applied sciences

MDPI

Article

RoseCliff Algorithm: Making Passwords Dynamic

Afamefuna P. Umejiaku and Victor S. Sheng *

Computer Science Department, Texas Tech University, Lubbock, TX 79409, USA
* Correspondence: victor.sheng@ttu.edu

Abstract: Authentication in the digital landscape faces persistent challenges due to evolving cyber threats. Traditional text-based passwords, which are vulnerable to various attacks, necessitate innovative solutions to fortify user systems. This paper introduces the RoseCliff Algorithm, which is a dual authentication mechanism designed to enhance resilience against sophisticated hacking attempts and to continuously evolve stored passwords. The study explores encryption techniques, including symmetric, asymmetric, and hybrid encryption, thereby addressing the emerging threats posed by quantum computers. The RoseCliff Algorithm introduces introduces dynamism into passwords that allows for more secured communication across multiple platforms. To assess the algorithm's robustness, potential attacks such as brute force, dictionary attacks, man-in-the-middle attacks, and machine learning-based attacks are examined. The RoseCliff Algorithm, through its dynamic password generation and encryption methodology, proves effective against these threats. Usability evaluation encompasses the implementation and management phase, focusing on seamless integration, and the user experience, emphasizing clarity and satisfaction. Limitations are acknowledged, thus urging further research into encryption technique resilience, robustness against breaches, and the integration of emerging technologies. In conclusion, the RoseCliff Algorithm emerges as a promising solution, thereby effectively addressing the complexities of modern authentication challenges and providing a foundation for future research and enhancements in digital security.

Keywords: encryption; password; security

check for updates

Citation: Umejiaku, A.P.; Sheng, V.S. RoseCliff Algorithm: Making Passwords Dynamic. *Appl. Sci.* **2024**, *14*, 723. https://doi.org/10.3390/app14020723

Academic Editor: Gianluca Lax

Received: 30 November 2023
Revised: 20 December 2023
Accepted: 5 January 2024
Published: 15 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In order to obtain entry to data or a service, it is imperative to initially establish the user's identity through authentication. Authentication stands as a persistent and formidable challenge in the digital realm, thus requiring continuous efforts from security experts to devise progressively sophisticated methods. These methods aim to strengthen users' systems, thus providing robust defense against potential breaches [1]. Typically, when verifying a user's identity, we take into account one of three factors: knowledge-based authentication (something you know), possession-based authentication (something you have), and biometric authentication (something you are). Additionally, we can also incorporate the location and time zone of users (Geo-Time-Zone) as an authentication factor. "Something you know" encompasses passwords and personal identification numbers (PINs). Examples of "something you have" include smart cards and mobile devices. "Something you are" refers to biometric authentication. Authentication systems store identity provided in the user information database within the computing system. During verification, if the credentials entered match with this information stored in the database, the verification process is completed, and the user gets permission to access the system. Using something you know is easy to implement and has the added advantage of being easy to change from one authentication system to another or even in the same authentication at will or after a potential data breach [2–8].

Text-based passwords remain the predominant authentication method, and their susceptibility often arises from human-related factors, thereby introducing diverse security risks [9]. To tackle these challenges, organizations and security experts advocate

for the adoption of password strength meters; the creation of memorable passwords in Sadat et al. [9] proposed generating memorable passwords using the user's input like, time and location data, and the enforcement of comprehensive password policies encompassing criteria such as complexity, length, expiration, and periodic changes. Incorporating unique combinations of letters, numbers, and symbols further enhances security, thus complicating malicious actors' attempts to guess passwords [10,11]. Unfortunately, the relentless march of technological progress continuously equips hackers with an ever-expanding array of tools and tactics to exploit vulnerabilities, thus often evading evolving safety protocols. The escalating security threats extend beyond user behaviors, as the proliferation of computational resources empowers malicious actors to launch various attacks against authentication systems, including sophisticated strategies like relentless brute force attacks, cunning dictionary attacks, man-in-the-middle (MitM) attacks, and artificial intelligence (AI)- and machine learning-based attacks [3,12,13]. Ding and Horster [14] classify password guessing attacks into three categories: detectable online password guessing attacks, undetectable online password guessing attacks, and offline password guessing attacks. Although password attacks can manifest online or offline, the latter proves most advantageous for attackers, thus relying solely on recorded messages from successful authentication protocol runs.

Text-based passwords remain the predominant authentication method, and their susceptibility often arises from human-related factors, introducing diverse security risks [9]. To tackle these challenges, organizations and security experts advocate for the adoption of password strength meters, as well as the enforcement of comprehensive password policies encompassing criteria such as complexity, length, expiration, and periodic changes for the creation of memorable passwords. Sadat et al. [9] propose generating memorable passwords using the user's input, such as time and location data. Incorporating unique combinations of letters, numbers, and symbols further enhances security, thus complicating malicious actors' attempts to guess passwords [10,11]. Unfortunately, the relentless march of technological progress continuously equips hackers with an ever-expanding array of tools and tactics to exploit vulnerabilities, thus often evading evolving safety protocols. The escalating security threats extend beyond user behaviors, as the proliferation of computational resources empowers malicious actors to launch various attacks against authentication systems. These include sophisticated strategies like relentless brute force attacks, cunning dictionary attacks, man-in-the-middle (MitM) attacks, and artificial intelligence (AI)- and machine learning-based attacks [3,12,13]. In their publication, Guan and Chen [14] demonstrated that password attacks can manifest as either offline or online threats. They introduced a novel verification scheme designed to thwart online password guessing attacks. The proposed solution evaluates the entropy of user-entered passwords and deems the user authentic only when the entropy remains below a predefined threshold.

In the contemporary landscape of heightened communication across diverse authentication systems, a noteworthy scenario arises wherein a user engaged with a particular application necessitates authorization from a distinct entity [15]. In this context, the imperative to bolster the security framework becomes evident, thereby mandating a dynamic alteration of user authentication details. The rationale behind this approach lies in the mitigation of potential vulnerabilities that may arise from leaks within the system. Therefore, it becomes paramount to adopt techniques that enable the tracking of such leaks, thereby enhancing the overall resilience of the authentication infrastructure. A proactive response to this security challenge involves the development of a password-based authentication system that seamlessly integrates with the broader authentication architecture. This integration facilitates the dynamic transformation of a user's password, thereby introducing an additional layer of defense against unauthorized access. Moreover, the fortification of the password itself through advanced encryption techniques and stringent security measures serves as a deterrent to malicious entities seeking to exploit vulnerabilities in the authentication process.

In this paper, our objective is to enhance password security by rendering encrypted or hashed passwords dynamic and incorporating dual authentication without imposing an additional burden on users. The structure of our exploration is as follows:

1. **Reviewing Existing Encryption Techniques:** We provide an overview of existing encryption methods and techniques.
2. **RoseCliff Algorithm:** Our proposed algorithm has been crafted with the specific goal of infusing dynamism into current encryption techniques and introducing the innovative concept of dual authentication, all of which are achieved seamlessly from a single password.
3. **Implementation:** We implement the RoseCliff algorithm to evaluate its effectiveness in imbuing passwords with dynamism. This step is crucial for assessing the practical implications of our proposed approach.
4. **Adoption Benefits:** We delve into the potential advantages of adopting the RoseCliff algorithm, thus exploring how its implementation can substantially elevate security measures and enhance user experiences across a spectrum of authentication systems.

Our overarching goal is to cultivate a more secure and user-friendly authentication landscape. Through our research and proposed RoseCliff algorithm, we aim to contribute to a safer digital experience for all users, thereby ensuring that robust security measures do not come at the expense of user convenience.

2. Encryption

Encryption comes in two main forms: symmetric and asymmetric. Symmetric encryption offers speed, efficiency, and ease of implementation, thus making it well-suited for encrypting large data volumes and commonly used for password encryption. However, it presents challenges in key distribution, scalability as participants increase, and lacks inherent authentication. In contrast, asymmetric encryption (public key encryption) addresses the key distribution problem, provides authentication, and supports nonrepudiation. Yet, it demands more computational resources, longer key lengths for equivalent security, and involves complex key management [16,17]. Presently, hybrid encryption combines the advantages of both symmetric and asymmetric encryption, thus ensuring secure data protection. It starts with asymmetric encryption for secure key exchange, thereby subsequently transitioning to symmetric encryption for efficient data encryption and decryption. Hybrid encryption is widely applied in secure communication systems, thus striking a balance between security and performance. Effective data encryption has proven resilient against most attack forms, as the resulting ciphertext is typically lengthy and randomized, thereby thwarting various attacks, including those leveraging machine learning [18,19].

2.1. Symmetric Encryption

Symmetric encryption, also referred to as secret key or private key encryption, utilizes a single shared key for both encrypting and decrypting data. This shared secret key is known to both the sender and receiver and is employed to transform plaintext into ciphertext (encryption) and vice versa (decryption). One prominent example of symmetric encryption is the Advanced Encryption Standard (AES), which was endorsed by the National Institute of Standards and Technology (NIST) in 2001. The AES, replacing the older Data Encryption Standard (DES) and Triple DES (3DES), operates on 128-bit data blocks, thus employing transformation rounds like SubBytes, ShiftRows, MixColumns, and AddRoundKey. The number of rounds varies based on the key size (AES-128, AES-192, or AES-256), thereby offering differing levels of security [20]. Notable characteristics of symmetric encryption include:

1. **Shared Key**—Both the sender and receiver possess and safeguard the same secret key, thus ensuring its confidentiality.
2. **Efficiency**—Symmetric encryption typically outperforms asymmetric encryption (public key encryption) in terms of computational efficiency due to its simpler mathematical operations.
3. **Confidentiality**—Symmetric encryption ensures that unauthorized parties cannot decipher encrypted data unless they possess the secret key.

4. **Data Integrity**—While symmetric encryption primarily focuses on confidentiality, additional techniques like message authentication codes (MACs) or digital signatures are used to guarantee data integrity and authentication.

The AES is recognized for its high security and has been adopted globally as an encryption standard, thereby serving various industries and countries to secure data and communications. It is a foundational component of modern cryptography, thereby offering robust security and versatility in preserving data confidentiality and integrity across numerous applications [17,20,21].

2.2. Asymmetric Encryption

Asymmetric encryption, often termed public key encryption, plays a fundamental role in securing digital data exchanges. It stands apart from symmetric encryption, which relies on a single shared key for both encryption and decryption processes. Instead, asymmetric encryption harnesses a dual key mechanism: a public key for encryption and a private key for decryption. In this framework, users create a key pair—consisting of a publicly shared key and a confidential private key. This setup empowers anyone to encrypt messages using the recipient's public key, while only the recipient, holding the corresponding private key, can decrypt them. Its versatile applications span secure communication (e.g., emails and financial transactions), digital signatures (verification of document authenticity), key management (secure key distribution without divulging secrets), and secure authentication methods (e.g., SSH and SSL/TLS protocols for web browsing) [22–26]. Asymmetric encryption encompasses a spectrum of algorithms and methods, each with unique strengths and applications. Among the prominent choices are RSA (Rivest–Shamir–Adleman), known for its reliance on prime numbers, and Diffie–Hellman, a key exchange protocol enabling secure symmetric encryption. Additionally, the Digital Signature Algorithm (DSA), rooted in modular arithmetic and discrete logarithms, is widely recognized for creating digital signatures. Noteworthy variants include ECDSA (Elliptic Curve Digital Signature Algorithm) [23,27], ElGamal (ElGamal Encryption Algorithm) [24], and ECDH (Elliptic Curve Diffie-Hellman) [26].

2.3. Hybrid Encryption

Hybrid encryption combines the strengths of symmetric and asymmetric encryption, thus striking a balance between security and efficiency. In this method, two parties establish a secure communication channel for key exchange, typically using asymmetric encryption. They exchange a shared secret key, which then facilitates a switch to efficient symmetric encryption for data encryption and decryption. Symmetric encryption algorithms, such as AES, are used for the actual data protection, and the shared secret key remains confidential. This approach ensures secure and efficient data transmission and storage, thereby overcoming the limitations of pure symmetric or asymmetric encryption methods [19]. Hybrid encryption is crucial in modern cryptography for several reasons. Researchers have employed hybrid models fusing encryption algorithms or hashing with honey encryption to create a two-layer protection mechanism. This approach optimizes security and efficiency by harnessing the strengths of both encryption paradigms and scales effectively for varying data sizes. Additionally, it ensures perfect forward secrecy, thus bolstering security with new symmetric keys generated for each session. Hybrid encryption promotes compatibility across systems and devices, thus offering resilience against emerging threats, including those from quantum computing. In summary, it is a versatile cryptographic technique playing a crucial role in efficiently securing data across a broad range of applications [28–30].

2.4. Quantum-Resistant Cryptographic Algorithms

According to NIST, a part of the U.S. Department of Commerce responsible for setting standards in digital security, existing encryption technologies face threats from quantum computers, as these quantum machines could undermine the security of daily digital

activities. They chose four encryption algorithms as the postquantum cryptographic standard, which is set to be finalized in about two years. These algorithms focus on two primary encryption functions: general encryption for securing data over public networks and digital signatures for identity verification. They were collaboratively developed by experts from various countries and institutions. NIST has picked the CRYSTALS-Kyber algorithm for general encryption, which is known for its compact keys and operational efficiency. In the realm of digital signatures, NIST has selected three algorithms: CRYSTALS-Dilithium, FALCON, and SPHINCS+ (pronounced "Sphincs plus"). CRYSTALS-Dilithium is the primary choice, FALCON suits smaller signatures, and SPHINCS+ serves as a unique backup due to its distinct mathematical approach. The three algorithms are rooted in structured lattices, while SPHINCS+ relies on hash functions. Four additional algorithms are under consideration for general encryption, each with distinct approaches [31].

3. RoseCliff Algorithm

In this section, we introduce our innovative algorithm crafted to enhance the resilience of authentication systems. This algorithm not only makes password dynamic but can be used to build an advanced trigger mechanism. Termed the RoseCliff Algorithm, its primary objective is to implement dual authentication without adding any extra complexity or burden on users, all while ensuring the continuous evolution of stored passwords in hashed forms called ciphertext in cryptography. The algorithm itself can be broken down into five distinct phases: Input Detection, Password Splitting, One Time Number, Dynamic Password/Ciphertext Password Generation , and Authentication.

A summary of the steps in the RoseCliff Algorithms is as follows:

1. Input Detection
   - Ensure input device used to enter at least part of a the password.
2. Password Splitting
   - Split password into two segments.
   - Identify and record the positions of characters prior to their separation.
3. One Time Number
   - Authentication system generates one-time number.
4. Dynamic Password/Ciphertext Password Generation
   - Utilize public–private key encryption to generate a randomizing value using the confidential one-time number and a segment of the password.
   - Merge the randomizing value with the portion of the password not used in randomization, thus generating a temporary password.
   - Encrypt and send the temporary password.
5. Authentication
   - During authentication, decrypt the encrypted password, as well as the previously stored password to be used for verification.
   - Verify that the nonrandomized portion of the decrypted password matches that of the nonrandomized portion of the previously stored password.
     (a) If there is a match, the system proceeds to check if the randomization value at the authentication system matches that obtained from the decrypted password.
        i. If there is a match, the user is authenticated.
        ii. If there is not a match, the authentication system signals a potential breach.
     (b) If there is no match, the authentication system records a failed attempt. If the number of failed attempts reaches a preset threshold, the authentication system activates its security measures.
   - If the user is authenticated, update the previously stored password to a new one.

4. Implementation

In this implementation, we have incorporated our proposed algorithm in a way that is both comprehensible and replicable. We have employed well-established techniques to provide a foundational guide, thus enabling cybersecurity to construct robust systems based on the RoseCliff algorithm. Our implementation is meticulously divided into five distinct phases, thereby aligning with the sequential steps of the RoseCliff algorithm.

4.1. Input Detection

In order to thwart users from storing passwords in vulnerable locations and merely copying and pasting or relying solely on password managers, the RoseCliff algorithm mandates that a portion of the password be manually typed. This ensures human input even when utilizing password management techniques like using password manager. In our implementation, numerical values in a user's password must be entered via an input device, while non-numerical values can be inputted through any means. When a user is prompted to enter their password, the system captures all the keyboard inputs and stores numerical entries. It then compares these numeric inputs with the numerical values extracted from the password. If the numeric inputs extracted from the password and those entered from an input device fails to match, the system prompts the user to enter any missing numeric values in the password and subsequently verifies their accuracy.

Algorithm 1 outlined below enforces this security measure.

Algorithm 1 Numerical Input Detection

```
Ensure: KeyNumbers ← []
  KeyCount ← 0
  PasswordNumbers ← []
  while Password_Entry = True do
      if Key_Press is Detected then
          if Entry is Numeric then
              KeyCount ← KeyCount + 1
              KeyNumber is updated
          end if
      end if
  end while
  for char in Password do
      if char is Numeric then
          <PasswordNumbers IS updated>
      end if
  end for
  if KeyCount = count of PasswordNumbers then
      if KeyNumbers = PasswordNumbers then
          StartAuthentication
      end if
  else if KeyCount < count of PasswordNumbers then
      Request Numbers be Typed
  end if
```

4.2. Password Splitting

Moving on to the second phase of the RoseCliff algorithm, we divide the password into two segments. In this process, one part undergoes randomization, while the other remains unaltered. For ease of implementation and clarity, we chose to split the password by isolating the numerical values from all other characters. This separation sets the stage for subsequent operations in the algorithm, thereby contributing to the overall security and uniqueness of the password transformation.

4.3. One-Time Number

An automated one-time number serves as a dynamic authentication mechanism for a user, which is valid for a single transaction or login session. The server enhances security by generating a unique code that changes rapidly. When an unauthenticated user attempts system access or a transaction, the network server's authentication manager generates a number using a one-time password algorithm. Unlike conventional one-time passwords transmitted over a network, this number remains confidential on the server.

In our implementation, we employ the 'Sieve of Atkin' [32] for prime number generation due to its improved time and memory efficiency. We compile a list of prime values slightly exceeding 10 million, from which we extract a million primes, each with at least eight digits. This extensive prime number set plays a crucial role in the password encryption process. The dynamically chosen prime number adds an extra layer of security, as it is not a static constant, thus making it challenging for attackers to predict or reuse. Utilizing a one-time prime number significantly reduces the window of opportunity for the potential interception and exploitation of cryptographic data.

4.4. Dynamic Password/Ciphertext Generation

To ensure a dynamically changing encrypted password, the RoseCliff algorithm employs a randomizing value to alter the user-entered password before encrypting and transmitting it over a network. The algorithm suggests the use of public–private keys, where the one-time number and a segment of the password serve as private keys. A value is generated to randomize the user-entered password, and this value is intended to be combined with common language characters before being encrypted and sent over the network. Notably, the algorithm prioritizes the nontransmission of the one-time number over the internet to prevent man-in-the-middle attacks. Additionally, it infuses common characters to thwart dictionary attacks.

In our implementation, we use the extracted password segment as the client-side private key and the one-time number as the server or authentication-side private key. Employing the Diffie–Hellman key exchange, we generate a value, convert it to base 16, hash it, and use the indexing locations of the numbers from the original password (NumIndex) to infuse the ten most commonly used English characters. Subsequently, we merge the other segment of the password with the newly generated value using the index locations, and all other values are stored at the end to create a temporary password. Finally, we employ the AES-256 encryption technique to encrypt the temporary password, where AES-256 stands for the Advanced Encryption Standard with a 256-bit key length, is renowned for its robustness and ability to withstand contemporary hacking challenges. In today's dynamic and ever-evolving digital landscape, where cyber threats continue to grow in complexity and sophistication, AES-256 serves as a stalwart guardian against malicious intrusions and data breaches. Its 256-bit key length provides an exceptionally large number of possible combinations, thus rendering brute force attacks nearly infeasible. Moreover, it benefits from extensive scrutiny by the cryptographic community, which has further validated its resilience against known vulnerabilities.

Algorithm 2 outlined below enforces this security measure.

4.5. Authentication Process

The authentication process in RoseCliff comprises two essential steps due to the introduction of dynamism in passwords before encryption. Initially, the received password over the network is decrypted using the same techniques employed just before its transmission. Subsequently, the randomized password segment is distinguished from the nonrandomized segment. The randomized portion is utilized to compare values, which are also generated on the server side. In parallel, the nonrandomized segment undergoes decryption of the previously stored password for comparison. Authentication is granted only when both comparisons are valid as seen in Algorithm 3.

Algorithm 2 Dynamic Password/Ciphertext Password Generation

Server and Client
$Pr \leftarrow$ server generates a **Prime number** and shares with client
$PP \leftarrow$ server generates a **Primitive-Prime** number and shares with client
Server Side
$OTN \leftarrow$ server generates a one time number
$SV \leftarrow (PP^{OTN}) \mod Pr$
$RandomValue \leftarrow (SV^{OTN}) \mod Pr$
Client Side
$Ps \leftarrow$ Password segment
$CV \leftarrow (PP^{Ps}) \mod Pr$
$RandomValue \leftarrow (CV^{Ps}) \mod Pr$
$RandomHex \leftarrow$ convertToBase16$(RandomValue)$
$RandomPass \leftarrow RandomHex([a,b,c,d,e,f] \leftarrow [e,t,a,i,o,n,s,h,r])$
$DynamicPassword \leftarrow$ **Map**$(Password, RandomPass, NumIndex)$
procedure ENCRYPTPASSWORD(DynamicPassword)
    $key \leftarrow$ GenerateAESKey() ▷ Generate a secure AES key
    $iv \leftarrow$ GenerateInitializationVector() ▷ Generate an initialization vector
    $encryptedPassword \leftarrow$ AES-256-Encrypt(password, key, iv)
    return $encryptedPassword$
end procedure

Algorithm 3 Authentication Process

$EncP \leftarrow$ Encryted Password ▷ Recently sent over network
$OEncP \leftarrow$ Old Encryted Password ▷ Previously store by Authentication system
$NumIndex \leftarrow$ Indexing locations of the numbers from the original password
$RandomValue \leftarrow$ Previously calculated during Dynamic Password generation
$RandomHex \leftarrow$ convertToBase16$(RandomValue)$
$RandomPass \leftarrow RandomHex([a,b,c,d,e,f] \leftarrow [e,t,a,i,o,n,s,h,r])$
procedure DECRYPTPASSWORD(EncP)
    $decryptedPassword \leftarrow$ AES-256-Encrypt(EncP, key, iv)
    return $decryptedPassword$
end procedure
$PasswordNew, RandomPassNew \leftarrow$ **Map**$(decryptedPassword, NumIndex)$
if $RandomPass = RandomPassNew$ **then**
    Verification1 $\leftarrow$ **true**
else
    Verification1 $\leftarrow$ **false**
end if
procedure DECRYPTPASSWORD(OEncP)
    $decryptedPassword2 \leftarrow$ AES-256-Encrypt(OEncP, key, iv)
    return $decryptedPassword2$
end procedure
$Password2Old, RandomPassOld \leftarrow$ **Map**$(decryptedPassword, NumIndex)$
if $PasswordNew = Password2Old$ **then**
    Verification2 $\leftarrow$ **true**
else
    Verification2 $\leftarrow$ **false**
end if
if $RandomPassNew = RandomPassOld$ **then**
    Verification2 $\leftarrow$ **true**
else
    Verification2 $\leftarrow$ **false**
end if
if $Verification1$ **is true and** $Verification2$ **is true then**
    Authenticated
end if

In this particular implementation, the verification process initiates with the decryption of the password received over the network. By utilizing index locations, the nonrandom-

ized password segment is then separated from its randomized counterpart. Subsequently, employing the NumIndex, the value obtained from the Diffie–Hellman Algorithm undergoes hashing and mapping. This result is matched against the randomized password segment to ascertain its validity. Concurrently, the previously stored password is decrypted, the NumIndex is applied to isolate the nonrandomized password segment, and a comparison is conducted. The user is authenticated only when both comparisons yield successful outcomes.

5. Experimental Results and Discussion

Table 1 displays dynamic passwords derived from six examples of commonly used passwords that are typically deemed unsafe [33]. In an effort to assess their dynamic nature, numbers were systematically added at various locations within these passwords. An interesting observation emerged when analyzing the impact of these additions, wherein appending numbers solely at the end of the password facilitated a more straightforward deduction of the original password structure compared to when the numbers were added in the middle. However, introducing numbers at different locations within the password rendered the task of deducing the original password more challenging. This held true even when employing one of the most commonly used phrases exemplified by 'ILoveYou'.

Table 1. Dynamic passwords derived from common password.

Common Password	Password Used	Dynamic Password 1	Dynamic Password 2	Dynamic Password 3	Dynamic Password 4	Dynamic Password 5
Qwerty	123Qwerty	3h68419Q werty	610o57oQ werty	20a7204Q werty	3si9t7iQ werty	2s774Q werty
Password	Password 1234	Password 2h5i718	Password 1416nin	Password 1h928i6	Password 4h7o4o	Password 259h9t5
Monkey	59Mon34 key23	27oMon3n key5h	1nMon23k ey45	32sMons2k ey9t	17nMonon key74	1unMonon keye3
ILoveYou	I1Love2y ou3	ItrLovet 8you90	I3e1Love 27you64	I382Love 0ayout6	I151Love e1yours	I27Love3 5youae
Princess	Prin1234 5cess	Prin6736 18cess	Prin32sa ratcess	Prin5044 eo0cess	Prin21e8 e91cess	Prin3e5n 8e3cess
Password	Pa12ss34 wo#rd	Pa262ssr6 59wo#rd	Pa22r3ss92 0wo#rd	Pa2nt1ss34 5wo#rd	Pa41i1ssao awo#rd	Pa3n34ssa1 awo#rd

Table 2 presents the ciphertexts, and, as anticipated, they exhibited distinct and dissimilar patterns. The absence of any trace indicating a common origin implies that these ciphertexts could not have originated from the same password. This reinforces the effectiveness of the dynamic password generation approach in enhancing security and obscuring the underlying password structure.

The outcomes of our analysis vividly illustrate the algorithm's success in introducing dynamism to passwords. Notably, this dynamic infusion has contributed to a notable increase in both the length and randomness of the passwords before encryption. This enhancement plays a pivotal role in fortifying the security of the authentication system.

By extending the length and incorporating greater randomness, the algorithm effectively mitigates the vulnerabilities associated with static and easily guessable passwords. This proactive measure serves as a robust defense against various unauthorized access attempts, thus bolstering the overall security posture of the authentication system. The deliberate efforts to make passwords more intricate and less predictable contribute significantly to the resilience of the system against potential security threats, thereby reflecting the algorithm's efficacy in promoting a secure authentication environment.

Table 2. Ciphertexts derived from common passwords.

Common Password	Password Used	Cipher Text 1	Cipher Text 2	Cipher Text 3	Cipher Text 4	Cipher Text 5
Qwerty	123Qwerty	4a63ae62 b19dc88a 164a7…	cc195b49 19ebd5e0 4f3f1…	3e02c66a 516e4a72 aaa74…	7d063ff5 f37e120d d9b73…	3bc0cc76 09afb5fe 005ea…
Password	Password 1234	20efde97 c15dbed2 098ba…	c66a0f0d da153a2b c7d13…	1d76ab92 d7ae954f 17007…	b4451f018 ac483b31 a9578…	df9ded16 ad1c2a97 875f7…
Monkey	59Mon34 key23	e403122c 143531c4 b6c17…	cde4b00c 147034ea 8c0f6…	c4f531df 8c654382 d3cf9…	a6b8f943 ce466abe ea540…	f95d7366 32f36009 5e060…
ILoveYou	I1Love2y ou3	bb95606a d4fd22f3 ed545…	6c4afda7 a35eb797 d067d…	1d6a5f94 e63aa7ab 93ccc…	6f8e2580 1213ee58 2991c…	9dd26405 ce515966 043ff…
Princess	Prin1234 5cess	54172606 cc32e13d 32925…	026c873e 19b7ea79 4d2bc…	64533a24 f61e504a 450c3a…	dd75ec3d 132dcc7d 48a14…	0ab0bca6 f3b14f97 58a69…
Password	Pa12ss34 wo#rd	15540021 93db1451 dd7b1…	63795fcd 67b1242d b50a4…	d47f0851 185d7175 64ccf…	bab4108d bc65600a c704e…	a54f7501 5dc4ceee a4a28…

5.1. Potential Attacks

To understand the algorithm's performance with respect to potential attacks, we subjected the algorithm to various attack vectors with the aim of assessing its robustness and resilience in the face of security threats. In the attack scenarios, we scrutinized the algorithm's response to common threats such as brute force attacks, dictionary attacks, man-in-the-middle (MitM) attacks, and machine learning-based attacks. This rigorous testing allowed us to evaluate the algorithm's ability to withstand adversarial strategies and adapt dynamically to evolving attack methodologies. Through this in-depth analysis, we aim to provide a comprehensive understanding of the algorithm's performance in real-world scenarios, thereby allowing for informed assessments of its security capabilities and potential areas for further refinement.

5.1.1. Brute Force and Dictionary Attacks

Alkhwaja et al. [34] conducted experiments employing both brute force and dictionary attack techniques. They discovered that the brute force method efficiently unlocked relatively short passwords, typically six to seven characters in length. However, brute force was less effective in cracking passwords consisting of eight or more characters. Interestingly, they found success in using dictionary attacks to parallelize password cracking efforts for longer passwords.

Combating dictionary attacks, though similar to brute force attacks but more sophisticated—as can be seen in Widianto et al. [35]—often requires multiple attempts and can be frustrated when passwords combinations are complex. Recognizing the importance of password length, in our incorporation of dynamism, we ensured an increase in password length and added extra computational steps as suggested by Chakrabarti & Singhal [36]. To further evaluate the robustness of our approach, we subjected the passwords to Passfault—Password Strength Tester [37]. This comprehensive test assessed the strength of both the original passwords and the dynamic passwords, thereby providing insights into the overall security of our password system.

Passfault was utilized to assess the strength of the passwords listed in Table 1, thereby determining the time required for potential cracking. The results can be seen in Table 3.

Table 3. Time to crack passwords in Table 1.

Common Password	Password Used	Dynamic Password 1	Dynamic Password 2	Dynamic Password 3	Dynamic Password 4	Dynamic Password 5
<1 s	1 s	20 h	1 day	1 day	1 day	17 min
<1 s	2 s	1 day	3 h	3 h	3 h	1 day
<1 s	4 days	10 years	2 years	31 years	17 days	11 months
<1 s	7 days	20 years	Centuries	Centuries	5 years	20 years
<1 s	1 month	1 year	74 years	Centuries	Centuries	Centuries
<1 s	Centuries	Centuries	Centuries	Centuries	Centuries	Centuries

Notably, base passwords often derived from commonly used ones could be cracked in under a second. Additionally, commonly used passwords augmented with numbers were generally more vulnerable to cracking compared to dynamically generated passwords. This underscores one of the advantages of employing dynamic passwords. The resilience of dynamic passwords varies based on the placement of numbers within commonly used passwords. This emphasizes the importance of the implementation of the RoseCliff method and the strength of a user's password. Specifically, appending or prepending numbers alone can lead to quicker cracking, ranging from 17 days to 31 years. Conversely, incorporating numbers in multiple locations significantly prolongs the time required for potential cracking, thus potentially extending it to centuries.

5.1.2. Man-in-the-Middle (MitM) Attacks

The RoseCliff algorithm goes beyond the conventional approach of merely transmitting a dynamically generated password after encryption over the network. This distinctive feature ensures that even if an eavesdropper is actively monitoring network traffic or gains unauthorized access to the password file, deciphering the original password remains an insurmountable challenge. One of the unique strengths of the RoseCliff algorithm lies in its utilization of a one-time number for randomization purposes. This implies that the dynamic password, which is derived from the amalgamation of the original password and this one-time number, undergoes constant and unpredictable changes. The dynamic nature of the password adds an extra layer of complexity, thus significantly heightening the difficulty level for any potential adversaries attempting to compromise the security of the system.

In essence, the algorithm's design not only focuses on secure transmission but also incorporates a dynamic element that contributes to the continual transformation of the password. This continual change ensures that the authentication process remains robust against various forms of cyber threats, thus making it inherently more challenging for malicious actors to intercept, decipher, or exploit the cryptographic data being transmitted.

5.1.3. Machine Learning-Based Attacks

In their comprehensive exploration of the intersection between deep learning algorithms and cybersecurity, Dixit and Silakari [38] delved into the profound implications of these algorithms on the security landscape. Meanwhile, Hitaj et al. [39] made significant strides in enhancing password guessing attacks, thereby demonstrating how advancements in artificial intelligence could effectively generate passwords not confined to conventional dictionaries. They proposed that adherence to certain setting rules, coupled with the integration of artificial intelligence, could substantially enhance the potency of password guessing attacks. Their research underscored the notion that employing a combination of techniques yields the most effective strategy for password guessing.

Machine learning techniques, which excel at discerning patterns, face limitations when applied to encrypted text due to its inherently random nature. Despite their inherent

limitations, these techniques can still be leveraged to refine guessing attacks by using cryptanalysis to extract decryption keys from ciphertext blocks and find solutions in the search space [40]. However, the RoseCliff algorithm introduces a distinctive layer of security by necessitating feedback from the authentication system. As a consequence, potential attacks using machine learning must be conducted online, allowing the authentication system to promptly detect and respond to any suspicious activities.

This paradigm shift highlights the resilience of the RoseCliff algorithm against advanced password guessing attacks, as it necessitates a real-time interaction with the authentication system, thereby fortifying the defense against evolving cybersecurity threats. The synergistic approach of leveraging artificial intelligence in conjunction with traditional security measures underscores the dynamic nature of modern cybersecurity strategies.

5.2. Usability Issues and Considerations

The evaluation of the usability of the RoseCliff Algorithm entails a thorough examination structured into two key stages: the implementation and management phase, which focuses on integrating the algorithm into the authentication system, and the overarching user experience, which considers various associated factors.

5.2.1. Implementation and Management

The implementation of the RoseCliff Algorithm aims to provide a user-friendly process. Its adaptability to diverse environments and its simple implementation underscore its relevance and usefulness in real-world applications. Moreover, managing the algorithm entails assessing whether authentication systems that incorporate it require minimal maintenance while enabling easy adjustments and scalability to address the ever-evolving landscape of security requirements. This focus on manageability contributes significantly to ensuring that the algorithm remains accessible and functional within varying operational contexts.

5.2.2. User Experience

Beyond the technical intricacies, the user experience is pivotal in influencing the algorithm's usability. The algorithm places minimal to no extra burden on users, as it only requires them to manually enter at least a portion of their passwords.

6. Limitations and Future Recommendations

In the course of this study, it is imperative to acknowledge certain limitations and propose avenues for further exploration and enhancement of the RoseCliff Algorithm. The study assumed the relative security of the encryption techniques employed for both symmetric and asymmetric aspects of the algorithm. While this assumption provides a foundational framework, future research should delve deeper into the resilience of these encryption techniques. Robustness against potential breaches and the algorithm's ability to safeguard users even in the event of a breach warrant thorough investigation.

Multifactor authentication (MFA) and two-factor authentication (2FA) constitute security processes wherein users furnish two distinct authentication factors, generally categorized as knowledge factors (something the user knows), possession factors (something the user has), and inherence factors (something the user is) [1,41]. In the specific context discussed here, only the knowledge factor (what the user knows) was employed. While this may be considered a less secure technique, research indicates that users generally prefer to minimize the frequency of providing input multiple times.Consequently, security professionals often restrict such requests to instances where they are deemed necessary, thus balancing security measures with user preferences [3]. Our study predominantly concentrated on the aspect of randomizing user passwords, thereby incorporating evolving techniques such as model building to assist users in crafting more effective passwords for use with the algorithm. However, further research avenues could explore additional dimensions, including the integration of emerging technologies and methodologies to enhance the overall effectiveness and security of password randomization.

In the realm of future research, there is a need to delve into the development and detection of potential adversarial attacks, particularly in scenarios where the password undergoes segmentation, infusion with common characters, and the tracking of previously used dynamic passwords. Understanding the algorithm's vulnerability to adversarial exploits and developing countermeasures will be instrumental in fortifying its robustness and ensuring a resilient defense against sophisticated cyber threats.

7. Conclusions

Authentication in the digital realm faces relentless challenges posed by an ever-evolving landscape of cyber threats. Text-based passwords, despite their ubiquity, remain susceptible to a myriad of attacks, ranging from traditional brute force to sophisticated machine learning-based strategies. The quest for a robust authentication system necessitates innovative approaches that not only address human-related vulnerabilities but also counteract advanced hacking attempts.

This study introduces the RoseCliff Algorithm, which is a solution designed to elevate the resilience of authentication systems and encourage secure crosscommunication across different authentication systems. The RoseCliff Algorithm ensures the dynamic transformation of stored passwords, thus enhancing their security through hashed forms known as ciphertext in cryptography. The algorithm's performance against potential attacks was thoroughly evaluated, including brute force attacks, dictionary attacks, man-in-the-middle attacks, and machine learning-based attacks. The RoseCliff Algorithm demonstrates a multifaceted defense, thereby thwarting various attack vectors and introducing a novel approach to password security. In the realm of encryption, the algorithm employs both symmetric and asymmetric techniques, thus ensuring a balance between efficiency and security.

Beyond technical intricacies, the usability of the RoseCliff Algorithm was scrutinized in two key dimensions: implementation and user experience. This dual-faceted evaluation ensures that the algorithm not only integrates seamlessly into diverse environments but also offers an intuitive and satisfying experience for end users. However, acknowledging certain limitations, such as assumptions about encryption technique security, opens avenues for future research. Deeper investigations into encryption resilience, the exploration of additional dimensions for password randomization, and the development of countermeasures against potential adversarial attacks are proposed for future enhancements of the RoseCliff Algorithm.

In conclusion, the RoseCliff Algorithm presents a pioneering step toward fortifying authentication systems in the face of evolving cyber threats. Its innovative approach, combining advanced encryption, dynamic password evolution, and multifaceted defense mechanisms, positions it as a promising solution for the contemporary challenges of digital security.

Author Contributions: Conceptualization, A.P.U. and V.S.S.; Methodology, A.P.U.; Validation, A.P.U.; Formal analysis, A.P.U.; Investigation, A.P.U.; Data curation, A.P.U.; Writing—original draft, A.P.U.; Writing—review & editing, A.P.U.; Visualization, A.P.U.; Supervision, V.S.S.; Project administration, V.S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available at https://github.com/afamumejiaku/RoseCliff-Algorithm.git (accessed on 1 January 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Papathanasaki, M.; Maglaras, L.; Ayres, N. Modern authentication methods: A comprehensive survey. *AI Comput. Sci. Robot. Technol.* **2022**, 1–24. [CrossRef]
2. Lal, N.A.; Prasad, S.; Farik, M. A Review Of Authentication Methods. *Int. J. Sci. Technol. Res.* **2016**, *5*, 246–249.

3. Konoth, R. K.; van der Veen, V.; Bos, H. How anywhere computing just killed your phone-based two-factor authentication. *Financ. Cryptogr. Data Secur.* **2017**, 405–421. [CrossRef]
4. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI). Available online: https://ieeexplore.ieee.org/xpl/conhome/8386711/proceeding (accessed on 1 January 2024).
5. Towhidi, F.; Manaf, A.A.; Daud, S.M.; Lashkari, A.H. The Knowledge Based Authentication Attacks. In Proceedings of the International Conference on Security and Management (SAM), Las Vegas, NV, USA, 18–21 July 2011.
6. Waruwu, B.K.; Tandoc, E.C, Jr.; Duffy, A.; Kim, N.; Ling, R. Telling lies together? Sharing news as a form of social authentication. *New Media Soc.* **2020**, *23*, 2516–2533. [CrossRef]
7. Taher, K.A.; Nahar, T.; Hossain, S.A. Enhanced Cryptocurrency Security by Time-Based Token Multi-Factor Authentication Algorithm. In Proceedings of the 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 10–12 January 2019; pp. 308–312. [CrossRef]
8. Mohammed, A.H.; Dziyauddin, R.A.; Latiff, L.A. Current multi-factor of authentication: Approaches, requirements, attacks and challenges. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*. [CrossRef]
9. Sadat, S.E.; Lodin, H.; Ahmadzai, N. Highly secure and easy to remember password-based authentication approach. *J. Res. Appl. Sci. Biotechnol.* **2023**, *2*, 134–141. [CrossRef]
10. Bonneau, J.; Herley, C.; Oorschot, P.C.; Stajano, F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012. [CrossRef]
11. Umejiaku, A.P.; Dhakal, P.; Sheng, V.S. Balancing password security and user convenience: Exploring the potential of prompt models for password generation. *Electronics* **2023**, *12*, 2159. [CrossRef]
12. Zhou, Q.; Yang, Y.; Hong, F.; Feng, Y.; Guo, Z. User identification and authentication using keystroke dynamics with acoustic signal. In Proceedings of the 2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Hefei, China, 16–18 December 2016. [CrossRef]
13. Weir, M.; Aggarwal, S.; de Medeiros, B.; Glodek, B. Password cracking using probabilistic context-free grammars. In Proceedings of the 2009 30th IEEE Symposium on Security and Privacy, Oakland, CA, USA, 17–20 May 2009. [CrossRef]
14. Guan, A.; Chen, C.-M. A Novel Verification Scheme to Resist Online Password Guessing Attacks. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 4285–4293. [CrossRef]
15. Randa, A.-W.; Adi, M. Authentication and Role-Based Authorization in Microservice Architecture: A Generic Performance-Centric Design. *J. Adv. Inf. Technol.* **2023**, *14*, 758–768. [CrossRef]
16. Simmons, G.J. Symmetric and Asymmetric Encryption. *ACM Comput. Surv.* **1979**, *11*, 305–330. [CrossRef]
17. Yassein, M.B.; Aljawarneh, S.; Qawasmeh, E.; Mardini, W.; Khamayseh, Y. Comprehensive study of symmetric key and asymmetric key encryption algorithms. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–7. [CrossRef]
18. Dixit, P.; Gupta, A.K.; Trivedi, M.C.; Yadav, V.K. Traditional and Hybrid Encryption Techniques: A Survey. In *Networking Communication and Data Knowledge Engineering*; Lecture Notes on Data Engineering and Communications Technologies; Perez, G., Mishra, K., Tiwari, S., Trivedi, M., Eds.; Springer: Singapore, 2018; Volume 4. [CrossRef]
19. Zhang, Q. An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption. In Proceedings of the 2021 2nd International Conference on Computing and Data Science (CDS), Stanford, CA, USA, 28–29 January 2021; pp. 616–622. [CrossRef]
20. Abdullah, A. Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. *Cryptogr. Netw. Secur.* **2017**, *16*, 11.
21. D'souza, F.J.; Panchal, D. Advanced encryption standard (AES) security enhancement using hybrid approach. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 647–652. [CrossRef]
22. Galla, L.K.; Koganti, V.S.; Nuthalapati, N. Implementation of RSA,. In Proceedings of the 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, India, 16–17 December 2016; pp. 81–87. [CrossRef]
23. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
24. Mikhail, M.; Abouelseoud, Y.; Elkobrosy, G. Extension and application of El-Gamal encryption scheme. In Proceedings of the 2014 World Congress on Computer Applications and Information Systems (WCCAIS), Hammamet, Tunisia, 17–19 January 2014; pp. 1–6. [CrossRef]
25. Olutola, A.; Olumuyiwa, M. Comparative analysis of encryption algorithms. *Eur. J. Technol.* **2023**, *7*, 1–9. [CrossRef]
26. Li, N. Research on Diffie-Hellman key exchange protocol. In Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 16–19 April 2010; pp. V4-634–V4-637. [CrossRef]
27. Bedoui, M.; Bouallegue, B.; Ahmed, A.M.; Hamdi, B.; Machhout, M.; Mahmoud; Khattab, M. A secure hardware implementation for elliptic curve digital signature algorithm. *Comput. Syst. Sci. Eng.* **2023**, *44*, 2177–2193. [CrossRef]

28. Kurosawa, K.; Desmedt, Y. A New Paradigm of Hybrid Encryption Scheme. In *Advances in Cryptology—CRYPTO 2004. CRYPTO 2004*; Lecture Notes in Computer Science; Franklin, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3152. [CrossRef]
29. Gupta, S.; Sharma, J. A hybrid encryption algorithm based on RSA and Diffie-Hellman. In Proceedings of the 2012 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2012; pp. 1–4. [CrossRef]
30. Jain, S.; Muntean, C.H.; Verma, R. Honey2Fish-A Hybrid Encryption Approach for Improved Password and Message Security. In Proceedings of the 2023 IEEE 9th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), New York, NY, USA, 6–8 May 2023; pp. 198–203. [CrossRef]
31. NIST Announces First Four Quantum-Resistant Cryptographic Algorithms. NIST. 2022. Available online: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms (accessed on 1 January 2024).
32. Atkin, A.O.; Bernstein, D.J. Prime sieves using binary quadratic forms. *Math. Comput.* **2003**, *73*, 1023–1030. [CrossRef]
33. Violettas, G.E.; Papadopoulos, K. Passwords to absolutely avoid. In Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Bangalore, India, 17–19 February 2014; pp. 60–68. [CrossRef]
34. Alkhwaja, I.; Albugami, M.; Alkhwaja, A.; Alghamdi, M.; Abahussain, H.; Alfawaz, F.; Min-Allah, N. Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming. *Appl. Sci.* **2023**, *13*, 5979. [CrossRef]
35. Widianto, S.R.; Maulana, M.S.; Pratama, E.B.; Firmansyah, Y.; Nurmalasari, N. Python gmail dictionary attack using Wordlist. *AIP Conf. Proc.* **2023**, *2714*, 030033. [CrossRef]
36. Chakrabarti, S.; Singhal, M. Password-Based Authentication: Preventing Dictionary Attacks. *Computer* **2007**, *40*, 68–74. [CrossRef]
37. Passfault-Password Strength Tester. 2023. Available online: https://www.malwarefox.com/passfault/ (accessed on 28 March 2023).
38. Dixit, P.; Silakari, S. Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review. *Comput. Sci. Rev.* **2020**, *39*, 100317. [CrossRef]
39. Hitaj, B.; Gasti, P.; Ateniese, G.; Perez-Cruz, F. PassGAN: A Deep Learning Approach for Password Guessing. In *Applied Cryptography and Network Security. ACNS 2019*; Lecture Notes in Computer Science; Deng, R., Gauthier-Umaña, V., Ochoa, M., Yung, M., Eds.; Springer: Cham, Switzerland, 2019; Volume 11464. [CrossRef]
40. Alani, M.M. Applications of machine learning in cryptography. In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, Kuala Lumpur, Malaysia, 19–21 January 2019. [CrossRef]
41. National Institute of Standards and Technology (NIST). Digital Identity Guidelines: Authentication and Lifecycle Management. Special Publication 800-63-3. 2017. Available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-3.pdf (accessed on 1 January 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.