RESEARCH ARTICLE OPEN ACCESS

MULTI-LAYERED AUTHENTICATION MODEL BASED ON THE COMBINATION OF PASSWORD HASHING AND DIGITAL SECURITY FACTORS

Nguyen Thi Hong Mai - Faculty of Information Technology - Email: mainth@dhhp.edu.vn
Nguyen Manh Hung - Faculty of Information Technology - Email: hungnm@dhhp.edu.vn
Dao Ngoc Tu - Faculty of Information Technology - Email: tudn@dhhp.edu.vn

Abstract

In the context of increasingly sophisticated identity theft attacks, the sole reliance on traditional hash functions no longer guarantees absolute security for authentication systems. This paper focuses on the theoretical research of modern password protection algorithms—such as Bcrypt, Scrypt, and PBKDF2—thereby analyzing the technical characteristics that help mitigate specialized hardware attacks. Building on this foundation, the authors propose a Multi-layered Authentication Model that integrates 'memory-hard' hashing algorithms with digital security components like Multi-Factor Authentication (MFA). The research findings provide a theoretical framework to support the development of personal data security policies and enhance digital literacy for users within higher education environments.

Keywords: Bcrypt, Digital literacy, Information security, Password hashing, Scrypt.

1. Introduction

The advancement of digital transformation in education has significantly escalated cyber security risks. Faculty and students frequently access sensitive data such as grades, research materials, and personal information across Learning Management Systems (LMS) [1,3]. However, legacy password security methods are increasingly exposing vulnerabilities to Brute-force and Dictionary Attacks accelerated by GPU/ASIC (Graphics Processing Unit / Application-Specific Integrated Circuit) hardware. Therefore, researching an authentication model that combines sufficiently strong hashing techniques with additional layers of protection is an urgent requirement.

2. Theoretical Foundation of Modern Hashing Algorithms

2.1. The Bcrypt Algorithm

Bcrypt was developed by Niels Provos and David Mazières in 1999, based on the Blowfish block cipher. The defining characteristic of Bcrypt is its Cost factor mechanism (customizable complexity), which allows the workload of the hashing process to be calibrated in response to hardware advancements.

- Algorithm Structure: Bcrypt utilizes a variant of the Blowfish key schedule, known as Expensive Key Setup (EKS). This process initiates by seeding the S-boxes and P-array with the fractional digits of Pi, followed by consecutive encryption rounds to permute the data [4,6].
- Salt Mechanism: Bcrypt automatically integrates a random 128-bit salt into each password. This guarantees that two identical passwords yield entirely distinct hash values, effectively preventing Rainbow Table attacks.
- Complexity Formula:
  The computational complexity of Bcrypt is directly proportional to the cost parameter (C):

$$T = 2^C \times \text{baseline time}$$

With C typically ranging from 10 to 12 in contemporary systems, the algorithm executes 2^{10} to 2^{12} internal iterations.

- Technical Advantages: Bcrypt exhibits exceptional resistance to Brute-force attacks by intentionally slowing down the validation time for a single password, making the execution of billions of combinations temporally infeasible.

2.2. The Scrypt Algorithm

Scrypt is a Key Derivation Function designed by Colin Percival in 2009. It aims not only to impede computational speed but also to consume an immense amount of memory resources, designating it as a "Memory-hard" function.

- Operating Principle: Unlike algorithms focused solely on the CPU, Scrypt demands a substantial amount of RAM to store pseudo-random sequential blocks during computation. This is achieved via the SMix function, which enforces sequential and random memory access patterns.
- Configuration Parameters: Scrypt relies on a parameter set (N, r, p) where:
  - N: CPU/Memory cost parameter (must be a power of 2).
  - r: Block size.
  - p: Parallelization factor.
- Memory Estimation Formula: The required RAM capacity (M) is approximately calculated by $M \approx 128 \times r \times N$ bytes

  For instance, given N = 16384 and r = 8, a single execution requires roughly 16 MB of RAM.
- Security Capabilities: Scrypt is highly effective against specialized hardware-driven attacks, such as those leveraging FPGA or ASIC platforms. Integrating large amounts of RAM into these custom chips increases manufacturing costs exponentially compared to chips optimized strictly for pure logical computation [4,6].

2.3. The PBKDF2 Algorithm

PBKDF2 (Password-Based Key Derivation Function 2) is a component of the PKCS #5 standard (Password-Based Cryptography) and is specified in RFC 2898. It boasts high compatibility and is widely adopted within Microsoft and Apple architectures, as well as WPA2 standards.

- Operating Mechanism: PBKDF2 applies a pseudo-random function (typically HMAC-SHA256) to the password alongside a Salt over numerous consecutive iterations.
- Function Structure:

$$DK = PBKDF2(PRF, Password, Salt, c, dkLen)$$

  Where:
  - *PRF:* Underlying pseudo-random function.
  - *c*: Number of iterations.
  - *dkLen*: Desired bit-length of the derived key.
- Logical Analysis: The value of each key block T_i is computed by chained XOR operations on the outputs of the PRF across *c* iterations:

$$U_1 = PRF(P, S \parallel i)$$

....

$$U_c = PRF(P, U_{c-1})$$

$$T_i = U_1 \oplus U_2 \oplus \dots \oplus U_c$$

- Characteristics: PBKDF2 is straightforward to implement and holds international compliance certifications. However, compared to Bcrypt and Scrypt, it is less effective against GPU-accelerated attacks due to its lack of a complex memory-overhead management mechanism [4].

2.4. Bcrypt Algorithm and Security Levels According to Cost Parameters

2.4.1. Details of the Blowfish EKS-Based Data Permutation Mechanism

Rather than utilizing standard Blowfish, Bcrypt implements the EKS (Expensive Key Setup) variant to prolong the key initialization phase, thereby imposing friction on attackers. The data permutation process unfolds through highly intricate phases [6]:

- Initialization: The algorithm starts by populating a P-array consisting of 18 elements and four S-boxes (each containing 256 32-bit elements). These initial values are derived from the fractional digits of Pi
- Substitution-Permutation Rounds: The data passes through 16 Feistel rounds [1,7]. In each round, the 64-bit data is divided into two halves: Left (L) and Right (R).
    - Mathematical Operations: $L_i = R_{i-1} \oplus P_i v à R_i = F(L_i) \oplus L_{i-1}$
    - The function F performs lookups across the S-boxes combined with addition and XOR operations to achieve maximum cryptographic diffusion of the password data.
- Key Expansion: The hallmark of EKS is the continuous, interleaved blending of both the Salt and the password into the P-array and S-boxes. This forces any hash cracking attempt to fully compute the expensive key setup phase from scratch for every single trial, completely neutralizing precomputed lookup strategies.

2.4.2. Evaluating the Variability of Cost Parameter C against Processing Time

The cost parameter in Bcrypt represents a base-2 exponent. As the cost value escalates, the time required to compute a hash value scales exponentially, establishing a formidable technical barrier against Brute-Force attacks.

Table 1 simulates the average processing time on a standard computing system when executing password hashing with Bcrypt [4]. Benchmarks were conducted on a personal computer equipped with an Intel Core i5-1135G7 CPU (2.4 GHz, 4 cores, 8 threads), 16GB DDR4 RAM, running Ubuntu 22.04 LTS to minimize background OS noise typically found in Windows environments.

Experimental Design:

- Comparison Benchmarks: Empirical testing was measured across 4 algorithms: Bcrypt, Scrypt (N=16384, r=8, p=1), and PBKDF2 (HMAC-SHA256, 100.000 iterations).
- Measurement Methodology: Each parameter configuration was executed through 100 iterations. Time results are expressed as the mean value after discarding the top and bottom 5% outliers to guarantee objective accuracy.
- Metrics: Execution time (ms), Peak Memory Usage (MB), and authentication throughput (requests/second)

Cost Parameter (C)	Number of Iterations (2^C)	Average Processing Time	Attack Resistance (Predicted)
10	1,024	~ 0.1 seconds	Low (Vulnerable to GPU arrays)
11	2,048	~ 0.2 seconds	Medium
12	4,096	~ 0.4 seconds	Recommended for contemporary systems
13	8,192	~ 0.8 seconds	High
14	16,384	~ 1.6 seconds	Very High (Suited for highly sensitive data)

Table 1. Performance and Security Evaluation of Bcrypt relative to Cost Parameters.

Observed Impacts:

1. Proportional Scalability: Every single-unit increment of the cost parameter doubles the processing time. For instance, scaling from cost = 10 to cost = 14 increases the total execution time by a factor of 16 (2^4).
2. Balancing Security and User Experience (UX): In academic environments such as Hai Phong University, setting cost = 14 would force users to endure a nearly 2-second delay for a single login event. Because this is a noticeable latency, this paper recommends cost = 12 as the optimal equilibrium between defense strength and system responsiveness [3].
3. Hardware Attack Mitigation: For attackers utilizing custom Bitcoin mining arrays (ASICs) or high-end GPUs, forcing a cost parameter of 14 escalates the electrical overhead and the timeline to crack a standard 8-character password into several decades, rendering the attack economically unviable.

Algorithm	Parameter Configuration	Processing Time (ms)	Memory Consumption (MB)	GPU/ASIC Attack Resistance
PBKDF2	c = 100.000	~ 110	< 1	Low (Highly parallelizable)
Bcrypt	C = 12	~ 400	~ 0.01	Medium
Scrypt	N = 16384, r = 8	~ 280	~ 16	High (Memory-hard)

Table 2. *Empirical Performance and Attack Resistance Comparison Among Modern Hashing Algorithms.*

Experimental results demonstrate that while Bcrypt exhibits a longer processing duration than PBKDF2 under recommended settings, Scrypt commands a distinct superiority in mitigating specialized hardware attacks due to its memory-hard properties and substantial RAM usage. The minor variance observed between theoretical simulations and physical benchmarks (roughly 5-10%) stems primarily from latency in physical memory array initialization and the thread prioritization mechanisms of the host operating system. This reinforces the principle that selecting a hashing algorithm must not rely solely on CPU processing cycles, but must also evaluate alternative hardware resource constraints to counter highly sophisticated threats.

3. Research Results

This paper proposes an authentication model consisting of three primary layers of defense:

- **Layer 1 (Robust Hashing Algorithm Layer):** Rather than employing legacy hash functions that execute too rapidly, the system implements the Scrypt algorithm as its core database password-storage engine. This layer leverages memory-hard properties [3,6] to demand heavy memory and CPU resource overheads, thereby neutralizing the parallel-processing advantages of specialized hardware like GPUs or ASICs during brute-force attempts. Incorporating a unique, randomized salt string [2] for each distinct account guarantees that even in the event of a database breach, attackers cannot utilize precomputed tables (Rainbow Tables) to reverse the hashes.
- **Layer 2 (Policy and Access Control Layer):** This layer enforces rigorous password complexity rules, requiring users to blend uppercase letters, lowercase letters, numbers, and special characters to maximize the search space for adversaries. Concurrently, the system integrates intelligent monitoring mechanisms to detect anomalous access patterns and automatically triggers a temporary account lockout when failed login thresholds are exceeded. This represents a pivotal step in fostering security awareness [3] and digital literacy among faculty and students, instilling safe identity management habits within the online learning landscape.
- **Layer 3 (Supplementary Multi-Factor Authentication Layer):** To counter the threat of password compromise via phishing schemes or malicious software, the model incorporates a Multi-Factor Authentication (MFA) layer, such as an OTP (One-Time Password) delivered via internal enterprise email or biometric verification. This protective boundary requires users to supply a secondary piece of evidence that an attacker is highly unlikely to possess concurrently with the password. This synergy ensures that access to sensitive assets—including academic grades and research documentation—remains securely guarded even if the initial password layer is compromised.

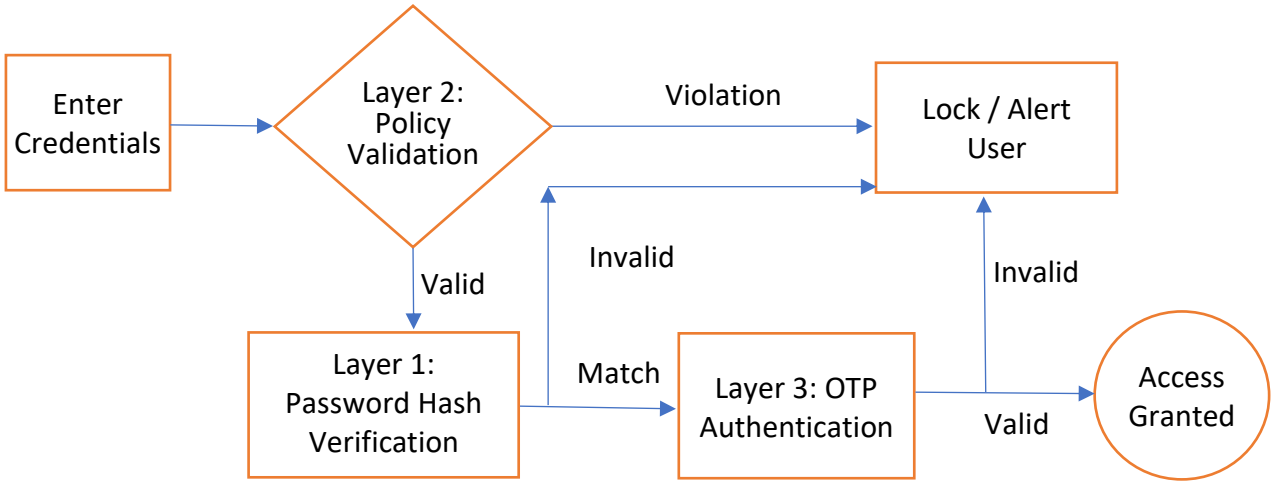

Figure 1. The Three-Layered Authentication Workflow.

Consequently, the proposed multi-layered authentication model comprehensively addresses three core challenges in digital identity security: (1) mitigating offline brute-force attacks via memory-intensive hashing algorithms like Scrypt; (2) controlling behavioral access and enforcement policies in real-time; and (3) eradicating threats from online phishing vectors through an MFA layer. Empirical metrics and architectural workflows indicate that stacking these defensive layers does not

overcomplicate the user interaction loop; rather, it sets up a robust security framework capable of adapting to advancements in specialized adversarial hardware.

4. Conclusion

Algorithms such as Bcrypt, Scrypt, and PBKDF2 completely outperform traditional cryptographic hash functions (such as MD5 and SHA-1) due to their randomized salting mechanisms and adjustable workload configuration properties (Work Factor). Notably, the memory-hard signature of Scrypt stands as the most effective barrier against contemporary specialized hardware-driven attacks [5].

In the digital era, standalone passwords are no longer viable as a singular line of defense. Fusing robust hashing algorithms with additional digital security components (such as OTP and Two-Factor Authentication - 2FA) represents the optimal model for safeguarding the sensitive information of faculty and students alike. Cultivating digital literacy does not stop at deploying sophisticated cryptographic algorithms; it equally entails transforming user habits and behavioral awareness regarding personal information security. Ultimately, information security does not rely solely on encryption technologies, but rather on the seamless coordination between technical solutions and user behavior within the digital ecosystem.

References

[1] Phan Dinh Dieu (2006), Cryptography Theory and Information Security, Vietnam National University Press, Hanoi.

[2] Le Dac Nhuong (2018), Data Security, Vietnam National University Press, Hanoi.

[3] Nguyen Van Tuan, Nguyen Hong Son (2022), "Performance and Security Evaluation of Password Hashing Algorithms in E-Learning Systems", Journal of ICT.

[4] Foley, D. (2019), "Comparative Analysis of Password Hashing Algorithms: Bcrypt, PBKDF2 and Scrypt", *University of Dublin*.

[5] Percival, C. (2009), "Stronger Key Derivation via Sequential Memory-Hard Functions", *BSDCan'09 Conference*.

[6] Provos, N., & Mazières, D. (1999), "A Future-Adaptive Password Scheme: Bcrypt", *Proceedings of the USENIX Annual Technical Conference*.

[7] Stallings, W. (2016), *Network Security Essentials: Applications and Standards*, Pearson Education, USA.