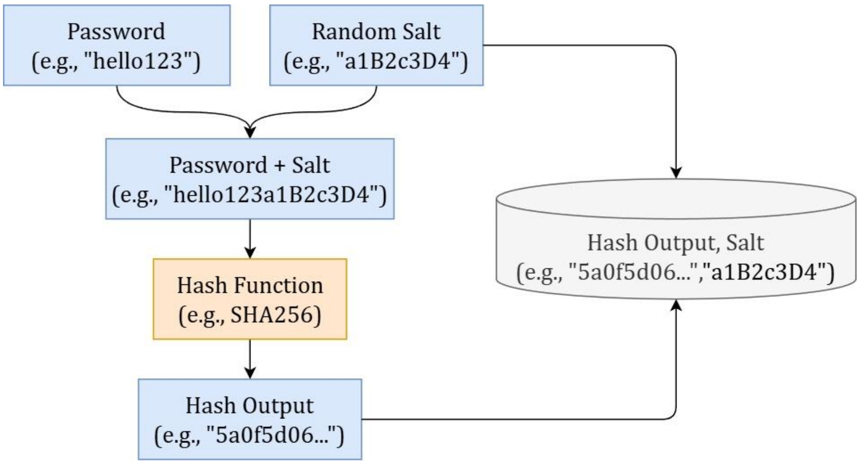


Figure 2. Process of using salt in password hashing.

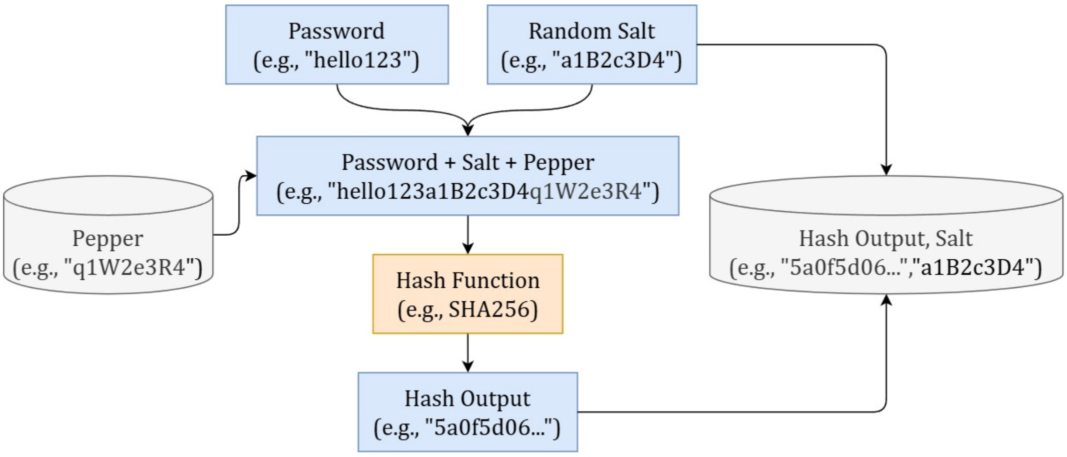


Figure 3. Process of using pepper in password hashing.

ASVS v4.0.3 recommends using a KDF for password hashing, with a minimum salt length of 32 bits to reduce collision risks (OWASP ASVS, 2021).

Peppering, on the other hand, involves adding a system-wide secret value that is not stored in the database, significantly increasing resistance against precomputed attacks. Unlike salts, peppers are securely stored in separate vaults or Hardware Security Modules (HSMs), making hash database breaches significantly less effective (Blocki and Sridhar, 2016). Figure 3 demonstrates how a system-wide pepper is added to the password prior to hashing, further enhancing resistance against offline attacks.

Modern KDFs, including Argon2, Catena (Forler et al., 2013), Lyra2 and MAKWA (Pornin, 2015), inherently incorporate salting. However, peppering must be implemented separately, as it provides an additional layer of security against offline brute-force attacks (Blocki and Sridhar, 2016). Even if an attacker compromises a hash database, peppering prevents straightforward password recovery. Peppering is not inherently built into PHC finalists, but it is discussed in our recommendations as a practical enhancement.

3.4. Password Hashing Competition

The PHC was a major initiative aimed at identifying advanced and secure password hashing schemes to address weaknesses in traditional password-based KDFs. Launched in 2013, PHC evaluated 22 candidate schemes, selecting 9 finalists based on security, efficiency, and adaptability to different application scenarios (Hatzivasilis et al., 2015).

The PHC provided a platform for cryptographic researchers to propose, analyze, and refine password hashing schemes that improved upon PBKDF2, bcrypt, and scrypt. The competition led to the selection of more memory-hard algorithms, making brute-force attacks using GPUs, FPGAs, and Application-specific integrated circuits (ASICs) significantly more difficult (Blocki et al., 2018).

A primary focus of PHC was memory-hardness, a property that forces an adversary to consume large amounts of RAM, thus limiting the effectiveness of parallel attacks. The Argon2 algorithm, one of the PHC winners, demonstrated superior memory-hard properties and was later standardized as RFC 9106 (RFC 9106, 2021), becoming the recommended password hashing function by NIST (Grassi et al., 2023).

The PHC contributed to the adoption of memory-hard hashing techniques in real-world applications, including operating systems, cloud authentication, and cryptographic key derivation. Argon2, Lyra2, and Catena, among others, gained traction in various industries (Hatzivasilis et al., 2015).

The competition provided an empirical analysis of execution time, memory consumption, and resilience to various attacks, including side-channel attacks, brute-force, and garbage-collector exploits. This rigorous benchmarking ensured that the selected schemes were not only secure but also practical for different environments, including web authentication, embedded systems, and cryptographic key management (Hatzivasilis, 2017).

By bringing together cryptographers, security researchers, and developers, PHC fostered innovation in password security. It introduced novel approaches, such as hybrid password hashing and adaptive cost parameters, ensuring flexibility in different security models (Hatzivasilis et al., 2015).

PHC addressed the vulnerabilities in traditional hashing functions, such as SHA-1 and MD5, and KDFs such as PBKDF2, bcrypt and scrypt by promoting modern password hashing algorithms that offer improved resistance to dictionary attacks, rainbow table attacks, and brute-force attacks (Wetzels, 2016).

This study is deliberately focused on the Password Hashing Competition (PHC) finalists, as they represent the most systematically evaluated and openly scrutinized class of modern password-hashing and key-derivation functions.

3.5. Memory-Hard Functions (MHFs) and Time-Memory Trade-offs

MHFs are cryptographic primitives designed to increase computation cost by demanding significant memory resources. MHFs slow down brute-force attacks by making them inefficient for hardware-accelerated adversaries, such as those using GPUs, FPGAs, or ASICs (Alwen et al., 2018).

Notable examples include Argon2 and yescrypt, both designed to resist attacks that rely on parallelization and memory optimization. These functions ensure that brute-force attempts require high memory bandwidth, making mass-parallelized attacks infeasible (Choe et al., 2019).

The time-memory trade-off describes the balance between computational cost and memory usage in password hashing. By increasing memory requirements, brute-force attacks become more expensive and less feasible for attackers (Luo et al., 2021). However, implementing MHFs in real-world systems requires careful consideration of resource constraints, particularly for low-power embedded devices and cloud-based environments.

Memory hardness is the main differentiator between PHC finalists and legacy KDFs. It is also a decisive factor in compliance with modern security standards, which emphasize protection against hardware-accelerated brute-force attacks.

3.6. Post-Quantum Cryptography and KDFs

Quantum computation significantly jeopardizes cryptographic security, namely that of traditional KDFs. Grover's algorithm enables quantum computers to discover unsorted databases at quadratically faster rates than classical computers and therefore undermines the security of symmetric cryptographic schemes (Grover, 1996). Shor's algorithm can factor large numbers efficiently and compute discrete logarithms, compromising asymmetric cryptographic schemes such as RSA and ECC (Shor, 1994).

Computing power increases rapidly due to Moore's Law, which asserts that an integrated circuit's transistor count doubles about every 18 months (Moore, 1965). While Moore's Law initially referred to traditional computing, similar spectacular progress is now being witnessed in quantum computing hardware, accelerating the rate at which conventional cryptographic algorithms are to be cracked. With increased qubit stability and error correction developments, quantum computers will have thresholds at which it may be feasible to break RSA-2048 or ECC-256 encryption in the near term, over the next twenty years (Grassi et al., 2023).

To mitigate these risks, post-quantum secure KDFs must be developed resistant to quantum brute-force attacks. While symmetric cryptographic functions can theoretically double their key lengths to withstand Grover's algorithm, this approach is only a temporary solution as hardware advancements continue to accelerate. Argon2 and yescrypt, among the PHC finalists, have demonstrated resilience against quantum threats since they are memory-hard, which make quantum-based brute-force attacks computationally infeasible (Choe et al., 2019).

With quantum technology developing at a Moore's Law rate, cryptographic systems must adopt quantum-resistant hashing and key derivation techniques.

When considering the PHC finalists, their quantum resilience varies substantially:

- Argon2 and yescrypt: Their memory-hard designs remain robust against Grover’s quadratic speedup, as the limiting factor is not computation alone but memory bandwidth. This makes them highly suitable for quantum-threatened domains such as electronic health records and online banking, where adversaries may invest in large-scale quantum resources.
- Lyra2 and Catena: Also memory-hard, with tunable parameters that can be adapted over time. These are promising candidates for regulated industries that anticipate progressive increases in adversarial computing power.
- MAKWA and Parallel: Rely primarily on computational hardness rather than memory, making them more vulnerable under Grover’s model. Their use in safety-critical sectors would require significant compensating controls (e.g., stronger multi-factor authentication).
- POMELO and Pufferfish: Show weaker randomness guarantees and incomplete formal validation. Under quantum threat assumptions, these limitations

become more critical, suggesting restricted applicability in compliance-heavy sectors.

- battcrypt: As a simplified derivative of scrypt, it inherits partial memory-hardness but lacks the extensive validation necessary to justify deployment in high-assurance systems facing quantum adversaries.

To make our post-quantum claims testable, Table 4 reports the effective security against Grover's algorithm as a function of password entropy E. While the information-theoretic upper bound is E/2 bits for any scheme, memory-hard finalists (Argon2, Catena, Lyra2, yescrypt) reduce the practical quantum advantage by turning the oracle into a bandwidth-bound computation; compute-bound designs (MAKWA, Parallel) offer less damping and therefore require stronger operational compensations (e.g., higher-entropy secrets, aggressive rate-limiting, and MFA).

From a sectoral perspective, domains such as finance and healthcare are characterized by long retention periods for sensitive data and high-value adversaries. In these environments, adopting memory-hard PHC finalists such as Argon2, Catena, or Lyra2 provides stronger assurances of durability against both classical and quantum-enabled brute-force attempts. Conversely, adopting lighter, computationally bound functions risks premature obsolescence in the face of quantum advances.

In regulated environments where long-term confidentiality and compliance are paramount, conservative parameter choices are recommended. Specifically, security margins should account for the quadratic speed-up offered by Grover's algorithm: either by doubling the assumed password entropy requirements, or by doubling memory and time-cost parameters for memory-hard functions to constrain any potential quantum advantage. These adjustments provide a practical buffer against future quantum adversaries while maintaining alignment with compliance frameworks.

Table 4. Quantum resistance of PHC finalists: entropy and effective security bits under Grover's algorithm

KDF	Memory-hard design	Salt support	Theoretical effective bits vs. Grover	Practical quantum advantage (qual.)	Notes for parameterization under quantum threat
Argon2	Yes	Yes	E/2	Low–Medium	Increase memory (≥ 64–128 MiB per hash) and lanes as platform allows; prioritize Argon2id.
Catena	Yes	Yes	E/2	Low–Medium	Use higher garlic levels; rely on data-independent access to reduce side channels.
Lyra2	Yes	Yes	E/2	Low–Medium	Favor larger matrix sizes; ensure constant-time configuration where possible.
Yescrypt	ROM/seq. memory hardness	Yes	E/2	Medium	Leverage large ROM ("ROM-port") where feasible; tune to maximize bandwidth pressure.
battcrypt	Derived from scrypt (MH)	Yes	E/2	Medium	Increase N,r,p to push memory; validate implementation maturity.
POMELO	Yes	Yes	E/2	Medium	Only where vetted; use high memory; note limited KDF adoption.
Pufferfish	Yes	Yes	E/2	Medium	High memory settings advised; limited formal analysis reported.
MAKWA	No (compute-bound)	Yes	E/2	High	Grover applies more directly; compensate with stronger MFA and rate-limits.
Parallel	No (compute-bound)	Yes	E/2	High	Prefer MH alternatives for long-lived secrets; tighten operational controls.

4. Standardization and Regulatory Alignment

Various regulatory frameworks establish standards for password hashing, authentication, and cryptographic key management. This section outlines some existing security standards and regulatory requirements that impact password hashing implementations.

4.1. Cryptographic Standards for Password Storage

Modern cryptographic measures underline the secure storage of passwords and the implementation of strong authentication protocols. Below are key guidelines that must be considered:

NIST SP 800-63B: It is one of the most influential and widely adopted standards in both U.S. federal systems and the private sector. Its strong technical stance on secure password handling (e.g., avoiding composition rules, using slow hashing functions) aligns with your article's evaluation criteria for secure, compliant password hashing.

Recommends password hashing techniques resistant to brute-force and offline attacks. A memory-hard KDF is recommended to increase the cost of such attacks. While a minimum salt length of 32 bits is required, NIST emphasizes that salts should provide at least 112 bits of entropy in practice. This distinction highlights that salt length and entropy are not equivalent, and salts should be generated using a cryptographically secure pseudorandom number generator (CSPRNG) to ensure unpredictability and collision resistance (Grassi et al., 2023).

ISO/IEC 27001: It's an international standard and provides a broader context for organizational compliance beyond technical cryptographic choices. It ensures your analysis considers not just technical efficacy but organizational readiness and alignment with globally accepted security frameworks.

Specifies that organizations must implement appropriate organizational, people, physical, and technological controls to ensure the confidentiality, integrity, and availability of information assets like credentials and authentication mechanisms. Within the Information Security Management System (ISMS) framework, the standard requires the deployment of secure password policies—covering aspects such as protected storage, restricted access, and regular review of authentication controls.

Although ISO/IEC 27001 does not specify specific cryptographic techniques, it strongly advices the use of industry-standard mechanisms such as KDFs, secure hashes, and salting mechanisms to ensure password data protection. Annex A of ISO/IEC 27001:2022, specifically Control A.9.4.3 – Password Management System, outlines requirements for implementing a password management system that enforces strong, unpredictable, and securely stored passwords (ISO/IEC 27001: 2022, 2022).

PCI DSS: It is mandatory for organizations handling credit card data, making it highly relevant to commercial environments. Including PCI DSS strengthens the compliance mapping by reflecting industry-specific requirements where password protection is legally and financially critical.

Requires organizations that store, process, or transmit cardholder data to implement strong cryptography to secure authentication credentials, including user passwords. Stored passwords need to be made unreadable using strong one-way hashing with a salt value, and should be implemented in accordance with recognized cryptographic standards.

Although PCI DSS does not prescribe a specific algorithm, it recommends the use of keyed cryptographic hashes (e.g., HMAC) or industry-approved KDFs, and mandates that the resulting cryptographic strength be equivalent to at least 128 bits of security. Additionally, passwords must be changed at least every 90 days, and MFA is required for administrative access to systems handling payment data (PCI Security Standards Council, 2024).

OWASP Application Security Verification Standard (ASVS): It represents the developer and application security perspective, bridging the gap between compliance and implementation. Its inclusion ensures your article speaks to secure coding practices and real-world development guidelines, not just regulatory checkboxes.

OWASP ASVS calls for password hashing implementations to support adaptive security features, enabling systems to adjust computational costs in response to evolving attack techniques. ASVS v4.0.3 recommends using a KDF for password hashing, with a minimum salt length of 32 bits to reduce collision risks. When using PBKDF2, the iteration count should be at least 100,000. For bcrypt, a work factor of at least 13 is advised. However, this version does not include any PHC finalists (OWASP ASVS, 2021). The upcoming ASVS v5.0 introduces Argon2, the winner of the PHC, as a recommended algorithm. Specifically, Argon2id should be configured with a memory cost of 19MB, a time cost of 2, and a parallelism factor of 1. Additionally, PBKDF2_SHA512 will require 210,000 iterations, while PBKDF2_SHA256—recommended for FIPS-140 compliance—must have 600,000 iterations. The new version also includes scrypt as an alternative when Argon2id is unavailable, with a required configuration of 2^{15} cost, a minimum block size of 8, and a parallelization parameter of 1. Recommendations for bcrypt remain unchanged, making it suitable for legacy systems (OWASP ASVS, 2025).

These standards emphasize salting, key stretching, and MHFs to eliminate or mitigate password-guessing attacks and enforce best practices in cryptography.

4.2. Data Protection Regulations and Cryptographic Expectations

While no specific regulation explicitly mandates the use of KDFs, modern privacy laws strongly recommend cryptographically secure password handling techniques. This includes advanced KDFs such as Argon2, or legacy alternatives like bcrypt, PBKDF2, or scrypt, where Argon2 is unavailable.

Organizations are expected to implement high-entropy, memory-hard, and computationally adaptive KDFs to comply with established security best practices. GDPR Article 32, for instance, requires appropriate technical and organizational measures (TOM) to safeguard personal data. Although it does not specify algorithms, it implicitly demands strong encryption and hashing mechanisms (GDPR, 2016).

Similarly, CCPA, LGPD, PIPL, PIPEDA, and KVKK enforce regulatory obligations that emphasize secure data processing. These frameworks impose legal penalties for non-compliance and recommend the use of state-of-the-art cryptographic protections for sensitive data (KVKK, 2016; CCPA, 2018; LGPD, 2018; PIPEDA, 2000; PIPL, 2021).

Among the finalists of the PHC, Argon2 stands out as the most popular based on its solid security properties, memory hardness, and alignment with contemporary cryptographic norms. Although no formal regulations mandate its use, Argon2 remains a preferred choice due to its adherence to contemporary cryptographic standards.

5. Results

This section presents a multi-dimensional evaluation of PHC finalists, including performance benchmarks, security assessments, and compliance mapping with established cryptographic standards and privacy regulations.

5.1. Performance Analysis

The performance comparison of the PHC finalists, as presented in Table 5, highlights significant differences in execution time, memory usage, and code size among the various KDFs.

Execution time varies significantly across the finalists, with MAKWA exhibiting the fastest computation at 0.015621 ms, followed closely by Parallel (0.047051 ms) and yescrypt (0.058253 ms). On the other hand, Argon2 and Pufferfish demonstrate higher execution times, indicating a trade-off between security features and computational efficiency.

Memory usage is another critical factor, where POMELO has the broadest range, consuming between 1KB to 8GB, making it highly flexible yet potentially resource-intensive. Similarly, Argon2 and Lyra2 can demand up to 1GB, which may be challenging for memory-constrained environments. In contrast, Parallel has negligible memory usage, while MAKWA and yescrypt exhibit moderate requirements.

Code size further differentiates these KDFs, with Pufferfish requiring the largest footprint at 103KB, followed by Lyra2 (98KB) and MAKWA (95KB), suggesting a more complex implementation. Conversely, Catena and battcrypt are among the most lightweight solutions with code sizes of 25KB and 27KB, respectively.

Overall, the selection of an appropriate PHC finalist depends on the balance between security, computational efficiency, and system resource constraints. While MHFs like Argon2 and POMELO enhance security, they impose significant memory overhead, making them less suitable for constrained environments. Meanwhile, lightweight solutions like MAKWA and Parallel offer speed and low memory usage but may require supplementary cryptographic validation and side-channel protection to meet the security expectations of high-assurance systems.

In practical terms, Argon2 and POMELO provide strong memory-hardness but their high memory consumption makes them less suitable for mobile or IoT devices. MAKWA and Parallel are very fast and lightweight but, being non-memory-hard, they are more vulnerable to GPU brute-force attacks. Yescrypt offers a balanced trade-off, delivering moderate memory usage and execution time while retaining partial resistance to hardware acceleration. These distinctions suggest that constrained environments may prioritize MAKWA or Parallel for performance, whereas high-assurance systems in finance or healthcare should adopt Argon2 or similar MHFs despite higher costs.

Table 5. PHC Finalists performance comparison (Forler et al., 2013; Simplicio et al., 2014; Thomas, 2014; Gosney, 2015; Hatzivasilis et al., 2015; Peslyak, 2015; Pornin, 2015; Thomas, 2015; Wu, 2015; RFC 9106, 2021)

Feature	Execution Time (ms) MIN-MAX	Memory Usage (KB)	Code Size (KB)
Argon2	0,008917 - 577.02208	1KB - 1GB	82
Battcrypt	0,000312 - 2.853051	18KB - 128MB	27
Catena	0,353742 - 5.461030	8MB	25
Lyra2	0,000084 - 2.916398	400MB - 1GB	98
MAKWA	0,000096 - 0.015621	335KB	95
Parallel	0,001000 - 0.047051	neglected	71
POMELO	0,000031 - 8.504152	1KB - 8GB	67
Pufferfish	0,000057 - 38.341005	4KB - 16KB	103
yescrypt	0,000094 - 0.058253	44KB - 3MB (RAM), 3GB (ROM)	36

While execution time provides a useful baseline for comparative performance, it does not fully capture real-world efficiency. Future studies should incorporate additional metrics such as energy consumption per hash, computational complexity under hardware constraints (e.g., embedded devices, FPGAs), and scalability across platforms, especially in scenarios involving green computing or battery-limited environments.

5.2. Security Analysis

Table 6 presents a security comparison of the PHC finalists, evaluating their resistance to GPU-based attacks, memory-hardness (MH), side-channel resistance (SCR), and protection against preimage resistance (PR). These security properties are critical in determining the robustness of KDFs against brute-force attacks and cryptanalytic techniques.

GPU resistance is a key factor in preventing attackers from efficiently parallelizing brute-force attempts. Most finalists, including Argon2, battcrypt, Catena, Lyra2, POMELO, and Pufferfish, demonstrate GPU resistance, making them well-suited for mitigating hardware-based attacks. In contrast, MAKWA, Parallel, and yescrypt do not exhibit strong GPU resistance, potentially making them more vulnerable to attacks using specialized hardware.

MH is an essential property for slowing down adversarial computations, particularly in password cracking. The majority of finalists, including Argon2, battcrypt, Catena, Lyra2, and POMELO, are memory-hard, ensuring that they require significant computational and memory resources to compute. However, MAKWA and Parallel lack memory-hard properties, while yescrypt implements a unique ROM (Read-Only Memory)-Port sequential approach, which influences its memory usage.

Side-channel resistance (SCR) varies among the finalists, with Catena and Parallel explicitly designed to resist side-channel attacks. Argon2 includes the Argon2i variant, which provides some level of protection, while MAKWA and POMELO offer partial side-channel resistance. The remaining finalists, such as battcrypt, Lyra2, and Pufferfish, lack documented side-channel resistance, which could be a security limitation in environments where such attacks are a concern.

Overall, this comparison highlights the trade-offs among the PHC finalists, with some prioritizing GPU resistance and memory-hardness, while others focus on side-channel resistance. The selection of an appropriate KDF depends on the specific security requirements of the application, balancing resilience against brute-force, side-channel, and preimage attacks with operational efficiency, scalability, and deployment feasibility across heterogeneous environments.

From a security perspective, Catena and Argon2 are among the strongest candidates, as both provide robust memory-hardness and resistance against parallel hardware acceleration, making them effective against GPU and ASIC-based brute-force attacks. Lyra2 also offers strong security properties with tunable memory parameters, though its adoption remains limited outside research contexts. Yescrypt improves upon scrypt by incorporating additional defenses such as ROM-port hardness, giving it enhanced resistance to tradeoff attacks. By contrast, MAKWA and Parallel, while efficient, are compute-bound and therefore more exposed to quantum and massively parallel adversaries. Overall, the PHC finalists demonstrate varied security postures: memory-hard designs (Argon2, Catena, Lyra2, yescrypt) provide the highest assurance, while non-memory-hard schemes (MAKWA, Parallel) should be complemented with operational controls such as multi-factor authentication and rate limiting.

5.3. Compliance Analysis

Ensuring compliance with established security standards and privacy regulations is a critical aspect of evaluating password hashing mechanisms. This section provides a comprehensive compliance analysis of PHC finalists in relation to key security frameworks and privacy regulations.

To ensure transparency in our compliance mappings, we adopted explicit criteria to evaluate the alignment of PHC finalists with standards such as OWASP ASVS, PCI DSS, and NIST SP 800-63B. First, we examined whether an algorithm is explicitly listed in ASVS-recommended algorithms (e.g., Argon2id in OWASP ASVS v5.0). Algorithms not mentioned were considered non-aligned, regardless of their technical merits. Second, we assessed broad adoption and implementation support across libraries, frameworks, and security toolkits, since limited adoption diminishes the likelihood of consistent and secure deployment in practice. Third, we considered the extent of long-term peer review and maintenance, recognizing that algorithms without sustained academic scrutiny or community support may not meet the maturity expectations of compliance frameworks. For example, Lyra2 was not mapped to OWASP ASVS compliance because it is not explicitly recommended in the current ASVS versions and lacks the same ecosystem maturity and long-term validation as Argon2. These objective criteria provide a reproducible basis for our compliance mappings and allow practitioners to independently verify the rationale behind each assignment.

Table 6. PHC finalists security comparison (Simplicio et al., 2014; Thomas, 2014; Forler et al., 2015; Gosney, 2015; Hatzivasilis et al., 2015; Peslyak, 2015; Pornin, 2015; Thomas, 2015; Wu, 2015; RFC 9106, 2021)

Feature	GPU Resistance	Memory-Hardness	Side-channel Resistance
Argon2	✓	✓	Argon2i
Battcrypt	✓	✓	-
Catena	✓	✓	✓
Lyra2	✓	✓	-
MAKWA	-	-	Partially
Parallel	-	-	✓
POMELO	✓	✓	Partially
Pufferfish	✓	✓	-
yescrypt	-	ROM-Port, sequential	-

NIST SP 800-63B provides digital identity guidelines with specific requirements for password-based authentication. It emphasizes the use of salted password hashing, resistance against dictionary attacks, and avoidance of composition rules that decrease security. PHC finalists such as Argon2 align with these principles by implementing MHFs, which enhance resistance against brute-force and GPU-based attacks (Grassi et al., 2023).

ISO/IEC 27001 specifies requirements for establishing an information security management system (ISMS). Compliance with this standard necessitates the adoption of strong cryptographic controls, including secure password storage mechanisms. The PHC finalists comply with these controls by incorporating key stretching and secure KDFs, contributing to overall system resilience (ISO/IEC 27001: 2022, 2022).

PCI DSS mandates secure authentication and password storage mechanisms for payment systems. It requires hashing algorithms to be resistant to preimage attacks and brute-force attempts. Many PHC finalists, particularly Argon2 and Lyra2, are designed to provide such protections, making them suitable for PCI DSS compliance (PCI Security Standards Council, 2024).

OWASP ASVS 4.0.3 provides guidelines for secure software development, emphasizing strong authentication and credential storage mechanisms. It recommends the use of adaptive, memory-hard password hashing algorithms, which PHC finalists generally fulfill, ensuring protection against offline attacks (OWASP ASVS, 2021).

GDPR mandates secure storage of personal data, including passwords, to prevent unauthorized access. Article 32 specifically calls for encryption and pseudonymization measures. PHC finalists with strong password hashing properties align with GDPR's security requirements by ensuring irreversible password storage (GDPR, 2016).

CCPA focuses on protecting personal data of California residents. While it does not mandate specific cryptographic techniques, it requires reasonable security measures to prevent data breaches. The implementation of PHC finalists with strong cryptographic properties supports CCPA's security expectations (CCPA, 2018).

Brazil's LGPD follows principles similar to GDPR, requiring organizations to implement technical and administrative measures for data protection. Password hashing mechanisms that resist brute-force attacks and credential leaks align with LGPD compliance (LGPD, 2018).

China's PIPL requires secure processing of personal data, emphasizing encryption and data minimization. Secure password hashing techniques, such as those employed by PHC finalists, ensure compliance by mitigating the risks of unauthorized access and credential compromise (PIPL, 2021).

PIPEDA mandates organizations to use appropriate security safeguards, including encryption, to protect sensitive data. The use of PHC finalists ensures compliance with these requirements by providing strong password storage mechanisms (PIPEDA, 2000).

Türkiye's KVKK aligns with GDPR principles, requiring adequate security measures for protecting personal data. The adoption of PHC finalists supports KVKK compliance by providing robust password protection techniques (KVKK, 2016).

The PHC finalists generally exhibit strong compliance with major security frameworks and privacy regulations. Their adoption in authentication systems enhances resilience against modern threats while aligning with regulatory requirements. Implementing PHC finalists ensures that organizations not only improve security but also adhere to international compliance obligations, reducing the risk of data breaches and legal liabilities. A detailed comparison of the PHC finalists' alignment with major global cryptographic standards and data protection regulations is presented in Table 7.

The cryptographic requirements outlined in the PCI DSS and NIST SP 800-63B standards align with Argon2, Catena, Lyra2 and yescrypt, as these algorithms are MHFs that have undergone extensive security analysis. Additionally, OWASP ASVS v5.0.0 explicitly recommends Argon2 as a preferred KDF.

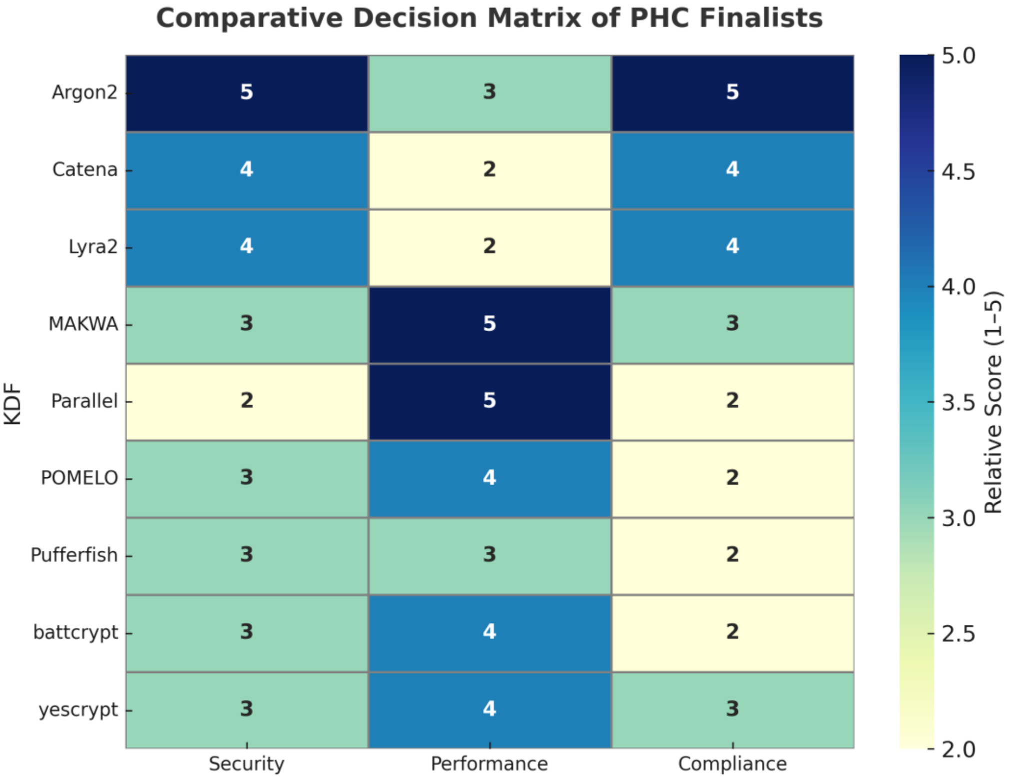


Figure 4. Comparative decision matrix of PHC finalists based on normalized scores (1–5) across three dimensions: security, performance, and compliance.

Table 7. Compliance mapping: PHC finalists vs. global cryptographic standards

Regulation	Cryptographic requirement	PHC alignment	Justification
OWASP ASVS	Adaptive, memory-hard KDFs	Argon2	Argon2 is explicitly recommended in OWASP ASVS as a memory-hard KDF suitable for password storage.
NIST SP 800-63B	Salting, MHFs, entropy	Argon2, Catena, Lyra2, yescrypt	These KDFs provide mandatory salting, support memory hardness, and allow parameterization to enforce entropy requirements consistent with NIST guidance.
PCI DSS	One-way hash, 128-bit strength	Argon2, Catena, Lyra2, yescrypt	All listed PHC KDFs implement one-way password hashing with tunable cost factors, achieving ≥128-bit effective strength in compliance with PCI DSS requirements.
ISO/IEC 27001	Secure credential handling in ISMS	Argon2, Catena, Lyra2, MAKWA, yescrypt	These KDFs align with ISO/IEC 27001 Annex A controls for cryptographic protection of authentication data; memory-hardness and salting address brute-force resilience.
GDPR	Technical and organizational measures (Art. 32)	Argon2, Catena, Lyra2, MAKWA, yescrypt	Provides strong, tunable password protection that satisfies GDPR's requirement for “state of the art” technical measures to secure personal data.
CCPA, KVKK, LGPD, PIPEDA, PIPL	Strong encryption and hashing	Argon2, Catena, Lyra2, MAKWA, yescrypt	Listed PHC finalists ensure strong hashing with salts and resistance to large-scale attacks, fulfilling international data protection laws' requirement for secure handling of personal data.

Furthermore, the ISO/IEC 27001 standard, along with other privacy-focused regulations such as GDPR, KVKK, LGPD, and PIPL, emphasize the necessity of strong cryptographic hashing mechanisms. These requirements are met by Argon2, Catena, Lyra2, MAKWA, and yescrypt, which provide robust security properties suitable for secure credential storage and authentication. To provide decision-makers with a more accessible comparative overview, we synthesized the results of Tables 5-7 into a visual decision matrix (Figure 4).

This heatmap aggregates three core evaluation dimensions—security strength, computational performance, and regulatory compliance—on a normalized 1–5 scale for each PHC finalist. The visualization highlights clear trade-offs: while Argon2 consistently achieves the highest scores across all categories, Catena and Lyra2 also demonstrate strong compliance and security properties, albeit with higher performance costs. Conversely, lightweight schemes such as MAKWA, Parallel, and yescrypt achieve higher efficiency but exhibit weaker compliance coverage or incomplete side-channel resistance. By consolidating multi-dimensional metrics into a single comparative figure, this decision matrix enhances the practical utility of the study, allowing developers, researchers, and compliance officers to identify algorithms that align with both technical requirements and regulatory expectations.

6. Conclusion

This study concludes that among the PHC finalists, Argon2 stands out as the most secure and effective modern KDF, particularly due to its memory-hard design, which makes it resilient against hardware-accelerated brute-force attacks (Choe et al., 2019). While bcrypt remains a viable option for legacy systems requiring long-term support and compatibility, and yescrypt offers a performance-security trade-off suitable for general-purpose environments. The choice of algorithm should be tailored to the security requirements and system limitations.

For web applications, Argon2id with properly tuned memory and time cost parameters remains the most balanced option, aligning with OWASP ASVS v5.0 and offering future-proof security. Employing Argon2 with appropriate memory and time cost parameters offers strong safeguarding against large-scale attacks. The usage of salt and pepper mechanisms further boosts security.

For resource-constrained environments such as IoT or embedded systems, lightweight schemes like yescrypt or optimized configurations of Argon2 provide workable trade-offs between performance and security. Argon2 with adjusted memory parameters can also be optimized to provide improved security without the excessive use of resources.

While standards such as NIST SP 800-63B, ISO/IEC 27001, and PCI DSS do not require specific KDFs, they point out requirements such as memory-hardness, entropy, and one-way functions. OWASP ASVS does, however, specifically suggest Argon2 by name and specific configuration parameters, a big step towards prescriptive guidelines for password hashing best practices (OWASP ASVS, 2025).

From both a research and deployment perspective, future password storage solutions must go beyond mere compliance and actively integrate quantum-resilient, memory-hard algorithms like Argon2id. Developing formally verified, post-quantum resistant designs that retain efficiency and adaptability will be critical as new attack vectors—particularly those involving specialized hardware and quantum computing—continue to emerge. Developers working in highly regulated environments such as banking or healthcare should prioritize algorithms that have strong community support, robust compliance mappings, and validated post-quantum resistance. In our view, the evolution of regulatory frameworks like OWASP ASVS v5.0 shows a trend toward prescriptive, technically grounded requirements. This trajectory will likely continue, making it crucial for both researchers and practitioners to anticipate these shifts when selecting KDFs for long-term systems. As password-based systems remain a cornerstone of digital authentication, balancing regulatory alignment, quantum resistance, and operational feasibility will define the next generation of secure credential storage strategies.

Author Contributions

The percentages of the authors' contributions are presented below. All authors reviewed and approved the final version of the manuscript.

	E.U.	B.C.
C	50	50
D	20	80
S		100
DCP	50	50
DAI	50	50
L	20	80
W	70	30
CR	20	80
SR	50	50
PM	20	80

C=Concept, D= design, S= supervision, DCP= data collection and/or processing, DAI= data analysis and/or interpretation, L= literature search, W= writing, CR= critical review, SR= submission and revision, PM= project management.

Conflict of Interest

The authors declared that there is no conflict of interest.

Ethical Consideration

Ethics committee approval was not required for this study because there was no study on animals or humans.

References

Álvarez R, Zamora A. 2017. Using spritz as a password-based key derivation function. In: Int Joint Conf SOCO'16-CISIS'16-ICEUTE'16: San Sebastián, Spain, October 19th-21st, 2016 Proceedings 11, pp: 518-525.

Alwen J, Gazi P, Kamath C, Klein K, Osang G, Pietrzak K, Rybár M. 2018. On the memory-hardness of data-independent password-hashing functions. In: Proceedings of the 2018 on Asia Conf Comp Commun Secur, pp: 51-65.

Andrade ER, Simplicio MA, Barreto PS, dos Santos PC. 2016. Lyra2: Efficient password hashing with high security against time-memory trade-offs. IEEE Trans Comput, 65(10): 3096-3108.

Anonymous. 2025. PHC string format. URL: https://github.com/P-H-C/phc-string-format/blob/master/phc-sf-spec.md (accessed date: April 1, 2025).

Aumasson JP. 2013. Password hashing: the future is now, Kudelski Security, Switzerland, pp: 1-10.

Backendal M, Clermont S, Fischlin M, Günther F. 2025. Key derivation functions without a grain of salt. In Annual Int Conf Theory Appl Cryptogr Tech, pp: 393-426.

Bellovin SM, Merritt M. 1993. Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: Proc of the 1st ACM Conf Computer Commun Secur, pp: 244-250.

Blocki J, Harsha B, Zhou S. 2018. On the economics of offline password cracking. In: 2018 IEEE Symp Secur Privacy (SP), pp: 853-871.

Blocki J, Sridhar A. 2016. Client-cash: Protecting master passwords against offline attacks. In: Proceedings of the 11th ACM on Asia Conf Comp Commun Secur, pp: 165-176.

CCPA. 2018. California Consumer Privacy Act (CCPA). URL: https://oag.ca.gov/privacy/ccpa (accessed date: March 04, 2025).

Choe J, Moreshet T, Bahar RI, Herlihy M. 2019. Attacking memory-hard scrypt with near-data-processing. In: Proc Int Symp Mem Syst, pp: 33-37.

Clark M, Seamons K. 2022. Passwords and cryptwords: the final limits on lengths. In: Proc 2022 New Secur Paradigms Workshop, pp: 75-89.

Forler C, List E, Lucks S, Wenzel J. 2015. Overview of the candidates for the password hashing competition: And their resistance against garbage-collector attacks. In: Technology and Practice of Passwords: Int Conf Passwords, Passwords'14, Trondheim, Norway, December 8-10, 2014, Revised Selected Papers 7, pp: 3-18.

Forler C, Lucks S, Wenzel J. 2013. Catena: A memory-consuming password-scrambling framework. Cryptology ePrint Arch, pp: 31.

GDPR. 2016. General Data Protection Regulation (GDPR). URL: https://gdpr-info.eu/ (accessed date: March 04, 2025).

Gosney J. 2015. Pufferfish2 password hashing scheme. URL: https://github.com/epixoip/pufferfish (accessed date: April 04, 2025).

Grassi PA, Fenton JL, Newton EM, Perlner RA, Regenscheid AR, Burr WE, Richer JP, Lefkovitz NB, Danker JM, Choong YY, Greene KK, Theofanos MF. 2023. Nist special publication 800-63b digital identity guidelines. Natl Inst Stand Tech (NIST), pp: 27.

Grover LK. 1996. A fast quantum mechanical algorithm for database search. In Proc twenty-eighth annu ACM symp Theory comp, pp: 212-219.

Hatzivasilis G, Papaefstathiou I, Manifavas C. 2015. Password hashing competition-survey and benchmark. Cryptology ePrint Archive, pp: 30.

Hatzivasilis G. 2017. Password-hashing status. Cryptography, 1(2): 10.

ISO/IEC 27001: 2022. 2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. URL: https://www.iso.org/standard/27001 (accessed date: March 04, 2025).

Kodwani G, Arora S, Atrey PK. 2021. On security of key derivation functions in password-based cryptography. In: 2021 IEEE Inter Conf Cyber Secur Resilience (CSR), pp: 109-114.

KVKK. 2016. Personal Data Protection Law (KVKK). URL: https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5 (accessed date: March 04, 2025).

Lata K, Bansal A. 2021. Timing side-channel attack resistant key derivation functions for cryptosystems. In: 2021 IEEE Int

Symp Smart Electron Syst (iSES), pp: 395-399.

LGPD. 2018. General Personal Data Protection Law (LGPD). URL: https://iapp.org/media/pdf/resource_center/Brazilian_General_Data_Protection_Law.pdf (accessed date: March 04, 2025).

Lu YF, Kuo CF, Fang YY. 2016. Efficient storage encryption for android mobile devices. In: Proc Int Conf Res Adapt Convergent Syst, pp: 213-218.

Luo Y, Su Z, Zheng W, Chen Z, Wang F, Zhang Z, Chen J. 2021. A novel memory-hard password hashing scheme for blockchain-based cyber-physical systems. ACM Trans Internet Technol, 21(2): 1-21.

Mcginthy JM, Michaels AJ. 2019. Further analysis of PRNG-based key derivation functions. IEEE Access, 7: 95978-95986.

Moore GE. 1965. Moore's law. Electron Mag, 38(8): 114.

OWASP ASVS. 2021. Open worldwide application security project application security verification standard (OWASP ASVS) v4.0.3. URL: https://github.com/OWASP/ASVS/blob/master/4.0/en/0x11-V2-Authentication.md (accessed date: March 04, 2025).

OWASP ASVS. 2025. Open worldwide application security project application security verification standard (OWASP ASVS) v5.0. URL: https://github.com/OWASP/ASVS/blob/master/5.0/en/0x97-Appendix-V_Cryptography.md (accessed date: March 04, 2025).

OWASP Password Storage Cheat Sheet. 2025. URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html#upgrading-the-work-factor (accessed date: March 23, 2025).

PCI Security Standards Council. 2024. Data security standard. Requir Secur Assess version, 4.0.1. URL: https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4_0_1.pdf (accessed date: March 04, 2025).

Peslyak A. 2015. yescrypt - a password hashing competition submission. URL: https://www.password-hashing.net/submissions/specs/yescrypt-v2.pdf (accessed date: April 04, 2025).

PIPEDA. 2000. Personal information protection and electronic documents act. Department of Justice (PIPEDA). URL: http://laws-lois.justice.gc.ca/PDF/P-8.6.pdf (accessed date: March 04, 2025).

PIPL. 2021. Personal information protection law of the people's republic of China (PIPL). URL: http://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm (accessed date: March 04, 2025).

Pornin T. 2015. The MAKWA password hashing function specifications v1.1. URL: https://www.bolet.org/makwa/makwa-spec-20150422.pdf (accessed date: April 04, 2025).

Provos N, Mazieres D. 1999. A future-adaptable password scheme. In: USENIX annual technical conference, FREENIX track, Vol. 1999, pp: 81-91.

RFC 6070. 2011. PKCS #5: Password-based key derivation function 2 (PBKDF2) test vectors. URL: https://www.rfc-editor.org/info/rfc6070 (accessed date: April 2, 2025).

RFC 9106. 2021. Argon2 Memory-hard function for password hashing and proof-of-work applications. URL: https://www.rfc-editor.org/info/rfc9106 (accessed date: March 20, 2025).

Saad MIM, Jalil KA, Manaf M. 2016. Secured authentication using anonymity and password-based key derivation function. In: Mobile Web Intell Inf Syst: 13th Intern Conf, MobiWIS 2016, Vienna, Austria, August 22-24, 2016, Proceedings 13, pp: 184-197.

Shen Y, Wang L, Gu D. 2025. Security analysis of nist key derivation using pseudorandom functions. Cryptology ePrint Archive, pp: 38.

Shor PW. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In Proc 35th annual symp found comp sci, pp: 124-134.

Simplicio Jr MA, Almeida LC, Andrade ER, dos Santos PC, Barreto PS. 2014. The Lyra2 reference guide. Tech Report v2. 3.2.

Thomas S. 2014. Battcrypt. URL: https://www.password-hashing.net/submissions/specs/battcrypt-v0.pdf (accessed date: April 04, 2025).

Thomas S. 2015. Parallel. URL: https://www.password-hashing.net/submissions/specs/Parallel-v1.pdf (accessed date: April 04, 2025).

Tran DN, Nguyen Tien X, Nguyen Xuan T, Le Viet P. 2024. A user-centric key management for cloud encryption using key derivation function. in: The Intern Conf Intell Syst Networks, pp: 479-487.

Wang M, Duan M, Zhu J. 2018. Research on the security criteria of hash functions in the blockchain. In: Proc 2nd ACM Workshop on Blockchains, Cryptocurrencies, Contracts, pp: 47-55.

Wetzels J. 2016. Open sesame: The password hashing competition and Argon2. arXiv preprint arXiv:1602.03097.

Wu H. 2015. POMELO a password hashing algorithm (Version 2). URL: https://www.password-hashing.net/submissions/specs/POMELO-v3.pdf (accessed date: April 04, 2025).

Yao FF, Yin YL. 2005. Design and analysis of password-based key derivation functions. In: Topics in Cryptology–CT-RSA 2005: The Cryptographers' Track at the RSA Conf 2005, San Francisco, CA, USA, February 14-18, 2005. Proceedings, pp: 245-261.