

# Triple-identity Authentication: The Future of Secure Access

Suyun Borjigin  
Independent Researcher  
yunsu000@hotmail.com

**Abstract**—In a typical authentication process, the local system verifies the user’s identity using a stored hash value generated by a cross-system hash algorithm. This article shifts the research focus from traditional password encryption to the establishment of gatekeeping mechanisms for effective interactions between a system and the outside world. Here, we propose a triple-identity authentication system to achieve this goal. Specifically, this local system opens the inner structure of its hash algorithm to all user credentials, including the login name, login password, and authentication password. When a login credential is entered, the local system hashes it and then creates a unique identifier using intermediate hash elements randomly selected from the open algorithm. Importantly, this locally generated unique identifier (rather than the stored hash produced by the open algorithm) is utilized to verify the user’s combined identity, which is generated by combining the entered credential with the International Mobile Equipment Identity and the International Mobile Subscriber Identity. The verification process is implemented at each interaction point: the login name field, the login password field, and the server’s authentication point. Thus, within the context of this triple-identity authentication system, we establish a robust gatekeeping mechanism for system interactions, ultimately providing a level of security that is equivalent to multi-factor authentication.

**Keywords**—triple-identity authentication, open hash algorithm, combined identity, unique identifier.

## 1. INTRODUCTION

In the digital age, multi-factor authentication (MFA) [1], [2] is nearly ubiquitous across online world. As a third-party service, MFA can help confirm a user’s identity by providing an extra code. However, it is an external auxiliary system and the code is transmitted over the network and then entered manually by users. On the other hand, the existing password-based authentication systems themselves lack robust internal mechanisms to protect their interactions with the outside world, and thus have to rely on external MFA services.

Authentication security depends largely on two aspects: password encryption and background authentication. Password encryption has been a mature field in authentication. However, it has shown insufficient for providing absolute security. Authentication security would be greatly enhanced if a password-based authentication system could independently and accurately identify and verify users’ identities without relying on external assistance. Unfortunately, this area has not yet received the attention it deserves in mainstream research within the field of authentication.

This article shifts the research focus to the establishment of

gatekeeping mechanisms in the interactions between the system and the outside world. Specifically, two measures are taken to address the above security issues. First, we redesign the representation of user identities to enable the system to accurately identify users. Secondly, we develop a new approach to securely verify these redesigned user identities.

In existing MFA-based authentication systems, the login credentials (i.e., login name and login password) entered by a user trigger the system to transmit an additional code to the user’s smartphone over the network. When the received code is entered, the user can then be granted access to their account. In this process, the login credentials as the first factor initiate the code transmission. The smartphone associated with the International Mobile Equipment Identity (IMEI) is used to receive the code as the second factor. The user’s subscribed mobile service related to International Mobile Subscriber Identity (IMSI) makes it possible to transmit the code.

This scenario illustrates that login credentials, along with the IMEI and IMSI, are indispensable factors for a successful login. However, this successful login was made possible by the collaboration of two systems. The first is the conventional password-based system, which is responsible for the initial identification of user’s login credentials. The second is an MFA system, tasked with verifying the user’s device through the network.

Considering this, we propose a novel user identification strategy that integrates user credentials with IMEI and IMSI numbers into a unified identity, referred to as a combined identity, represented as “credential+IMEI+IMSI.” This approach combines these three identity elements into a cohesive architecture, thus constituting a multi-dimensional identity protocol for the client, called a man-machine-service identity architecture. Subsequently, this architecture will be identified and then verified within one system. The remarkable characteristic of this architecture is its resilience against counterfeiting. No technology can replicate this combined identity on unauthorized devices, as it integrates IMEI-related physical identity and IMSI-associated service identity.

The existing algorithms typically hash a user password into a fixed-length string of characters [1], [3]. This process is not random; given the same input through any devices, the hash function always yields the same output. This inherent non-random and deterministic nature of the hash algorithms could be exploited by hackers to perform reverse engineering.

Moreover, the traditional authentication systems utilize the cross-system algorithms to create hash values for user-entered passwords. These local systems are not granted the autonomy to independently utilize the algorithm’s internal structure. Thus, these shared properties of the hash algorithms present a significant challenge in the realm of user authentication and are highly to be exploited by attackers.

If local systems were granted a certain degree of autonomy to utilize the hash algorithms independently, a truly secure identifier could be generated locally. Accordingly, the user's combined identity can be verified using this unique identifier, rather than relying entirely on hash values generated by cross-system algorithms.

To achieve this, a single-character conversion technique [4], [5], [6] is utilized to establish a matrix-like hash algorithm for password-based authentication systems, as shown in Fig. 1. This algorithm first converts the user's login credentials into a matrix of randomized hash elements, from which a longer and more complex authentication password can then be generated in case the input is a login password. In this study, everything entered into the system through the login fields will be hashed by the algorithm, whether secret or not.

To address the aforementioned issues, the internal structure of the algorithm (i.e., the hash elements of the matrix) is open to all user credentials, including the login name (i.e., username and phone number), the login and authentication passwords [6], [7], as shown in Figs. 1 and 2.

As the open algorithm is managed solely by the system in the background, its intermediate hash elements are concealed in the system, inaccessible to users, independent of personal information, not transmissible in cyberspace. Such elements are ideal components that can be utilized by the local system to generate a unique identifier to verify the combined identity. The identifiers made with such features are not only highly secure, but also virtually useless to hackers as they contain invalid characters. In addition, the length of the randomly generated authentication password and identifiers is variable, making it extremely difficult to reverse engineer the original credentials.

During registration process, the system combines the login credentials entered through a smartphone with the IMEI and IMSI numbers in a predefined salting manner and stores the combination. Following the conversion of the entered login credentials, the system selects a set of hash elements from the open algorithms to generate the identifiers associated with the entered credentials.

In the login process, once a credential is entered, the system checks both the IMEI of the smartphone and the IMSI of the registered service to determine whether the credential matched the stored combined identity. Following the identification, the combined identity will be verified using the stored credential identifier. Once verified, the process can proceed to the next round of user identification and authentication until the user's combined identities are identified and verified at all system interaction points.

For example, when a username (UN) or a phone number (PN) is entered via the login name field, the system initiates an identification process to confirm the compatibility of the username or phone number with the IMEI and IMSI. Once identified, the combined UN or PN identity can then be verified by the system using the UN or PN identifier. Upon the verification, the user can navigate to the password page.

Similarly, the compatibility of the login password (LP) with the IMEI and IMSI is identified when it is entered through the login password field. Once identified, the login password can then be transformed into a matrix of hash elements, thereby generating an LP identifier to verify the combined LP identity.

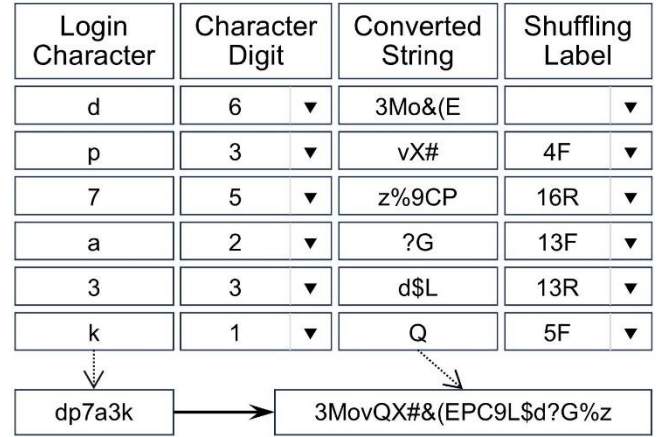
Following the verification, the authentication password (AP) will then be generated by the algorithm. Thus, an AP identifier can be created by selecting another set of hash elements from the same matrix, thereby verifying the combined AP identity.

All the implementations mentioned take place within a triple-identity authentication system in a login-authentication process. This approach serves as an effective gatekeeping mechanism, ensuring user authentication at three critical interaction points of the system: the login name field, the login password field, and the authentication point on the server.

## 2. MATRIX-LIKE HASH ALGORITHM

In this research, we employ the single-character conversion technique [4], [5], [6], which converts an individual character into a string of characters. The objective is to facilitate the conversion of each character within a string selected by a user, ultimately yielding a set of strings.

For instance, a user-selected character, "d" is randomly transformed into a six-character string "3Mo&(E" after the digit "6" is selected from the dropdown menu. This process is illustrated in the conversion unit of the second row, as shown in Fig. 1. Subsequently, other characters selected by the user, "p", "7", "a", "3", and "k", are individually transformed into a set of strings "vX#", "z%9CP", "?G", "d\$L", and "Q." This occurs after the corresponding digits "3", "5", "2", "3", and "1" are chosen from the dropdown menus, resulting in five additional strings produced by the relevant conversion units.



**Fig. 1.** The matrix-like hash algorithm and a pair of login and authentication passwords.

### 2.1. Generation of a Matrix-Like Hash Algorithm

By superimposing the above six units, a two-dimensional framework is created. Next, a column of instructions referred to as Shuffling Label is attached to the right of the framework, forming a matrix-like structure consisting of six rows and four columns, as shown in Fig. 1. The first column is designated as the Login Character. The second column is designated as the Character Digit, while the third is called Converted String. In this structure, each label indicates that its left-hand string is inserted into the preceding string, analogous to shuffling a deck of poker cards. For instance, the label "4F" directs the insertion of the second string "vX#" into the fourth insertion point of the first string "3Mo&(E" in a forward character order. This results in a temporary string as "3MovX#&(E."

The label “16R” is to insert the third string into the sixteenth insertion point of the first temporary string in reverse character order (i.e., “PC9%z”). However, there are only 10 insertion points in the first temporary string, which means that the third string can only be inserted into the 10th point, thus generating the second temporary string as “3MovX#&(Ez%9CP.” When all label instructions are executed, the original string “dp7a3k” is processed through the matrix-like structure, resulting in a longer, more complex string: “3MovQX#&(EPC9L\$d?G%z”, as shown at the bottom of Fig. 1.

It is clear that this matrix-like structure converts a string of login characters entered by a user into a longer, more complex string, thereby serving as a hash algorithm. Subsequently, we integrate this structure into a password-based authentication system, making it the hash algorithm of the system. Once this system is in place, users can easily enter their login characters, such as “dp7a3k”, into the login password field. This triggers the system to generate a matrix of six rows and four columns of hash elements, subsequently producing a longer and more complex string “3MovQX#&(EPC9L\$d?G%z.” In this way, all the operations, such as the selection of drop-down menus, conversion of characters, and implementation of labels, can be automatically executed in the background without users’ participation.

In the context of this system, the string of login characters used for logging in is defined as a login password, which users enter in the password field. The longer and more complex string is defined as an authentication password, functioning as a hash value. Together, these establish a pair of login and authentication passwords, as shown at the bottom of Fig. 1.

## 2.2. Unique Features of the Matrix-Like Hash Algorithm

This seemingly simple algorithm in Fig. 1 diverges greatly from usual hash algorithms in its primary functionalities. In addition to the function of hashing a user password into an adequately complex hash value, this algorithm also has several unusual features that can be exploited to their full potential.

1) Hash algorithms are typically designed to operate across different systems, hashing various user-input passwords into a uniform fixed-length output. Notably, the internal structure of these algorithms is not disclosed to the systems or users, and local systems lack the autonomy to leverage the internal workings of the algorithms. This situation highlights that the security of traditional password-based authentication processes fundamentally relies on the hashing algorithms themselves. While this reliance has undoubtedly led to continuous enhancements of these algorithms, the absence of a substantial number of local systems participating as implementers of the authentication process is a significant limitation. To address this issue, this study makes the internal structure of the cross-system matrix-like hash algorithm accessible to local systems, enabling them to enhance the security of the authentication process beyond the fundamental capabilities of the hash algorithm itself.

2) The local system has the capability to independently select the Character Digit column. By choosing various sets of digits from the drop-down menus, it generates variable-length authentication passwords (i.e., hash values). Furthermore, the local system autonomously selects the label set for the Shuffling Label column, allowing for the creation of

authentication passwords with diverse internal compositions. This approach to password construction significantly enhances their uniqueness, which can effectively address the inherent flaws associated with the non-random and deterministic nature of hash algorithms. Consequently, this method is particularly advantageous in mitigating the threats posed by reverse engineering.

3) More importantly, the local system can be granted the autonomy to randomly select a set of internal elements from the matrix, thereby generating a fully localized string within the system itself. This string functions as an identifier that is directly linked to the user’s login identity by the local system. Consequently, the system can utilize this identifier instead of traditional hash values to implement user authentication. This novel approach establishes a brand-new authentication paradigm. The advantages that this model brings to the authentication system are numerous, which will be discussed in detail in the subsequent chapters.

## 3. IDENTITY AND IDENTIFIER IN THE AUTHENTICATION SYSTEM

In the realm of digital identity authentication, the concepts of “identity” and “identifier” often overlap due to their contextual usage, making them frequently interchangeable in various contexts. However, the ways in which these terms are applied can vary significantly depending on the field of study, the theoretical framework, or the specific situation being examined. Therefore, it is essential to explicitly define and illustrate these two concepts to enhance our understanding of their roles.

User identification and authentication are fundamental components of the security of authentication. In this study, we propose that any data entered into the system through the login fields should be classified as “identity.” The accuracy and security of user identification are significantly influenced by the strategic combination of various types of identity factors, as single-factor authentication has been proven to be insecure. Instead, the user authentication process significantly depends on the uniqueness of these identifiers. Unfortunately, existing systems have yet to attain a satisfactory level of uniqueness in this regard. More importantly, the effectiveness of secure user authentication significantly hinges on the independent creation of identifiers by local systems. This is particularly crucial, as conventional hash values produced through cross-system hash algorithms have revealed certain vulnerabilities.

The concepts of “identity” [1], [8] and “identifier” [1], [9] are pivotal, possessing varying meanings depending on the context. Given the various interpretations of these concepts, it is essential to define them clearly and precisely. This clarity is crucial for paving the way for the generation of tamper-proof identities and unique identifiers. In the following sections, we will explore these concepts in details particularly in relation to their roles in user identification and authentication.

### 3.1. Identification Factors: The User’s Identity

**Identity in the Context of This Study:** In essence, an identity serves as evidence of an individual’s self-claimed persona, thereby establishing the user’s identification. Additionally, an identity can also act as proof of ownership

regarding a specific object, thus establishing that object's identification [10], [11]. If the object belongs to that user, it may then be considered a facet of the user's identity.

In reality, there are numerous practical examples of unique identification systems. For instance, each vehicle possesses a unique identity in the form of a license plate, which enables law enforcement agencies to trace the vehicle's owner. In this context, the authenticity of the owner is identified by the law enforcement using the officially registered license plate. Similarly, each subscribed mobile phone is assigned unique identities, namely the IMEI and IMSI numbers. These identities allow service providers to locate the phone's owner. In this scenario, the authenticity of the owner is identified by service providers based on the subscribed services.

Therefore, we argue that in this study, any identity that belongs to a user or is used to distinguish the user must be a verifiable entity. This verification must be done by another entity to confirm who the user claims to be or who an object belongs to. This indicates that an identity must be known by at least two entities, the owner and a verifier [1]. Otherwise, the owner cannot be verified as the uniqueness that only one entity apprehends is unprovable and hence meaningless. Thus, the argument above can be rephrased as: anything that is known by at least two entities may only serve as an identity, rather than an identifier. This serves as the fundamental basis for defining user identity in this study, and it will also facilitate the subsequent definitions and explanations of identifiers.

**Classification of Identity:** User identities can be conceptualized as a large group. In the realm of authentication, a username is traditionally classified as identities, representing a unique existence of an individual within a system. Conversely, a password functions as an identifier—an element that provides access to that identity. This differentiation highlights the essential roles that both elements play in securing user identities and facilitating authentication processes.

In addition to traditional credentials such as usernames and passwords, there exists a diverse array of electronic identities within the digital landscape. Examples of these identities include email addresses, social security numbers, digital certificates, and user IDs, among others.

However, as previously mentioned, any data entered into the system through the login fields is classified as identity. This distinction marks a significant difference between this study and traditional authentication scenarios. Our approach involves using the matrix-like hash algorithm established in Section 2 to hash any data entered in the login fields separately. This essentially confers upon the input data from the password field the same status as that of the username, subjecting both to the same hashing process. In other words, this study regards the login password as an essential part of user identity group as well.

Furthermore, IMEI and IMSI are sometimes categorized as identifiers despite being originally designed as identities for devices and subscribers [12], [13]. However, the IMEI and IMSI have previously been classified under the category of user identity group, the subsequent discussion will integrate these elements with user credentials, thus constructing a combined user identity ecosystem. However, this does not

imply that users will operate or manage the IMEI and IMSI. Instead, the system will seamlessly combine these identifiers with the user credentials in the background.

**Multi-Factor Identity:** In the context of text-based user identification, an “identity” refers to a distinctive element that sets an individual user apart from a broader population, enabling their recognition by various systems or entities. Given the considerable number of users with established identities, relying solely on a single type of identification factor for user identification poses significant risks. For instance, multiple users might inadvertently share the same password, complicating the verification process. Additionally, systems that rely on a single identification factor often require that this factor possesses a high degree of complexity to satisfy the security requirements. This necessity for increased complexity can lead to a notable decline in usability for clients, highlighting the inherent trade-offs faced in balancing usable security and secure usability.

The key to addressing this issue lies not merely in simply increasing the number of identities, but rather in diversifying the types of identities involved. Mainstream multi-factor authentication solutions illustrate a fundamental principle: integrating as many different kinds of identity factors as possible during a successful login process—such as user credentials, IMEI, and IMSI—is essential for ensuring the security of user identification. By emphasizing variety in identity factors, we can develop a single, comprehensive, and robust security architecture that safeguards users against potential threats.

More importantly, it is essential to securely identify this singular identity architecture within one system through a one-time process. Otherwise, we may still encounter issues related to the MFA-based dual-system user identification processes. For instance, the additional MFA code transmitted over the network may be intercepted, or manually entered MFA code may be compromised by malware. It is crucial to address these vulnerabilities to enhance the overall security and efficiency of user identification.

**Triple-Identity Authentication System:** It is important to note that the user identification relying on a single factor often presents a considerable risk. Let us review a prevailing login process. Upon entering the login credentials by a user, the system verifies the username and password as the first login factor. However, this does not verify whether the login attempt is from that user's registered device or not. Therefore, an MFA service is utilized to transmit a code as the second factor to the user's mobile device. Upon entering this code, it validates the device as “trusted” and access can then be granted to the user account. In this MFA-based process, the combination of the user's login credentials, IMEI (related to the user's device), and IMSI (pertaining to the mobile service) plays a crucial role in strengthening identification security.

Accordingly, a combined identity can be generated for user identification by combining the three identities of a credential, IMEI, and IMSI, and described as “credential+IMEI+IMSI.” Here, the credential may be a username, phone number, or login password, which means in this study the login password is endowed with the same status as a login name. In this way, all the essential components required for a successful login are encompassed within this trinity identity framework.

It is worth noting that this framework brings significant benefits to this study in mitigating the strength of login credentials. The relevant content will be discussed in detail in subsequent chapters.

In a typical login process, when a user submits their login credentials in the designated fields, the system generates a combined identity associated with those credentials. This identity is subsequently verified against the corresponding credential identifiers, which will be explored in the following section.

For instance, when a username is entered in the login name field, it can be collectively identified with the IMEI and IMSI, leading to the creation of a combined UN identity. The system then verifies this combined UN identity using the associated identifier. After this round of identification and authentication, the user can proceed to enter their login password page.

When the user types their password into the relevant field, a combined LP identity is generated, much like the username process. Upon successful verification of the combined LP identity with the relevant identifier, the login password can then be transformed into an authentication password.

On the server, this generated authentication password can be collectively identified with the IMEI and IMSI, allowing for the creation of a combined AP identity. Finally, the system verifies this combined AP identity with the relevant identifier to ensure a secure login process.

This successful login process establishes a triple-identity authentication system. By incorporating all essential elements necessary for a successful login—such as credentials, IMEI and IMSI—the system independently implements a level of identification and authentication equivalent to multi-factor authentication.

### 3.2. Authentication Factors: The System's Identifier

**Identifier in the Context of This Study:** Mainstream user authentication methods predominantly rely on verifying user identities through hash values produced by established cross-system hash algorithms. While these algorithms play a critical role in the authentication process, their inherent non-random and deterministic nature present vulnerabilities: they always yield the same hash value. Such weaknesses can be exploited by hackers who may reverse-engineer input passwords. Especially, the local systems generally lack autonomy in generating hash values (i.e., identifiers), primarily depending on those produced by the cross-system hash algorithm solely for authentication purposes.

To address the above issues, we grant the local system the autonomy to select a set of randomized hash elements from the matrix. This selection will be used to generate a string, which is subsequently defined as the identifier and linked to the user's combined identity corresponding to their credentials. By adopting this approach, the local system is empowered to verify the user's identity through a locally generated identifier, rather than relying on hash values produced by cross-system hashing algorithms.

In the field of identity authentication, the characteristics of uniqueness, consistency, simplicity, stability, and relevance represent fundamental requirements for identifiers [14]. Each of the attributes plays a crucial role in ensuring authentication security, however, the concept of uniqueness stands out as a

cornerstone of any identifier system. One of the primary purposes of an identifier is to distinguish each entity within a system. To ensure that identifications remain unique to each instance, it is essential that the identifier is fully managed by the local system in the background.

To enhance the essence of uniqueness in identifiers, it is useful to integrate other features such as non-transmissibility of identifiers, the elimination of the need to manually enter MFA codes, and the inaccessibility of identifiers for users. The feature of non-transmissibility reinforces the security framework by ensuring that identifiers cannot be easily shared or replicated, thus preserving their unique characteristics. Similarly, eliminating the requirement for manual input (e.g., MFA codes) can simplify user interactions, thereby fostering a seamless authentication process while maintaining security integrity. Furthermore, ensuring that certain identifiers are inaccessible to users minimizes the risk of leakage, ensuring that uniqueness is upheld within the system.

**Identifier Requirements:** In contrast to the concept of “identity,” an “identifier” functions primarily to facilitate the comparison of two entities, thereby ascertaining whether they correspond to one another. This comparison involves a collaborative exchange between the subject of authentication - commonly the possessor of the identifier - and the object of authentication, which is the entity requiring verification. Typically, the identifier's owner supervises this interaction, establishing the standards and criteria that the object must satisfy in order to be deemed legitimate. When precisely designing these identifiers, it becomes essential to integrate specific conditions that will guide the verification process effectively.

Securing successful authentication is highly dependent on the design and implementation of identifiers. In this context, we design the novel authentication identifiers to verify the combined identities described earlier without relying on MFA services. To achieve this goal, we stipulate that an effective authentication identifier should:

- be fully managed by the system in the background.
- be only known to the system.
- not contain any personal information.
- not be transmitted over cyberspace.
- be inaccessible to users.
- not be permitted to enter the system via login fields.

The conditions outlined above directly correspond to the principal weaknesses inherent in existing authentication systems. The identifiers that satisfy all these conditions are exceptionally effective in verifying the user's combined identities. In Section 6, we will introduce a specific method for generating such identifiers to ensure accurate verification of each combined identity.

## 4. GATEKEEPER MECHANISM UTILIZING THE IDENTITY-IDENTIFIER PROTOCOL

In Fig. 1, the matrix-like algorithm performs dual functions. First, it transforms a login password into an authentication password, analogous to traditional hash algorithms. Secondly, the local system is granted autonomy to use the intermediate hash elements of the algorithm to generate the identifiers that can be used to verify the user's identity, acting as hash values.

This second function is the key focus of this study.

To implement the second function, we take two measures. First, we convert all user login credentials, which include the login name and login password, into a matrix of hash elements using the matrix-like algorithm. Second, we make the internal structure of these matrices accessible to all user credentials, such as the login name as well as the login and authentication passwords. Therefore, this algorithm is characterized as an open hash algorithm.

By opening up the hashing algorithm, the system can effectively represent the readily accessible login name using a matrix of randomized hash elements. Furthermore, the local utilization of these algorithmic hash elements to create unique identifiers for authentication provides a noteworthy alternative to the conventional reliance on hash values produced by cross-system hashing algorithms.

When the login password is subjected to a hashing process, a matrix of hash elements is generated, accompanied by an authentication password. These hash elements of this matrix can subsequently be utilized to create identifiers by selecting varying sets of hash elements for both the login password and the authentication password, thus enabling the authentication of the corresponding combined identities.

Once each identifier is generated, it is matched with the corresponding combined identity. This process establishes a specific identity-identifier authentication pair for each user credential at the system's interaction points, such as the login name field, the login password field, and the authentication point on the server. Consequently, these three identity-identifier pairs collectively form an internal autonomous gatekeeping mechanism, significantly enhancing the security of user authentication.

Based on the analysis presented, it is evident that our research emphasis has transitioned from password encryption to the gatekeeping mechanism of user authentication. While encryption is designed to protect against external threats, traditional security measures in authentication have primarily relied upon external services. In contrast, our gatekeeping mechanism seeks to address the system vulnerabilities that arise during interactions with external environments, enabling secure authentication without the need for external support. By adopting this innovative identity-identifier approach, we establish a more independent and robust security framework, ultimately enhancing the overall security of authentication.

Identity impersonation and remote attacks pose significant threats to user authentication. The former primarily concerns how identity is presented on the client side, whereas the latter pertains to the internal authentication mechanism on the server side. While we cannot directly control remote attacks, there are measures we can take to bolster our systems' defenses. Empowering the local system to effectively safeguard user identification and authentication is crucial. To this end, implementing a gatekeeping mechanism characterized by the identity-identifier protocol emerges as a promising solution. In the subsequent sections, we will delve into the process of generating the combined identity along with the corresponding credential identifier.

## 5. CLIENT-SIDE PROTOCOL: REDESIGNING THE USER IDENTITY

### 5.1. Trinitarian Identity Architecture of the Mobile Login

The initial step in authentication is to ensure the security of user identification, preventing legitimate users from being impersonated. In today's Internet era, the vast majority of users leverage their mobile devices to access various web services, making mobile login the preferred method for user identification. Typically, these mobile logins are implemented through multi-factor authentication systems to guarantee the security of identification.

Technically, an MFA-based mobile login comprises three essential components: a user's login credentials, a mobile device, and a subscribed online service. The login credentials represent the user's privately known factor. The mobile device—specifically, a smartphone for the purpose of this study—is identified by its IMEI number, which links to the user. Furthermore, the online service is recognized by the IMSI number, signifying the user's service registration. When a user inputs their login credentials, this action serves as the first factor for identification. Subsequently, an MFA code is sent to their mobile device, forming the second factor that verifies the user's access to their subscribed services.

Among the three components discussed, the absence of any credential suggests that the login process may not yet have been initiated. A lack of an IMSI implies that the transmission of the code is not feasible. Furthermore, the absence of IMEI indicates that there is no available device to receive the MFA code. In summary, the collaborative integration of these components is essential for a successful mobile login.

In the process of a typical login attempt, the IMSI is usually identified first to determine if an MFA code can be dispatched. Once the user receives the code after service identification, the system integrates this code with the entered credentials and the IMEI using a predetermined salting technique. This approach establishes a multi-dimensional identity protocol for the client side, referred to as a "trinitarian identity architecture." This man-machine-service framework serves as the defining characteristic of the user and is conceptualized as a combined identity, termed a "credential+IMEI+IMSI."

While the concept of a trinitarian identity may seem intricate at first glance, it is designed to streamline the interface operation. Specifically, users simply enter their login credentials through their smartphones. After identifying the IMEI and IMSI numbers, the system merges these numbers with the credentials in a predetermined manner to generate the combined identity.

### 5.2. Generation of the Combined Identity

In this study, the term "login credentials" pertains to both the login name and password. The login name can consist of an email username, a phone number, or any user-customized text username.

When referring specifically to a username (UN), the combined UN identity is represented as "UN+IMEI+IMSI." Similarly, for a smartphone's phone number (PN), the combined PN identity is formatted as "PN+IMEI+IMSI." Once these combined identities are created, they are securely stored in the database for future authentication comparisons.

In traditional authentication scenarios, the primary function of a user password is to be converted into a hashed password. In addition to this function, this study proposes additional functions for the pair of login and authentication passwords by integrating them with the IMEI and IMSI numbers to generate a relevant combined identity.

Therefore, upon entering the login password (LP), a combined LP identity is generated in the formation of “LP+IMEI+IMSI,” which is subsequently verified using the associated LP identifier. Only after this round of identity-identifier authentication can the login password be converted into an authentication password (AP). After this conversion, a combined AP identity is generated as “AP+IMEI+IMSI,” which is then verified using the relevant AP identifier at the server’s authentication point. Alternatively, the authentication password may also serve as an identifier like a conventional hash value, facilitating the verification of the “IMEI+IMSI” structure.

The primary purpose of the above configurations is not to fundamentally alter the essential role of passwords. Rather, we seek to harness their intrinsic properties and functionalities to create additional layers of protection for user identification.

### 5.3. Benefits of the Combined Identity

Benefits of the combined identity are manifold. Firstly, by incorporating IMEI and IMSI, a user’s device can be linked directly to their service provider using their login credentials. This incorporation guarantees that only the authorized device registered with the service provider can access a user’s online account. In contrast, any login attempts made from devices other than the user’s authorized smartphone will fail during the IMEI and IMSI verification process, leading to a ban on access. Importantly, the absence of any one of these three elements (user, device, and service) can lead to identification failure. Therefore, this trinitarian identity model serves effectively to thwart the initial stages of remote attacks, specifically preventing impersonation from any unauthorized devices.

Secondly, this trinitarian architecture can ensure that even if any one of the identity factors are cracked, hackers are still unable to establish the legitimate user’s combined identity via their devices, unless they have physical control over the user’s smartphone.

Thirdly, this trinity also provides robust protection against a SIM swapping attack [15], [16]. In this type of malicious action, an attacker convinces a mobile phone carrier to switch the victim’s phone number to a new SIM card embedded in the attacker’s device. However, it is not possible for the combined identity of a legitimate user to be correctly identified on a device with different IMEI. In sum, this triple-identity authentication system implements a “trust no entry” philosophy to identify and verify every input into the system. This zero trust [17] principle treats every access attempt as a potential threat. Within this trinitarian identity architecture, user privileges are minimized to only allow access at the level of the IMEI-associated device and IMSI-registered service. Therefore, the identities of a user, device, and service are simultaneously verified at each interaction point, significantly boosting the security during user identification.

Fourthly, merging a login password into the “IMEI+IMSI”

structure creates a fairly complex combination, which makes the login password unnecessary as complex as before. Thus, we can individually set the login and authentication passwords to meet the requirements of password strength (i.e., length and character type) [2], [3], [18].

A login password can be specified in a range from five to fifteen characters in length and contain only lowercase letters and digits, which can be defined as valid characters in this study, while others are regarded as invalid. In contrast, an authentication password is required to be at least twenty characters long and must contain four-character classes, such as uppercase letters, lowercase letters, digits, and symbols [3], [18], [19]. Furthermore, a login name, which may encompass a username or phone number, is generally composed of alphanumeric characters. During the hashing process, all uppercase letters, if present, are converted into lowercase to meet the criteria established for the login password. These settings collectively offer several advantages.

1) When integrating a login password into a combined identity, the complexity associated with that password is effectively supplanted by the characteristics of the combined identity. Thereby, utilizing a login password composed solely of lowercase letters and digits not only provides users with an optimal level of secure usability but also represents the highest level of user-friendliness that a text-based authentication system can offer. This approach adeptly meets users’ needs and preferences while simultaneously upholding the overall security of the system. In contrast, authentication passwords may include any characters that a computer can process. This flexibility guarantees that the system’s stringent requirements for usable security are satisfied, thereby ensuring that the user’s login password configurations remain uncompromised. Within the framework of the triple-identity authentication system, this approach adeptly resolves the long-standing inherent trade-off conflict between usability and security [20], [21], enabling password usability and security to coexist in a harmonious manner.

2) The triple-identity authentication system implements an MFA-based dual-system user identification process within a single framework. This innovation reduces the need for multiple logins, thereby streamlining the traditional authentication process. This enhances the overall user experience (UX) [22] by making it more intuitive and user-friendly. In addition, the advent of combined identity offers a straightforward perspective for research in the realm of UX. The lowercase letters and digits are the easiest characters for users to input on any type of keyboard. This is particularly important for modern people working and living depending on the compact screens of their mobile devices. Simplicity is a fundamental principle in user-centered design. By simplifying login credentials, the process becomes more efficient, resulting in a vastly improved user experience.

3) In this study, the proposed format for the combined identity is represented as “credential+IMEI+IMSI.” Here, the term “credential” primarily refers to the user’s password, which is frequently reused [19] across various platforms and services. However, it is essential to note that both the IMEI and IMSI numbers are unique on a global scale. As long as the system can recognize this combination, the user’s identity will never be subject to impersonation. Therefore, integrating these



elements with the login credential can ensure that the combined identity remains distinct and never duplicates. This nuanced approach highlights the relationship between user-specific credentials and universally unique IMEI and IMSI, which contributes to a more secure and individualized digital identity framework.

4) Thanks to the implementation of strength settings, it is now possible to restrict the characters that are entered into the system through the login fields. This study has specifically classified lowercase letters and digits as valid characters and all others as invalid. As a result, the login fields can be configured to accept exclusively lowercase letters and digits, rejecting authentication passwords and identifiers because they contain invalid characters. This approach not only improves the user experience but also significantly bolsters overall system security and usability.

## 6. SERVER-SIDE PROTOCOL: CREATING THE IDENTIFIER

Presently, the prevalent authentication protocol for verifying a user's identity depends on a secret handshake. Typically, the handshake is achieved by using a stored hash value to verify the hashed password. In this process, the input password is associated with its fixed-length hash value in a non-random way: given the same input, the algorithm always yields the same hash value. This inherent non-random and deterministic nature of traditional hash algorithms can be exploited by attackers to reverse-engineer the input password from the stolen hash value.

In this study, instead of using the stored hash generated by a cross-system algorithm to verify the hashed password, the local system, i.e. the triple-identity authentication system, is endowed with the autonomy to generate an identifier using the intermediate hash elements of the open algorithm to verify the user's combined identity. Therefore, our handshake scenario is between the combined identity and the identifier generated by the system rather than by the cross-system hash algorithm.

### 6.1. CIA Triad: Guidelines for Identifier Design

Randomness, along with unpredictable variables, represent crucial components in any system's or individual's approach to information security. They hold a central role in the CIA triad: Confidentiality, Integrity, and Availability [23], [24], and therefore have a significant part to play in designing identifiers for server-side authentication. In contrast to existing authentication systems, our approach involves designing variable-length identifiers using randomized hash elements extracted from the open algorithm. This is intended to ensure that the information is only available to authorized users (confidentiality), maintains its accuracy and consistency (integrity), and must be readily accessible when needed (availability). Subsequently, the combined identities can be authenticated using a variable-length random identifiers rather than the stored hash created by a cross-system algorithm.

Fig. 1 illustrates two rounds of randomization of the login password. The first round involves selecting digits from the Character Digit column to randomly generate a set of converted strings. In the second round, labels in the Shuffling Label column are randomly selected to create a matrix of hash elements, which subsequently generates an authentication

password. This process ensures that the hash elements are concealed within the system, remaining inaccessible to users and independent of any personal information.

Moreover, when the algorithm's internal structure is made available to the local system for utilizing the hash elements, the login password undergoes a third randomization. This step effectively produces unique identifiers. Given that both the authentication password and the identifiers can vary in length, randomization can be highlighted as a prominent feature of this study.

Furthermore, considering the properties mentioned above, the identifiers of the triple-identity authentication system cannot be transmitted over cyberspace and thus do not need to be manually entered into the login fields. Each of the above characteristics is particularly significant in the context of confidentiality, integrity, and availability.

### 6.2. Credential Conversion and Identifier Definition

In the process of logging in, users typically provide a public login name. This practice is a standard step in conventional authentication protocols. However, it also presents a potential vulnerability that malicious actors may exploit. To address this significant concern, we propose a novel approach that employs the open algorithm to randomly hash the public login name into a matrix of hash elements. This method grants the local system the freedom to utilize the intermediate elements of the open algorithm to generate unique identifiers for the login name.

Take the virtual email address "Benz428@woxinet.com" as an example. During the registration, once the username (UN) "Benz428" is entered, the system converts it into a matrix of hash elements, as shown in Fig. 2. Following this conversion, the local system can then randomly select a set of elements from the matrix to generate a string, which is subsequently defined as the UN identifier. This identifier is then associated with the combined UN identity described in Section 5. In practice, while users may prefer to enter their complete email addresses, the system only hashes the username. Notably, any uppercase letters are converted to lowercase, and any non-alphanumeric characters are removed.

Username	Character Digit	Converted String	Shuffling Label
B	3 ▼	y]Q	▼
e	5 ▼	#ws%8	5F ▼
n	3 ▼	O^&	9R ▼
z	2 ▼	\$d	17R ▼
4	3 ▼	]Lh	13F ▼
2	3 ▼	zF=	8F ▼
8	1 ▼	m	11F ▼

Fig. 2. The conversion of username.

There are multiple ways to select the hash elements. In Fig. 2, an element row including the login character, an element column excluding the Login Character column, or a set of elements randomly selected by the system can be used to form



the string. For example, the system selects a set of elements “4”, “O^&”, “17R”, “2”, and “zF=”, combines them together into a string “4O^&17R2zF=”, defines it as a UN identifier, and then associates it with the corresponding combined UN identity “UN+IMEI+IMSI.” The length of the UN identifier is not fixed, depending on the random selection of the hash elements.

Similarly, a phone number (PN) can also be hashed into a matrix of hash elements by the algorithm. Once hashed, the system randomly selects a set of elements from the algorithm, combines them into the PN identifier, and then associates it with the combined PN identity “PN+IMEI+IMSI.”

Furthermore, after converting the login password (LP), the local system selects a set of hash elements from the matrix in Fig. 1 to create the LP identifier, thus verifying the combined LP identity “LP+IMEI+IMSI,” just as it did for the username. Following the generation of the authentication password (AP), the system generates the AP identifier using another set of elements selected from the same matrix, and then verifies the combined AP identity “AP+IMEI+IMSI.” Alternatively, the created authentication password may also serve as a traditional hash value to verify the identity structure like “IMEI+IMSI.”

### 6.3. Benefits of the Unique Identifiers

The standout feature of the handshake mechanism is that the local system is able to utilize the algorithm’s internal structure to generate a unique identifier, which in turn verifies the combined identity. This innovative approach, which does not rely solely on hash values generated by cross-system algorithms for identifier creation, significantly enhances the randomness and security of user authentication. As a result, this mechanism effectively addresses the inherent non-random and deterministic issues, thus providing several advantages for user authentication:

First, using variable-length random identifiers means that you may get different, or even an infinite number of identifiers from the same input. This makes the reverse-engineering process more computationally expensive and time-consuming without imposing any burden on the authentication system.

Secondly, by hashing all user login credentials, the use of autonomously generated identifiers can accurately verify combined identities at each interaction point (see Section 3). This approach equips each interaction point with a robust gatekeeping mechanism, thereby offering comprehensive and multi-layered protection for the authentication system.

Thirdly, system-managed identifiers do not need to be transmitted in cyberspace, nor do they need to be manually entered into the system. This effectively eliminates the risk of interception through cyberspace and reduces the likelihood of being compromised by malware that exploits manual input. By implementing this measure, the overall security of the system is streamlined, which further minimizes its attack surface and limits opportunities for unauthorized access to these critical identifiers.

Lastly, due to the uniqueness of combined identities and identifiers, the system must be able to verify the legitimate user without the assistance of MFA services. This capability significantly increases the operational efficiency of user authentication, while providing a superior level of security.

## 7. VERIFICATION AT THE INTERACTION POINTS

Typically, the login process starts with entering a username (UN) into the login name field, where the system ascertains whether the IMEI and IMSI numbers of the user’s smartphone match the input. Upon the successful identification of the username, IMEI, and IMSI, a combined UN identity is created and represented as “UN+IMEI+IMSI.” In the event that the identification is unsuccessful, a combined UN identity will not be generated, and access will be denied. Subsequently, that identity is verified using the stored UN identifier. Upon successful verification, the user can access the password page. In the event that a phone number is entered, the system applies a similar identification and authentication process.

In the context of the login password field, the verification of the linkage between the LP identifier and the combined LP identity, which consists of “LP+IMEI+IMSI,” should be strictly confined to the registered smartphone. Only under these conditions can an authentication password be effectively generated through the open algorithm.

At the server’s authentication point, there is no need for password identification as the authentication password (AP) is generated directly by the algorithm. This means that the combined AP identity, represented as “AP+IMEI+IMSI,” can be authenticated using the AP identifier. As a result, users are seamlessly granted access to their accounts.

## REFERENCES

- [1] Grassi, P. A., Garcia, M. E., and Fenton, J. L. “Digital identity guidelines.” NIST SP 800-63-3, National Institute of Standards and Technology, Mar. 2020. <https://doi.org/10.6028/NIST.SP.800-63-3>
- [2] NIST. “Multi-factor authentication.” NIST Small business cybersecurity corner, Feb. 2022. Available at: <https://nist.gov/itl/smallbusinesscyber/guidance-topic/multi-factor-authentication>
- [3] Greene, K. K., Kelsey, J., and Franklin, J. M. “Measuring the usability and security of permuted passwords on mobile platforms,” NISTIR 8040. National Institute of Standards and Technology, 2016. <http://dx.doi.org/10.6028/NIST.IR.8040>
- [4] Su, Y. and Xi, M. (2019, April 4). Password generation method which satisfies the requirement for security and usability simultaneously (PCT/IB2019/052719).
- [5] Su, Y. and Xi, M. (2023, Nov. 23). Method for a login-authentication system using a pair of login and authentication passwords (PCT/IB2023/061846).
- [6] Borjigin, S. (2024). “Systematic solutions to login and authentication security problems: A dual-password login-authentication mechanism.” arXiv:2404.01803.
- [7] Borjigin, S. (2024). “An alternative to multi-factor authentication with a triple-identity authentication scheme.” arXiv:2407.19459.
- [8] Cameron, K. (2006, May). “The laws of identity.” Available at: <http://www.identityblog.com/?p=354>
- [9] International Standard Organization, ISO/IEC 24760 -1 *Definition of digital identifier*, 2011.
- [10] Doe, J. “The Dichotomy of Self in Ownership and Identity.” Journal of Identity Studies, vol. 10, no, 1, pp. 15-30, 2022.

- [11] Johnson, L. M. "Digital Ownership and the Evolution of Identity in the 21st Century." *International Journal of Digital Culture*, 12(3), 45-60, 2023.
- [12] Strahilevitz, L. J. and Lifshitz, S. "Digital Contracts: A New Era of Consumer Protection." *Harvard Law Review*, vol. 126, no. 4, April 2013.
- [13] Stevan J. Davis, & Vidas, T. "The Role of Mobile Device Identifiers in User Identification and Tracking." *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 123-134, 2013.
- [14] Wand, Y., & Wang, R. Y. "Anchoring Data Quality Dimensions in Conceptual Data Modeling." *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 940-950, 1996.
- [15] Lee, K., Kaiser, B., Mayer, J. and Narayanan, A. "An empirical study of wireless carrier authentication for SIM swaps." *USENIX Symposium on Usable Privacy and Security (SOUPS' 20)*, pp. 61-79, 2020.
- [16] Johnson, K. (2023, Dec.). "How to solve 2 MFA challenges: SIM swapping and MFA fatigue. Available at: <https://www.techtarget.com/searchsecurity/feature/How-to-solve-MFA-challenges-SIM-swapping-and-MFA-fatigue>
- [17] Rose, S., Borchert, O., Mitchell, S., and Connelly, S. "Zero trust architecture." NIST Special Publication (Vol. 800-207), Aug. 2020. <https://doi.org/10.6028/NIST.SP.800-207>
- [18] Habib, H. *et al.*, "Password creation in the presence of blacklists." In *Proc. of the Workshop on Usable Security (USEC)*, San Diego, CA, USA, Feb. 2017. <http://dx.doi.org/10.14722/usec.2017.23043>
- [19] Das, A., Bonneau, J., Caesar, M., Borisov, N., and Wang, X. F. "The tangled web of password reuse." In *Proc. Network and Distributed System Security Symposium*, San Diego, CA, USA, 2014, pp. 23-26, <http://dx.doi.org/10.14722/ndss.2014.23357>
- [20] Altawy, R. and Youssef, A. M. "Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices." *IEEE Access*, vol. 4, pp. 959-979, 2016. <https://doi.org/10.1109/ACCESS.2016.2521727>
- [21] Sasse, M. A., Smith, M., Herley, C., Lipford, H., and Vaniea, K. "Debunking security-usability tradeoff myths." *IEEE Secur. & Priv*, vol. 14, no. 5, pp. 33-39, Sep./Oct. 2016. <https://doi.org/10.1109/MSP.2016.110>
- [22] Smith, J. A. "Understanding user experience: A holistic approach to design." *Journal of User Experience Studies*, vol. 12, no. 4, pp. 45-67, 2023. <https://doi.org/10.1234/jues.2023.0045>
- [23] GeeksforGeeks, (2023, Mar.). "The CIA triad in cryptography." Available at: <https://www.geeksforgeeks.org/the-cia-triad-in-cryptography/>
- [24] Nieves, M., Dempsey, K., and Pillitteri, V. Y. "An introduction to information security." NIST Special Publication 800-12 Revision 1, June 2017. <https://doi.org/10.6028/NIST.SP.800-12r1>