


Article

Assessment of Two Privacy Preserving Authentication Methods Using Secure Multiparty Computation Based on Secret Sharing

Diana-Elena Fălămaş *, Kinga Marton *  and Alin Suciu

Computer Science Department, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania; alin.suciu@cs.utcluj.ro

* Correspondence: falamasdiana@mail.student.utcluj.ro (D.-E.F.); kinga.marton@cs.utcluj.ro (K.M.)

Abstract: Secure authentication is an essential mechanism required by the vast majority of computer systems and various applications in order to establish user identity. Credentials such as passwords and biometric data should be protected against theft, as user impersonation can have serious consequences. Some practices widely used in order to make authentication more secure include storing password hashes in databases and processing biometric data under encryption. In this paper, we propose a system for both password-based and iris-based authentication that uses secure multiparty computation (SMPC) protocols and Shamir secret sharing. The system allows secure information storage in distributed databases and sensitive data is never revealed in plaintext during the authentication process. The communication between different components of the system is secured using both symmetric and asymmetric cryptographic primitives. The efficiency of the used protocols is evaluated along with two SMPC specific metrics: The number of communication rounds and the communication cost. According to our results, SMPC based on secret sharing can be successfully integrated in real-word authentication systems and the communication cost has an important impact on the performance of the SMPC protocols.



Citation: Fălămaş, D.-E.; Marton, K.; Suciu, A. Assessment of Two Privacy Preserving Authentication Methods Using Secure Multiparty Computation Based on Secret Sharing. *Symmetry* **2021**, *13*, 894. <https://doi.org/10.3390/sym13050894>

Received: 13 April 2021

Accepted: 12 May 2021

Published: 18 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

Keywords: password-based authentication; iris-based authentication; secure multiparty computation; secret sharing

1. Introduction

Nowadays, while the most widely used authentication method is represented by password validation, biometric authentication is becoming more and more popular due to the many advantages it provides. Biometric traits are strongly bonded to the person they belong to and they uniquely identify the owner. These traits are part of the human body and, unlike passwords, cannot be forgotten or lost, excepting the case somebody suffers serious injuries. Some common biometric traits are: iris, fingerprint, retina and face. While it is recommended for users to have different passwords for different accounts, each person has only one set of biometric traits. This raises security concerns because, when compromised, biometric information can hardly be revoked and never replaced. Disclosed credentials can have negative consequences and stolen biometric traits can even facilitate identity theft.

Designing secure authentication protocols and systems represents a challenge of great interest in the academic research, especially when it comes to biometric credentials. According to the architecture, the authentication components of these systems can be centralized or distributed. In a centralized architecture, a single authentication component matches the credentials received from clients against the legitimate credentials which are persistently stored in databases. A security requirement when it comes to password-based authentication is never to store passwords in plaintext but apply a hash algorithm such as SHA. The same rule applies for biometric data, usually stored after being encrypted with classical algorithms. Centralised storage of biometric data raises both privacy concerns and even

legal issues. In a distributed architecture, on the other hand, multiple components interact with each other in order to perform an authentication operation. The clients sensitive information is securely stored in multiple distributed databases using secret sharing and components perform secure multiparty computation (SMPC) to verify credentials.

SMPC has been studied extensively, mainly focusing on its theoretical contributions in providing security in multiparty interactive protocols, until the emergence of new application domains such as cloud computing, IoT, etc. which provide the possibility of cooperative computing and data processing, but at the same time stress the demand for preserving the privacy of confidential data. In this context, SMPC has known a tremendous re-activation as a research domain, now the focus being mainly on the application-oriented aspects, providing the much-needed mechanism of jointly computing on private data, without leaking confidential information. The state of the art regarding SMPC in practice is presented in [1]. The SMPC authentication systems have the great advantage of ensuring data privacy during the entire authentication process, which represent a motivation for our research in SMPC authentication protocols based on secret sharing.

Secret sharing consists in dividing a sensitive value into multiple shares, such that individual shares do not reveal any information about the initial value but, when recombined, the secret can be reconstructed. If a (k, n) threshold schema is used, the sensitive value is splitted into n shares and the secret can be reconstructed using a minimum number of k shares. There are several threshold schemes for secret sharing, such as those proposed by Shamir [2], Blakley [3] and Asmuth-Bloom [4]. In the present authentication system, passwords and iriscode are divided into shares using Shamir threshold schema and each share is stored in a distinct database such that, if $k - 1$ databases are compromised, the attacker gains no useful information.

In order to guarantee the security of sensitive user information during the entire authentication process, secret values should never be reconstructed by the system. This can be achieved through SMPC protocols, which allow secure processing on secret shared data. The multiple distributed computing parties, each having access to only one database, collaborate in order to perform SMPC authentication operations. The input party or the client divides the password/iriscode into shares that are sent to the computing nodes and matched against the stored user's information. The authentication result is also represented as a secret shared value and the corresponding shares are sent to the result party, which represents the system or the application the client is trying to authenticate to. Only then the public result is revealed.

For password-based authentication, the equality between vectors of secret shared values is computed using SMPC. The iris-based authentication method can be divided into two phases: (1) the extraction from an iris image of the feature vector as a sequence of bits and (2) the matching of the iriscode with a template stored in a database. In this paper we focus strictly on the validation/matching of biometric credentials, when the Hamming distance (HD) [5] is computed using SMPC.

The main contributions of our research are: the design of a SMPC system used for both password-based and iris-based authentication and the evaluation of the protocols considering efficiency and SMPC specific metrics. Furthermore, starting from the SMPC adapted Hamming Distance for iris-based authentication, we added three enhancements to the base algorithm:

- In order to increase security, an extra check is made for the minimum number of iriscode bits that are considered when computing the Hamming Distance;
- The fusion of Hamming Distance and fragile bit distance (FBD) [6] that improves recognition accuracy, is integrated in the SMPC system;
- The database storage requirements for secret shared authentication data is decreased using bit decomposition.

As expected, each enhancement comes with a smaller or larger performance penalty.

The paper is structured as follows. In Section 2, the related work and the state of the art in SMPC authentication are presented. Section 3 describes the architecture of

the distributed system and Section 4 introduces the base SMPC authentication protocols. The three mentioned enhancements are presented in Section 5, together with a theoretical evaluation. The methods and results for the experimental results are described in Section 6, followed by conclusions in Section 7.

2. Related Work

Nowadays, hardware architectures provide highly efficient computations, which enable the integration of SMPC in real-world applications. During the last few years, many SMPC frameworks that allow encrypted data processing were developed. The authors of [1] describe the security requirements that must be provided by these systems: privacy, correctness, independence of input, guarantee of output and fairness. They also present an overview regarding the state of the art in designing SMPC protocols that resist in various security models. In the semi-honest adversary model (the honest-but-curious model), the corrupted parties do not deviate from the protocol but try to gain as much information as possible from the other participants. Although it provides weaker resistance than the other models, the semi-honest model manages to cover several security needs in many practical scenarios. In the malicious adversary model, the corrupted parties may not execute the protocol correctly and may manipulate messages. The resistant protocols in this model provide strong security but their complexity results in performance penalties. The covert adversary model represents a tradeoff: the corrupted parties may deviate from the protocol but they are caught cheating with a given probability. The IPS compiler that converts honest majority agreements into agreements under the malicious model is one of the state of the art technologies mentioned in [1].

SMPC techniques are used in various contexts such as password or biometric authentication, electronic voting, computation of privacy-preserving statistics using financial or medical data. SMPC can even be used in order to securely evaluate the S-boxes of the well-known block ciphers, Triple DES and AES [7], or for threshold Elliptic Curve Digital Signature Algorithm signing [8]. The SMPC protocols became rather mature and, during the last few years, they were integrated in highly topical fields, for example cloud computing [9], data mining [10] and machine learning [11]. These techniques are based on cryptographic tools that ensure data privacy, such as secret sharing, garbled circuits, oblivious transfer and homomorphic encryption.

The authors of [12] propose an E-voting scheme that uses SMPC based on Shamir secret Sharing. The scheme is secure and keeps the anonymity of the voters while it also provides efficiency and reliability. The E-voting scheme from [13], also based on SMPC, provides enhanced security as Visual Cryptography is used for the biometric identification of the voters. Two shares are created from a fingerprint image: one share is stored by an administrator and one share is stored by the voter, such that none of them has full access to the biometric data.

When it comes to secret shared passwords, matching can be performed using two methods:

- The legitimate password is reconstructed from persistent-stored shares before it is compared to the password received from a client;
- The legitimate secret shared password is never reconstructed and matching against another secret shared password received from a client is performed through SMPC.

A password-based authentication system that uses the first method and Shamir secret sharing is presented in [14]. The password is divided into shares before being stored, but the secret is reconstructed when a client authentication is performed. The system has an architecture with several component roles: client, dealer (performs secret sharing and secret reconstruction), shareholders, service and external server. The authors of [14] also present a possible security enhancement for their system, based on the idea of hiding the abscissa vector (the vector of integer points in which the Lagrange interpolation polynomial used by Shamir schema is evaluated when computing the secret shares). If a (k, n) threshold schema is used and an attacker steals k shares, he cannot reconstruct the original value without knowing the abscissa vector.

While in [14] the passwords are reconstructed during authentication, in our system the SMPC method is implemented and passwords are not revealed in plaintext during the authentication process. Our system is secure in the semi-honest model with at most k corrupted parties out of n parties, according to the (k, n) threshold schema (the (2,3) schema or the (3,5) schema). The authors of [15] describe a distributed password-based authentication service that uses three-party computation based on garbled circuits. Their protocol provides security against a single malicious party.

Several SMPC authentication systems with various characteristics were also developed for biometric authentication. In [16], the authors present possible threats for these systems together with the main cryptographic tools used nowadays to prevent the leakage of biometric data: SMPC, Verifiable Computation and Bloom Filters.

The SMPC iris-based and fingerprint-based recognition methods in [17] rely on two-party protocols based on homomorphic encryption, garbled circuits and oblivious transfer. The presented methods achieve security against semi-honest adversaries. In [18], the authors present an authentication system resistant in the malicious security model. The SMPC system, called SEMBA, is used for multimodal recognition that relies on both facial and iris biometrics. SEMBA relies on the SPDZ protocol as the inner cryptographic tool and uses two computing parties. Our SMPC system relies on Shamir secret sharing and it uses unimodal biometric protocols (only iris-based authentication is implemented). As a security enhancement, symmetric and asymmetric cryptography is also used in order to secure the communication between components in our system. A comparison between these systems is provided in Table 1.

Table 1. Comparison between three biometric authentication Secure Multiparty Computation (SMPC) systems.

SMPC Protocols for Biometric Authentication—Characteristics			
Characteristic	The SMPC protocols from [17]	The SMPC protocols from [18]	The SMPC protocols from our system
Biometric traits	iris, fingerprint	face, iris	iris
Cryptographic primitives	homomorphic encryption, garbled circuits, oblivious transfer	SPDZ	Shamir secret sharing
Computing nodes number	2	2	3 or 5
Threat model	semi-honest	malicious	semi-honest

Another short description of the way SMPC can be used for password authentication and biometric identification, along with the presentation of a successfully implemented system, is presented in [19]. The system, developed by Unbound Technology for enterprise environments, makes use of virtual Hardware Security Modules (vHSM) for authentication. These modules are software implemented.

The motivation of our work is to provide an evaluation for the used SMPC authentication protocols in order to highlight that SMPC based on Shamir threshold schema can be successfully integrated in authentication frameworks.

Preliminaries:

This paper is a continuation of our previous work [20], where two sets of SMPC protocols based on secret sharing are evaluated and compared. Both of the sets contain three main algorithms: SMPC Equality (the equality of two secret shared values), SMPC Comparison (the comparison of two secret shared values) and SMPC Interval test (the belonging of a secret shared value to a public interval). While in our previous work we focused on finding efficient SMPC inner algorithms in order to ensure a better performance for the three main algorithms, in this paper we integrate these algorithms in a SMPC authentication system. The set of algorithms with the best efficiency from the two sets

benchmarked in our previous work are now adapted and used in several authentication protocols for passwords and iris codes matching.

The SMPC protocols in our system use the (k, n) Shamir secret sharing threshold schema as the inner cryptographic primitive. A confidential value is splitted into n shares and at least k shares are needed in order to reconstruct the original value. Shamir secret sharing is based on the Lagrange interpolation polynomial. A polynomial $f(x)$ with degree $k - 1$ is constructed (Equation (1)), where the first coefficient is the secret value s and the other coefficients are random positive integer values.

$$f(x) = s + r_1x + r_2x + \dots + r_{k-1}x^{k-1} \tag{1}$$

The n secret shares are computed by evaluating $f(x)$ in n different positive and nonzero integer points. The secret value is equal to $f(0)$ and it can be reconstructed by interpolation if we know k secret shares s_i and the points a_i the shares were evaluated in (Equation (2)).

$$s = f(0) = \sum_{i=1}^k s_i \cdot \prod_{j=1, j \neq i}^k \frac{a_j}{a_j - a_i} \tag{2}$$

Shamir secret sharing is explained in detail in [2] and the way the SMPC main protocols based on secret sharing work is presented in [21].

3. System Architecture

The proposed SMPC system authenticates users based on plaintext usernames and secret shared passwords/iris codes. Two secret sharing threshold schemas are supported: the (2,3) schema (the confidential values are splitted into 3 shares and can be reconstructed using 2 shares) and the (3,5) schema (the confidential values are splitted into 5 shares and can be reconstructed using 3 shares).

Figure 1 presents the architecture of the proposed SMPC authentication system when the (2,3) Shamir secret sharing threshold schema is used. The authentication result is computed by matching secret shared passwords/iris codes stored in distributed databases against secret shared passwords/iris codes received from the client. These confidential values are distributed across the computing nodes and they are never reconstructed. One share is not needed when reconstructing the authentication result (it is represented by a dotted line).

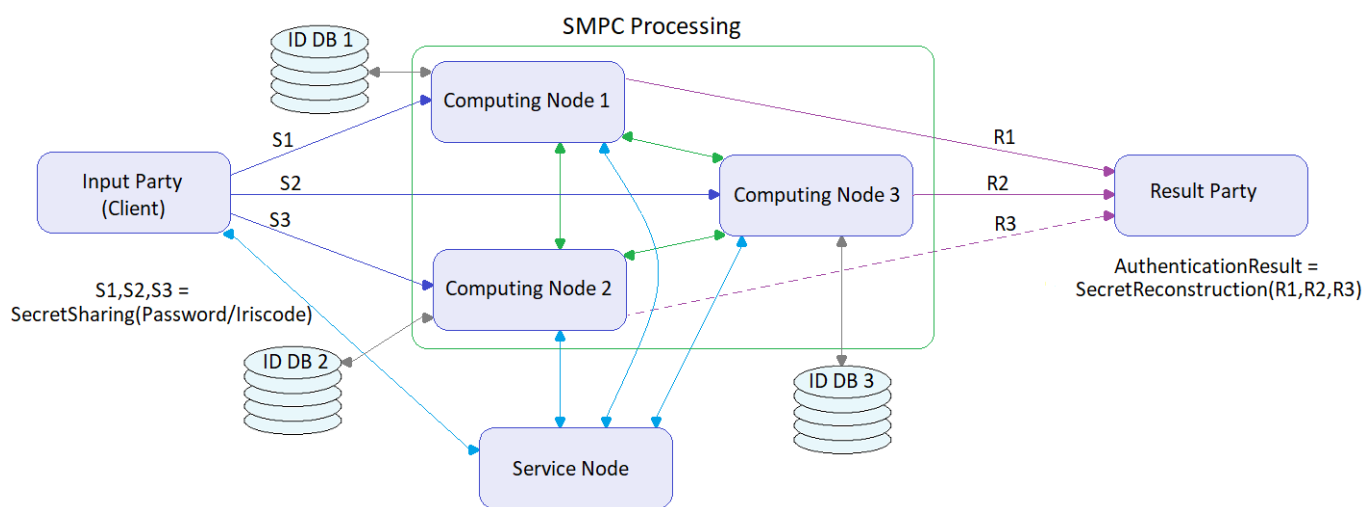


Figure 1. The architecture of the SMPC authentication system based on Shamir secret sharing.

The distributed architecture of the system consists of multiple components having one of the following roles:

- Input party: divides the password/iriscodes into shares and sends the shares to the computing nodes, together with the plaintext username;
- Computing nodes: exchange messages with one another in order to perform a SMPC authentication operation by matching the client input with the information stored in databases. They also have the role to store new usernames and secret shared passwords/iriscodes in databases when a client registers to the system. The computing nodes can dynamically connect to and disconnect from the system. When a (k, n) schema is used, at least n nodes must be connected to the system.
- Result party: computes the public authentication result by combining the shared output of the computing nodes;
- Service node: acts like a proxy and selects the n computing nodes required for each client request. The nodes are chosen according to a load balancing algorithm: the service node keeps evidence of how many active requests each computing node is serving at a given moment and, when a new authentication request is made, the service node selects those computing nodes that manage the fewest requests. The sets of n IP addresses of the selected computing nodes are sent to the clients before they can connect to the computing nodes for authentication or registration operations. The service node is also responsible for key management: the clients and the computing nodes provide an RSA public key every time they connect to the system and the server generates AES keys that are sent both to the clients and to the selected computing nodes after being encrypted with the corresponding RSA public keys. Clients encrypt the password/iriscodes shares using the AES keys in order to avoid man-in-the-middle attacks on the communication channels established with the computing nodes.

The steps for authentication operations (similar steps are used for registrations, excepting SMPC processing and secret reconstruction which are replaced by an insert operation in the distributed databases) are as follows:

1. The client connects to the service node and sends its public RSA key to the node;
2. The service node selects the IP addresses of n available computing nodes;
3. The service node generates an AES key, which is sent to each selected computing node after being encrypted with the node's RSA public key;
4. The service node sends to the client the selected computing nodes IPs and the AES key encrypted with the client's RSA public key;
5. The client divides his password/iriscodes into secret shares and encrypts the shares using the AES key;
6. The client connects to the selected computing nodes and sends the encrypted secret shared values to them, together with the plaintext username;
7. The computing nodes connect to one another and perform a SMPC authentication operation using the client's shares, decrypted with the received AES key, and the shares retrieved from databases;
8. The computing nodes send the secret shared result to the result party;
9. The result party reveals if the authentication succeeded.

The SMPC system security analysis and threats resistance:

The SMPC authentication protocols presented in this paper are secure in the semi-honest model and they rely on the (k, n) Shamir secret sharing threshold schema. Before being sent to the computing nodes, the biometric information and the passwords are splitted into secret shares, such that at least k out of the n shares must be known in order to reconstruct the initial values. The client's authentication data is also stored as secret shares in distributed databases. The biometric information and the passwords are never reconstructed during the authentication process as matching is performed using SMPC. The system tolerates at most $k - 1$ corrupted computing nodes such that security is not broken.

The shares transmitted between the input party and the computing nodes are encrypted using AES, which makes our system resistant against sniffing and man-in-the-middle attacks, even if more than $k - 1$ communication channels are eavesdropped. More-

over, different AES keys are used for every authentication operation, such that replay attacks are prevented. If the legitimate client sends his secret data to the computing nodes and then an attacker tries to use the same messages for another authentication, the stolen encrypted shares are decrypted using a wrong AES key. However, the values transmitted between the computing nodes are secret shared but they are not otherwise encrypted because, as our experimental results show, the number of messages exchanged between the computing nodes is large and encrypting them would represent a significant overhead.

A possible solution in order to secure the communication between the computing nodes against sniffing and man-in-the-middle attacks is to encrypt the messages exchanged between these nodes with a high-performance stream cipher such as Rabbit [22] or Trivium [23]. The secret keys needed by the stream ciphers can be encrypted using asymmetric protocols and distributed by the service node in the same way the AES keys are currently exchanged.

Several well-known attacks against password-based authentication protocols are presented in [14]: dictionary attacks, brute force, lookup tables, reverse lookup tables, hybrid attacks (a combination of different attacks). In [16], the main threats against biometric authentication systems are described. In our system, knowing $k - 1$ shares does not reduce the keyspace in order to brute force one more share. While the brute force attacks can be prevented using long passwords, the dictionary attacks can be prevented using complex passwords such that they are not among the preselected words and phrases tried by an attacker. Also, blind brute force and even set covering (an optimal brute-force attack) are not feasible for 6400-bit iriscodes. Lookup tables are based on precalculated hashes of possible passwords. As different sets of secret shared values can be obtained by secret sharing a password multiple times, the construction of lookup tables is very difficult. Moreover, the secret shares are distributed and stored in multiple databases, which makes this attack even harder to succeed in our system.

4. Authentication Protocols

This section presents a password-based SMPC authentication protocol and an iris-based SMPC authentication protocol, both secure in the semi-honest adversary model. The authentication protocols use four main inner algorithms: *SMPC_Equality* (the equality of two secret shared values), *SMPC_Comparison* (the comparison of two secret shared values), *SMPC_Interval_test* (the belonging of a secret shared value to a public interval) and *SMPC_Bit_decomposition* (simplified bit decomposition). These algorithms, along with their sub-algorithms, are introduced in [21] and are integrated in our system with some adaptations such as the possibility to be applied on vectors of secret shared values. The algorithms are implemented in a similar way as the second set of SMPC protocols in our previous work [20], with the same constraint: they work only on positive integer values. The *SMPC_Prefix_OR/AND* and *SMPC_Fan_in_OR/AND* sub-algorithms are computed bit by bit, as in [24]. The *SMPC_Bit_decomposition* algorithm presents an efficiency improvement compared to the original bit decomposition algorithm from [25], although it uses the same bitwise sum sub-algorithm as in [25].

Each secret shared value is represented on $m = 2^r$ bits ($M = \frac{m}{8}$ bytes). A (k, n) Shamir secret sharing threshold schema is used and computations are performed in the finite field \mathbb{Z}_p , where p is the largest prime number less than 2^m . The computing nodes have IDs from 1 to n and they execute the same algorithm flow in parallel, each processing secret shared values $[s]_{id}$ computed from the secret value s by Lagrange polynomial evaluation at point id . The computing nodes perform symmetrical operations during the authentication process. All algorithms were extended to work on vectors of shares. For example, *SMPC_VMultiply* $([u]_{id}, [v]_{id})$, where $[u]_{id}$ and $[v]_{id}$ are both vectors, means the multiplication in parallel of each pair of shares at the same index in the two vectors, during the same communication round. For clarity, the SMPC operations performed on vectors of shares are marked with “*SMPC_V*” and those performed on simple values are marked with “*SMPC*”.

The SMPC protocols involve multiple communication rounds between the computing nodes, when they exchange messages containing one or more secret shared partial results. The total number of values sent by all the computing nodes during a SMPC operation represents the communication cost. The evaluation of the number of communication rounds and the communication cost is performed using the method and the theoretical results from [20]. In the protocols described in this section, the final result of the match, revealed by the result party in the authentication system, is reconstructed at the end of each algorithm by the computing nodes in order to verify the correctness of the implementation during testing.

4.1. Password-Based Authentication Protocol

We consider a plaintext password represented on P_{LENGTH} bytes. Firstly, the password is padded with zeros so that its length in bytes is a multiple of M . Then, the password bytes are grouped in password chunks, each chunk containing M password bytes. Finally, the password chunks vector is splitted into secret shares according to the (k, n) Shamir threshold schema. n vectors of secret shared values are obtained. Each vector contains $P_{SHARES_NO} = \text{ceil}(\frac{P_{LENGTH}}{M})$ shares and it is stored in a different database. Each share is represented on M bytes. The secret sharing operation introduces randomness, so the results obtained by splitting the same password multiple times are not equal.

Before an authentication operation, the password is splitted again by the client and the shares are sent to n computing nodes with distinct IDs. The nodes perform SMPC in order to check the equality between the secret shared password stored in databases and the one provided by the client.

In Algorithm 1, the equality between all pairs of password shares is computed, then all the elements of the result vector are multiplied (a secret shared zero value, meaning that a pair of shares does not match, fails the entire authentication). The multiplication between all the vector's elements (Algorithm 2) is computed with a minimum number of communication rounds $P_{MUL_ROUNDS} = \text{ceil}(\log_2 P_{SHARES_NO})$ and a communication cost $P_{MUL_COST} < 2^{P_{MUL_ROUNDS}}$.

Algorithm 1: SMPC_Password_match

Input: vectors $[P_X]_{id}$ and $[P_Y]_{id}$

Output: result r

- 1 $[e]_{id} \leftarrow \text{SMPC}_V\text{Equality}([P_X]_{id}, [P_Y]_{id})$
 - 2 $[r]_{id} \leftarrow \text{SMPC}_V\text{Multiply_vector_elements}([e]_{id})$
 - 3 $r \leftarrow \text{SMPC_Declassify}([r]_{id})$
-

When forming groups of M password bytes from the plaintext password, the constraint is to obtain values smaller than the largest prime number $p < 2^m$. Otherwise, by secret sharing the password chunks (operation performed in \mathbb{Z}_p), the values equal to or greater than p are truncated.

If each password byte has the most significant bit unset, the above constraint is satisfied as each password chunk is smaller than p ($2^{m-1} < p < 2^m$). In the present system, passwords consisting of standard ASCII characters (ASCII codes below 128) are considered valid. The drawback of our method is that only $\frac{7m}{8}$ bits can be used for the password while the rest have values equal to '0'.

In Algorithm 2, $[v_i]_{id}$ represents the share at index i in a vector of shares $[v]_{id}$ and $[v_{i:j}]_{id}$ indicates the shares corresponding to indexes i to j .

SMPC_Password_match algorithm evaluation

- Number of communication rounds:
The equalities for each pair of shares at the same index in two vectors are computed in parallel. The total number of communication rounds is the sum of: (1) the number of rounds needed for a SMPC equality operation, (2) the number of rounds for

$SMPC_V_Multiply_vector_elements$ and (3) the number of rounds for the reconstruction of the result.

$$R = R(SMPC_Equality) + R(SMPC_V_Multiply_vector_elements) + R(SMPC_Declassify) = 2m + P_{MUL_ROUNDS} + 4$$

- **Communication cost:**
The total communication cost is the sum of: (1) the cost needed for P_{SHARES_NO} equality operations, (2) the cost for $SMPC_V_Multiply_vector_elements$ and (3) the cost for the reconstruction of the result.

$$C = P_{SHARES_NO} \cdot C(SMPC_Equality) + C(SMPC_V_Multiply_vector_elements) + C(SMPC_Declassify) \\ = n(n-1)[P_{SHARES_NO} \cdot (4m-2) + P_{MUL_COST}] + n(k-1)[P_{SHARES_NO} \cdot (m+2) + 1]$$

Algorithm 2: $SMPC_V_Multiply_vector_elements$

Input: vector $[v]_{id}$
Output: result $[r]_{id}$, communication cost P_{MUL_COST}

```

1  $P_{MUL\_COST} \leftarrow 0$ 
2  $l \leftarrow len([v]_{id})$ 
3 while  $l > 1$  do
4    $h \leftarrow floor(\frac{l}{2})$ 
5   if  $l$  is odd then
6      $[last]_{id} \leftarrow [v_{2h+1}]_{id}$ 
7      $[v]_{id} \leftarrow SMPC_V\_Multiply([v_{1:h}]_{id}, [v_{h+1:2h}]_{id})$ 
8     Append  $[last]_{id}$  to  $[v]_{id}$ 
9      $l \leftarrow h + 1$ 
10  else
11     $[v]_{id} \leftarrow SMPC_V\_Multiply([v_{1:h}]_{id}, [v_{h+1:2h}]_{id})$ 
12     $l \leftarrow h$ 
13  end
14   $P_{MUL\_COST} \leftarrow P_{MUL\_COST} + h$ 
15 end
16  $[r]_{id} \leftarrow [v_1]_{id}$ 

```

4.2. Iris-Based Authentication Protocol

A reliable and widely used method for iris pattern recognition is based on the Hamming Distance, which measures the dissimilarity between two iris feature vectors. Two bit vectors of equal length I_{LENGTH} are extracted from an eye image X : the iriscodes C_X and a mask M_X marking all bits unoccluded by artifacts such as eyelashes. The HD is computed according to Equation (3).

$$HD = \frac{\|(C_X \oplus C_Y) \cap M_X \cap M_Y\|}{\|M_X \cap M_Y\|} \quad (3)$$

The result should not exceed a given threshold ('0' means perfect match between the two iriscodes).

Algorithm 3 illustrates how to compute the Hamming Distance through SMPC. The iriscodes and masks are vectors of length I_{LENGTH} , containing secret shared bits. The SMPC comparison with threshold t is computed, where t represents a public integer value, $1 \leq t \leq 100$. A zero numerator represent perfect match and the comparison (strict inequality) never passes with a zero denominator. In order to avoid sum overflow, I_{LENGTH} should be chosen such that $I_{LENGTH} * 100 < p$.

SMPC_IM algorithm evaluation

- **Number of communication rounds:**

The multiplications on lines 1 and 2 are performed in parallel, during a single communication round between the computing nodes. Also, the multiplications for each pair of shares at the same index in two vectors are computed in parallel. The total number of communication rounds is the sum of: (1) two rounds for SMPC multiplications, (2) the number of rounds for a SMPC comparison and (3) the number of rounds for the reconstruction of the result.

$$R = 2 \cdot R(\text{SMPC_Multiply}) + R(\text{SMPC_Comparison}) + R(\text{SMPC_Declassify}) = 2m + 9$$

- Communication cost:

The total communication cost is the sum of: (1) the cost needed for three multiplications performed on vectors with I_{LENGTH} elements, (2) the cost for a comparison operation and (3) the cost for the reconstruction of the result.

$$C = 3 \cdot I_{LENGTH} \cdot C(\text{SMPC_Multiply}) + C(\text{SMPC_Comparison}) + C(\text{SMPC_Declassify}) \\ = n(n-1)(12m + 3 \cdot I_{LENGTH} - 1) + n(k-1)(3m + 7)$$

Algorithm 3: SMPC_Iris_match (SMPC_IM)

Input: vectors $[C_X]_{id}$, $[C_Y]_{id}$, $[M_X]_{id}$ and $[M_Y]_{id}$, threshold t

Output: result r

- 1 $[a]_{id} \leftarrow \text{SMPC}_V\text{Multiply}([C_X]_{id}, [C_Y]_{id})$
 - 2 $[b]_{id} \leftarrow \text{SMPC}_V\text{Multiply}([M_X]_{id}, [M_Y]_{id})$
 - 3 $[c]_{id} \leftarrow [C_X]_{id} + [C_Y]_{id} - 2 \cdot [a]_{id}$
 - 4 $[d]_{id} \leftarrow \text{SMPC}_V\text{Multiply}([b]_{id}, [c]_{id})$
 - 5 $[num]_{id} \leftarrow \text{sum}([d]_{id})$
 - 6 $[den]_{id} \leftarrow \text{sum}([b]_{id})$
 - 7 $[r]_{id} \leftarrow \text{SMPC_Comparison}(100 \cdot [num]_{id}, t \cdot [den]_{id})$
 - 8 $r \leftarrow \text{SMPC_Declassify}([r]_{id})$
-

5. Iris-Based Authentication Enhancements

In this section, three enhancements are added to the SMPC_Iris_match algorithm in order to increase security, improve recognition accuracy and decrease database storage requirements. The extra checks and operations are marked with blue in the next algorithms.

5.1. Security Enhancement

The SMPC_Iris_match algorithm considers only those bits unoccluded by artifacts. If an attacker sends a random iriscode $[C_Y]$ with a mask $[M_Y]$ containing only one set bit and if the corresponding bit in the stored mask $[M_X]$ is also set, the chances are 50 percent for the authentication to be successful.

An extra check is added in Algorithm 4, so that at least k bits to be considered when matching iriscodes. k is a public value but the actual number of pairs of bits at the same index which are set in both masks is not revealed.

SMPC_IMMT algorithm evaluation

- Number of communication rounds:

The total number of communication rounds is the sum of: (1) the number of rounds needed for SMPC_IM, (2) the number of rounds for an interval test and (3) the number of rounds for an extra multiplication.

$$R = R(\text{SMPC_IM}) + R(\text{SMPC_Interval_test}) + R(\text{SMPC_Multiply}) = 4m + 14$$

- Communication cost:

The total communication cost is the sum of: (1) the cost needed for $SMPC_IM$, (2) the cost for an interval test and (3) the cost for a multiplication.

$$C = C(SMPC_IM) + C(SMPC_Interval_test) + C(SMPC_Multiply) \\ = n(n-1)(17m + 3 \cdot I_{LENGTH} - 2) + n(k-1)(4m + 9)$$

Algorithm 4: $SMPC_Iris_match_with_masks_threshold$ ($SMPC_IMMT$)

Input: vectors $[C_X]_{id}$, $[C_Y]_{id}$, $[M_X]_{id}$ and $[M_Y]_{id}$, thresholds t and k

Output: result r

```

1  $[a]_{id} \leftarrow SMPC_V\_Multiply([C_X]_{id}, [C_Y]_{id})$ 
2  $[b]_{id} \leftarrow SMPC_V\_Multiply([M_X]_{id}, [M_Y]_{id})$ 
3  $[c]_{id} \leftarrow [C_X]_{id} + [C_Y]_{id} - 2 \cdot [a]_{id}$ 
4  $[d]_{id} \leftarrow SMPC_V\_Multiply([b]_{id}, [c]_{id})$ 
5  $[num]_{id} \leftarrow sum([d]_{id})$ 
6  $[den]_{id} \leftarrow sum([b]_{id})$ 
7  $[e]_{id} \leftarrow SMPC\_Comparison(100 \cdot [num]_{id}, t \cdot [den]_{id})$ 
8  $[f]_{id} \leftarrow SMPC\_Interval\_test([denom]_{id}, k, p - 1)$ 
9  $[r]_{id} \leftarrow SMPC\_Multiply([e]_{id}, [f]_{id})$ 
10  $r \leftarrow SMPC\_Declassify([r]_{id})$ 

```

5.2. Accuracy Enhancement

Reference [6] presents a method to improve the accuracy of iris recognition through fusion of Hamming Distance and fragile bit distance. An iriscode bit is consistent if it has the same value for most images of the same iris, otherwise it is considered fragile. The authors explain how to determine if a bit is consistent from a single image: when applying a Gabor filter to the image to a specific location during feature vector extraction, the magnitude of the real part, respectively the imaginary part, of the resulting complex number is considered. A large magnitude indicates a corresponding consistent bit in the iriscode and a small magnitude indicates a fragile bit. For each iris image X , three vectors are extracted: the iriscode C_X , the occlusion mask M_X and the fragility mask F_X . FBD is computed according to Equation (4).

$$FBD = \frac{\| \overline{F_X \cap F_Y} \cap M_X \cap M_Y \|}{\| M_X \cap M_Y \|} \quad (4)$$

The fusion score is computed from HD and FBD considering weight α (Equation (5)).

$$F_{score} = \alpha * HD + (1 - \alpha) * FBD \quad (5)$$

Algorithm 5 illustrates how the fusion score can be computed through SMPC. The iriscodes, occlusion masks and fragility masks are vectors of length I_{LENGTH} , containing secret shared bits. 1_V represents a vector of length I_{LENGTH} , with all the elements equal to '1'. We consider $\alpha = 0.6$ (the value for which the lowest equal error rate was obtained in [6]).

SMPC_IMFBD algorithm evaluation

- Number of communication rounds:
The multiplications on lines 1, 2 and 3 are computed in parallel during the first communication round and multiplications on lines 6 and 7 are computed in parallel during the second round. The total number of communication rounds is equal to the number of rounds needed for $SMPC_IM$.

$$R = R(SMPC_IM) = 2m + 9$$

- Communication cost:

The total communication cost is the sum of: the cost needed for $SMPC_IM$ and (2) the cost for two extra multiplications performed on vectors with I_{LENGTH} elements.

$$\begin{aligned} C &= C(SMPC_IM) + 2 \cdot I_{LENGTH} \cdot C(SMPC_Multiply) \\ &= n(n-1)(12m + 5 \cdot I_{LENGTH} - 1) + n(k-1)(3m + 7) \end{aligned}$$

Algorithm 5: $SMPC_Iris_match_with_FBD$ ($SMPC_IMFBD$)

Input: vectors $[C_X]_{id}$, $[C_Y]_{id}$, $[M_X]_{id}$, $[M_Y]_{id}$, $[F_X]_{id}$ and $[F_Y]_{id}$, threshold t

Output: result r

```

1  $[a]_{id} \leftarrow SMPC_V\_Multiply([C_X]_{id}, [C_Y]_{id})$ 
2  $[b]_{id} \leftarrow SMPC_V\_Multiply([M_X]_{id}, [M_Y]_{id})$ 
3  $[c]_{id} \leftarrow SMPC_V\_Multiply([F_X]_{id}, [F_Y]_{id})$ 
4  $[d]_{id} \leftarrow [C_X]_{id} + [C_Y]_{id} - 2 \cdot [a]_{id}$ 
5  $[e]_{id} \leftarrow 1_V - [c]_{id}$ 
6  $[f]_{id} \leftarrow SMPC_V\_Multiply([b]_{id}, [d]_{id})$ 
7  $[g]_{id} \leftarrow SMPC_V\_Multiply([b]_{id}, [e]_{id})$ 
8  $[num]_{id} \leftarrow sum([f]_{id})$ 
9  $[fnum]_{id} \leftarrow sum([g]_{id})$ 
10  $[den]_{id} \leftarrow sum([b]_{id})$ 
11  $[r]_{id} \leftarrow SMPC\_Comparison(60 \cdot [num]_{id} + 40 \cdot [fnum]_{id}, t \cdot [den]_{id})$ 
12  $r \leftarrow SMPC\_Declassify([r]_{id})$ 

```

5.3. Database Storage Requirements Enhancement

For all the previous SMPC iris match algorithms, each bit in iriscodes or in masks is secret shared according to the (k, n) schema and n shares, each represented on M bytes, are generated. Consequently, a plaintext iriscodes/mask containing I_{LENGTH} bits occupies $I_{LENGTH} \cdot M$ bytes in each of the n databases after secret sharing.

In order to reduce the storage requirements, the plaintext iriscodes/mask is padded with zeros so that its length is a multiple of $m - 1$, where $M = \frac{m}{8}$, then groups of $m - 1$ bits are formed. A vector containing values represented on m bits is generated, where the most significant bit of each element is unset and the rest of the bits are represented by one of the groups previously formed. All the elements of the vector should be smaller than the prime p in order not to be truncated during secret sharing, which is performed in \mathbb{Z}_p . The constraint is satisfied as each value has the most significant bit equal to '0'. By secret sharing the vector, n result vectors are obtained, each containing $I_{SHARES_NO} = \text{ceil}(\frac{I_{LENGTH}}{m-1})$ shares.

According to the above method, where groups of bits are treated as decimal values, the database storage requirements for secret shared iriscodes and masks is reduced about $m - 1$ times. For example, if we consider a plaintext iriscodes/mask with 6400 bits and shares represented on $M = 8$ bytes, 8×6400 bytes = 50 KB are needed in each of the n databases in order to store the secret shared iriscodes/mask without using bit decomposition. If each database has a storage capacity of 1 GB, 10,485 secret shared iriscodes can be stored along with their occlusion masks. If we use bit decomposition, $8 \times \text{ceil}(\frac{6400}{63}) = 816$ bytes are needed in each of the n databases for a secret shared iriscodes/mask. Consequently, 657,930 iriscodes-mask pairs can be stored if each database has a capacity of 1 GB.

In Algorithm 6, $[C_X]_{id}$ and $[M_X]_{id}$ represent stored vectors of decimal values. Before computing the Hamming Distance through SMPC, bit decomposition is applied on these values in order to obtain vectors of secret shared bits. As $SMPC_Bit_decomposition$ is relatively expensive and time consuming, the clients that try to authenticate to the system send $[C_Y]$ and $[M_Y]_{id}$ directly as secret shared bit vectors. $SMPC_Bit_decomposition$ can be applied only if M is a power of 2.

SMPC_IMBD algorithm evaluation

- Number of communication rounds:
The bit decomposition operations for all the elements of a vector are computed in parallel. Also, the operations on lines 1 and 2 are computed in parallel during the

same communication rounds between the computing nodes. The total number of communication rounds is the sum of: (1) the number of rounds needed for *SMPC_IM* and (2) the number of rounds for a bit decomposition operation.

$$R = R(\text{SMPC_IM}) + R(\text{SMPC_Bit_decomposition}) = \frac{9m}{2} + r + 13,$$

where:

$$R(\text{SMPC_Bit_decomposition}) = \frac{5m}{2} + r + 4$$

- Communication cost:

The total communication cost is the sum of: (1) the cost needed for *SMPC_IM* and (2) the cost for two bit decomposition operations performed on vectors with I_{LENGTH} elements.

$$C = C(\text{SMPC_IM}) + 2 \cdot I_{SHARES_NO} \cdot C(\text{SMPC_Bit_decomposition}) = n(n-1)\{12m + 3 \cdot I_{LENGTH} - 1 + 2 \cdot I_{SHARES_NO} \cdot [m \cdot r + 4m + 2 \cdot \sum_{i=2}^r (i-1) \binom{r}{i}]\} + n(k-1)[3m + 7 + 2 \cdot I_{SHARES_NO} \cdot (m+2)]$$

where:

$$C(\text{SMPC_Bit_decomposition}) = n(n-1)[m \cdot r + 4m + 2 \cdot \sum_{i=2}^r (i-1) \binom{r}{i}] + n(k-1)(m+2)$$

Algorithm 6: SMPC_Iris_match_with_bit_decomposition (SMPC_IMBD)

Input: vectors $[C_X]_{id}$, $[C_Y]_{id}$, $[M_X]_{id}$ and $[M_Y]_{id}$, threshold t

Output: result r

```

1  $[a]_{id} \leftarrow \text{SMPC}_V\text{\_Bit\_Decomposition}([C_X]_{id})$ 
2  $[b]_{id} \leftarrow \text{SMPC}_V\text{\_Bit\_Decomposition}([M_X]_{id})$ 
3  $[c]_{id} \leftarrow \text{SMPC}_V\text{\_Multiply}([a]_{id}, [C_Y]_{id})$ 
4  $[d]_{id} \leftarrow \text{SMPC}_V\text{\_Multiply}([b]_{id}, [M_Y]_{id})$ 
5  $[e]_{id} \leftarrow [a]_{id} + [C_Y]_{id} - 2 \cdot [c]_{id}$ 
6  $[f]_{id} \leftarrow \text{SMPC}_V\text{\_Multiply}([d]_{id}, [e]_{id})$ 
7  $[num]_{id} \leftarrow \text{sum}([f]_{id})$ 
8  $[den]_{id} \leftarrow \text{sum}([d]_{id})$ 
9  $[r]_{id} \leftarrow \text{SMPC\_Comparison}(100 \cdot [num]_{id}, t \cdot [den]_{id})$ 
10  $r \leftarrow \text{SMPC\_Declassify}([r]_{id})$ 

```

6. Experimental Results

Experiments were conducted on a computer equipped with 8 GB of RAM and an Intel(R) Core(TM) i7 CPU (64 bits) at 1.80 GHz, with 4 cores and 8 logical processors, running Windows 10 operating system.

All the components of the distributed authentication system, along with the SMPC authentication protocols, are implemented in Python 3. The communication between the components is implemented using sockets. The SMPC system uses 1024-bits RSA keys and AES GCM algorithm with 256-bits keys. Passwords, iris codes and masks were randomly generated for all testing scenarios and the SMPC authentication results were compared to those expected.

6.1. Experimental Evaluation of the SMPC Authentication System

In order to verify the correctness of our implementation and to evaluate the authentication system in terms of efficiency, a testing environment was created using VMware. Five virtual machines with Ubuntu 16.4 operating system were used: one machine for the service node, three machines as computing nodes and one machine having both the role of client and of result party. We chose MongoDB for persistent storage and the physical machine hosted the databases.

For this experiment, the (2,3) secret sharing threshold schema was used. For SMPC password-based authentication, we considered passwords containing 64 standard ASCII characters (with ASCII codes below 128) and secret shared values represented on M bytes, with $1 \leq M \leq 8$. For iris-based authentication, we considered the *SMPC_Iris_match_with_masks_threshold* algorithm, iriscodes containing 6400 bits and secret shared values represented on M bytes, with $3 \leq M \leq 8$. The condition $I_{LENGTH} * 100 < p$ is not fulfilled for $M = 2, p = 65521$.

Table 2 presents the efficiency of the authentication protocols, when 250 serial authentication operations were performed for each value of M . Efficiency is computed as the duration of a single operation in milliseconds. The time needed for the clients RSA keys generation, the AES key exchange and shares encryption is also considered. For both the authentication methods, efficiency is constant with M , but large values for M provide better security in the SMPC system than small values.

Table 2. Efficiency for the SMPC authentication system considering the (2,3) secret sharing schema.

SMPC System—Time Efficiency [ms] for Schema (2,3)		
Bytes no. M	SMPC Password-Based Authentication	SMPC Iris-Based Authentication
1	1063	-
2	1072	-
3	1070	1301
4	1091	1363
5	1049	1437
6	1079	1372
7	1121	1392
8	1101	1441

6.2. Experimental Evaluation of the SMPC Authentication Protocols

For each authentication algorithm we evaluated efficiency and two SMPC metrics: the number of communication rounds and the communication cost. Efficiency is computed in milliseconds as the processing time needed for a single SMPC operation. The number of communication rounds counts how many times the computing nodes exchange messages with one another during the execution of a SMPC algorithm. Those messages contain one or more secret shared partial results, each represented on M bytes. The total number of values transmitted between the nodes represents the communication cost. These experiments were performed on the physical machine, considering only the computing nodes and AES encryption was not applied. Secret shared data was already distributed to the computing nodes when evaluation started and the nodes also performed the reconstruction of the authentication result. For each algorithm and each value of M , 2500 serial operations were performed and the average values of the metrics for one operation were computed.

Evaluation for SMPC password-based authentication:

Passwords of various lengths P_{LENGTH} , containing standard ASCII characters, were considered. As presented in Table 3 for schema (2,3) and in Table 4 for schema (3,5), the duration of the password matching operations increases linearly with M . However, the best security is achieved for the largest value of M .

Table 3. Efficiency for SMPC password matching considering the (2,3) secret sharing schema.

Password Matching—Time Efficiency [ms] for Schema (2,3)			
Bytes no. M	$P_{LENGTH} = 32$ Chars	$P_{LENGTH} = 64$ Chars	$P_{LENGTH} = 128$ Chars
1	12	23	42
2	17	31	57
3	20	36	63
4	23	39	72
5	28	47	83
6	34	54	97
7	37	60	106
8	40	67	121

Table 4. Efficiency for SMPC password matching considering the (3,5) secret sharing schema.

Password Matching—Time Efficiency [ms] for Schema (3,5)			
Bytes no. M	$P_{LENGTH} = 32$ Chars	$P_{LENGTH} = 64$ Chars	$P_{LENGTH} = 128$ Chars
1	30	57	110
2	41	76	146
3	47	85	165
4	52	92	171
5	63	106	200
6	73	124	224
7	77	131	234
8	82	139	250

Table 5 presents the number of communication rounds for the (2,3) and (3,5) secret sharing schemas. Similar results for this metric were obtained for both schemas as the flow of the algorithm does not change with the n . The number of computing nodes n has no impact on the number of rounds but affects the communication cost and consequently the efficiency of the algorithm when different schemas are used.

Table 5. Number of communication rounds for SMPC password matching considering the (2,3) and (3,5) secret sharing schemas.

Password Matching—Number of Rounds for Schemas (2,3) and (3,5)			
Bytes no. M	$P_{LENGTH} = 32$ Chars	$P_{LENGTH} = 64$ Chars	$P_{LENGTH} = 128$ Chars
1	32.18	37.03	41.03
2	40.07	41.2	42.3
3	56	57	58
4	71	72	73
5	87	88	89
6	103	104	105
7	119	120	121
8	134	135	136

Tables 6 and 7 present the communication cost for the two schemas. Although the number of rounds increases with M , the communication cost remains constant. The cost

is higher for the inner *SMPC_Equality* operations when shares are represented on more bytes than when shares are represented on less bytes. But for large values of M fewer password shares are generated than for small values of M : $P_{SHARES_NO} = \text{ceil}(\frac{P_{LENGTH}}{M})$.

Table 6. Communication cost for SMPC password matching considering the (2,3) secret sharing schema.

Password Matching—Communication Cost for Schema (2,3)			
Bytes no. M	$P_{LENGTH} = 32$ Chars	$P_{LENGTH} = 64$ Chars	$P_{LENGTH} = 128$ Chars
1	7026.66	14,062.85	28,139.99
2	6910.01	13,824.17	27,649.88
3	7125	14,253	27,861
4	6909	13,821	27,645
5	7557	14,037	28,077
6	7773	14,253	28,509
7	7557	15,117	28,725
8	6909	13,821	27,645

Table 7. Communication cost for SMPC password matching considering the (3,5) secret sharing schema.

Password Matching—Communication Cost for Schema (3,5)			
Bytes no. M	$P_{LENGTH} = 32$ Chars	$P_{LENGTH} = 64$ Chars	$P_{LENGTH} = 128$ Chars
1	23,413.26	46,872.56	93,760.96
2	23,035.59	46,082.89	92,168.06
3	23,750	47,510	92,870
4	23,030	46,070	92,150
5	25,190	46,790	93,590
6	25,910	47,510	95,030
7	25,190	50,390	95,750
8	23,030	46,070	92,150

Evaluation for SMPC iris-based authentication:

Iriscodes and masks containing $I_{LENGTH} = 6400$ bits were considered. The penalty added by each enhanced algorithm to the base *SMPC_Iris_match* algorithm is provided in parentheses (+x) in all the following tables.

Tables 8 and 9 present the efficiency of the SMPC iris-based authentication algorithms considering the (2,3) and (3,5) secret sharing schemas. For iris codes of the same length and shares represented on 46 bits, the authors of [18] obtained an efficiency of 120 milliseconds using the SMPC iris-based authentication protocol implemented in their system. However, they use different testing machine specifications, $n = 2$ computing parties and the SPDZ protocol instead of Shamir secret sharing. For $n = 3$ and shares represented on 48 bits, we obtained an efficiency of 209 milliseconds.

Checking the number of set bits in the occlusion masks (*SMPC_IMMT*) has a small impact considering efficiency and represents an important security enhancement. HD and FDB fusion (*SMPC_IMFBD*) increases the accuracy of the iris-based authentication protocol by 8% according to [6], but the efficiency decreases by 1.55 times according to our results when this enhancement is integrated in the SMPC system. When bit decomposition (*SMPC_IMBD*) is used, the database storage requirements for secret shared iris codes decrease but performance is significantly affected. For $M = 8$, the duration of the algorithm

is 15 times longer than for the base *SMPC_IM* algorithm but the storage requirements decrease about 63 times. As in the case of passwords, the greater *M* is, the better the security of the authentication protocols is.

Table 8. Efficiency for SMPC iris matching considering the (2,3) secret sharing schema.

Iris Matching—Time Efficiency [ms] for Schema (2,3)				
Bytes no. <i>M</i>	SMPC_IM	SMPC_IMMT	SMPC_IMFBD	SMPC_IMBD
3	176	182 (+6)	282 (+106)	-
4	189	197 (+8)	301 (+112)	2358 (+2169)
5	195	205 (+10)	311 (+116)	-
6	209	221 (+12)	328 (+119)	-
7	220	234 (+14)	341 (+121)	-
8	229	249 (+20)	354 (+125)	3519 (+3290)

Table 9. Efficiency for SMPC iris matching considering the (3,5) secret sharing schema.

Iris Matching—Time Efficiency [ms] for Schema (3,5)				
Bytes no. <i>M</i>	SMPC_IM	SMPC_IMMT	SMPC_IMFBD	SMPC_IMBD
3	505	519 (+14)	830 (+325)	-
4	542	561 (+19)	878 (+336)	6033 (+5491)
5	558	580 (+22)	911 (+353)	-
6	590	621 (+31)	950 (+360)	-
7	609	643 (+34)	979 (+370)	-
8	639	680 (+41)	1017 (+378)	8538 (+7899)

The number of communication rounds is similar for the (2,3) and (3,5) schemas (Table 10). The *SMPC_IMFBD* algorithm adds no communication round to the base algorithm, while *SMPC_IMMT* adds more than 50 new rounds. However, *SMPC_IMMT* has smaller negative impact considering efficiency than *SMPC_IMFBD* has.

Table 10. Number of communication rounds for SMPC iris matching considering the (2,3) and (3,5) secret sharing schemas.

Iris Matching—Number of rounds for schemas (2,3) and (3,5)				
Bytes no. <i>M</i>	SMPC_IM	SMPC_IMMT	SMPC_IMFBD	SMPC_IMBD
3	57	110 (+53)	57 (+0)	-
4	73	142 (+69)	73 (+0)	162 (+89)
5	89	174 (+85)	89 (+0)	-
6	105	206 (+101)	105 (+0)	-
7	121	238 (+117)	121 (+0)	-
8	137	270 (+133)	137 (+0)	307 (+170)

The communication cost (presented in Tables 11 and 12) is more related to performance than the number of rounds. It can be observed that those algorithms that have an increased cost come also with significant efficiency penalties.

Table 11. Communication cost for SMPC iris matching considering the (2,3) secret sharing schema.

Iris Matching—Communication Cost for Schema (2,3)				
Bytes no. <i>M</i>	SMPC_IM	SMPC_IMMT	SMPC_IMFBD	SMPC_IMBD
3	117,159	117,951 (+792)	193,959 (+76,800)	-
4	117,807	118,863 (+1056)	194,607 (+76,800)	1,118,859 (+1,001,052)
5	118,455	119,775 (+1320)	195,255 (+76,800)	-
6	119,103	120,687 (+1584)	195,903 (+76,800)	-
7	119,751	121,599 (+1848)	196,551 (+76,800)	-
8	120,399	122,511 (+2112)	197,199 (+76,800)	1,259,943 (+1,139,544)

Table 12. Communication cost for SMPC iris matching considering the (3,5) secret sharing schema.

Iris Matching—Communication Cost for Schema (3,5)				
Bytes no. <i>M</i>	SMPC_IM	SMPC_IMMT	SMPC_IMFBD	SMPC_IMBD
3	390,530	393,170 (+2640)	646,530 (+256,000)	-
4	392,690	396,210 (+3520)	648,690 (+256,000)	3,729,530 (+3,336,840)
5	394,850	399,250 (+4400)	650,850 (+256,000)	-
6	397,010	402,290 (+5280)	653,010 (+256,000)	-
7	399,170	405,330 (+6160)	655,170 (+256,000)	-
8	401,330	408,370 (+7040)	657,330 (+256,000)	4,199,810 (+3,798,480)

7. Conclusions

The idea of using SMPC techniques for authentication is a relatively old one but, in the past, it was not facilitated by the hardware capabilities. In this paper, we considered classical password-based and iriscodes-based authentication algorithms and translated them into a SMPC form. We also presented how several enhancements regarding security, accuracy and database storage requirements can be added to the Hamming-Distance-based SMPC algorithm and how they affect the performance of the system. The architecture of the SMPC authentication system was described, along with the interactions between its components.

The efficiency of the algorithms was evaluated, along with two SMPC metrics: the number of communication rounds and the communication cost. This evaluation is relevant as it shows that the performance of the SMPC authentication protocols based on secret sharing facilitates their integration in real-world authentication systems, although the complex SMPC operations involve a large amount of information transmitted through the network. As far as we know, no similar evaluation was performed for SMPC authentication protocols based on Shamir secret sharing threshold schema. Using the (2,3) secret sharing schema, shares represented on 8 bytes and passwords containing 64 standard ASCII characters, the SMPC password matching algorithm is executed by the computing nodes in 0.067 s and the entire password-based authentication process (the SMPC password matching algorithm together with the AES/RSA encryptions/decryptions and the interaction between all the components of the system) is performed in 1.101 s. Using the same secret sharing configurations and 6400-bit iriscodes, the base SMPC iriscodes matching algorithm (*SMPC_IM*) is executed by the computing nodes in 0.229 s and the entire iris-based authentication process is performed in 1.441 s. The performance is comparable to that obtained in other similar systems. According to our results, the communication cost has a considerable impact on the efficiency of the protocols. The iris-based authentication algorithms whose enhancement implies additional SMPC operations with high communication cost have a significant performance penalty. For example, the enhanced iriscodes matching algorithm that used bit decomposition (*SMPC_IMBD*) has a communication cost that is about

10 times higher than the cost of *SMPC_IM*. However, the time needed in order to execute the *SMPC_IM* algorithm is about 13 times smaller than the time needed in order to execute the *SMPC_IMBD* algorithm.

As future work, we intend to integrate SMPC matching algorithms for multiple biometric traits into our system. The security of the system as a whole can be further improved, by performing message authentication between the computing nodes. And nonetheless, we aim to research and benchmark alternative SMPC protocols for improving the performance of the system.

Author Contributions: Conceptualization, D.-E.F. and K.M.; methodology, D.-E.F. and K.M.; software, D.-E.F.; validation, D.-E.F.; formal analysis, D.-E.F.; investigation, D.-E.F. and K.M.; resources, D.-E.F., K.M. and A.S.; data curation, D.-E.F.; writing—original draft preparation, D.-E.F.; writing—review and editing, K.M. and A.S.; visualization, K.M. and A.S.; supervision, K.M.; project administration, K.M.; funding acquisition, K.M. and A.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: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

SMPC	Secure Multiparty Computation
HD	Hamming Distance
FBD	Fragile Bit Distance

References

- Zhao, C.; Zhao, S.; Zhao, M.; Chen, Z.; Gao, C.Z.; Li, H.; Tan, Y. Secure Multi-Party Computation: Theory, practice and applications. *Inf. Sci.* **2019**, *476*, 357–372. [CrossRef]
- Bogdanov, D. *Foundations and Properties of Shamir's Secret Sharing Scheme, Research Seminar in Cryptography*; University of Tartu, Institute of Computer Science: Tartu, Estonia, 2007. Available online: http://kodu.ut.ee/~peeter_l/teaching/seminar07k/bogdanov.pdf (accessed on 16 May 2012).
- Bozkurt, I.N.; Guloglu, A.M.; Kaya, K.; Selcuk, A.A. Threshold Cryptography Based on Blakley Secret Sharing. In Proceedings of the Information Security and Cryptology Conference (ISC), Ankara, Turkey, 25–27 December 2008.
- Kaya, K.; Secuk, A.A.; Tezcan, Z. Threshold Cryptography Based on Asmuth-Bloom Secret Sharing. In Proceedings of the International Symposium on Computer and Information Sciences (ISCIS), Istanbul, Turkey, 1–3 November 2006; Lecture Notes in Computer Science (LNCS); Volume 4263, pp. 935–942.
- Daugman, J. How iris recognition works. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 21–30. [CrossRef]
- Hollingsworth, K.P.; Bowyer, K.W.; Flynn, P.J. Improved Iris Recognition through Fusion of Hamming Distance and Fragile Bit Distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2465–2476. [CrossRef] [PubMed]
- Keller, M.; Orsini, E.; Rotaru, D.; Scholl, P.; Soria-Vazquez, E.; Vivek, S. Faster Secure Multi-party Computation of AES and DES Using Lookup Tables. In Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS), Kanazawa, Japan, 10–12 July 2017; Lecture Notes in Computer Science (LNCS); Volume 10355, pp. 229–249.
- Doerner, J.; Kondi, Y.; Lee, E.; Shelat, A. Threshold ECDSA from ECDSA Assumptions: The Multiparty Case. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 1051–1066.
- Pattuk, E.; Kantarcioglu, M.; Ulusoy, H.; Malin, B. CheapSMC: A Framework to Minimize SMC Cost in Cloud. In Proceedings of the Data and Applications Security and Privacy XXX, DBSec, Trento, Italy, 18–20 July 2016; Lecture Notes in Computer Science (LNCS); Volume 9766, pp. 285–294.
- Liu, J.; Tian, Y.; Zhou, Y.; Xiao, Y.; Ansari, N. Privacy preserving distributed data mining based on secure multi-party computation. *Comput. Commun.* **2020**, *153*, 208–216. [CrossRef]
- Chen, V.; Pastro, V.; Raykova, M. Secure Computation for Machine Learning With SPDZ. *arXiv* **2019**, arXiv:1901.00329.
- Nair, D.G.; Binu, V.P.; Kumarc, G.S. An Improved E-voting scheme using Secret Sharing based Secure Multi-party Computation. *arXiv* **2015**, arXiv:1502.07469.

13. Naidu, P.S.; Kharat, R.; Tekade, R.; Mendhe, P.; Magade, V. E-Voting System Using Visual Cryptography & Secure Multi-party Computation. In Proceedings of the International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 12–13 August 2016.
14. Bissoli, A.; d'Amore, F. Authentication as a service: Shamir Secret Sharing with byzantine components. *arXiv* **2018**, arXiv:1806.07291.
15. Mohassel, P.; Rosulek, M.; Zhang, Y. Fast and Secure Three-party Computation: The Garbled Circuit Approach. In Proceedings of the 22nd ACM SIGSAC Conference, Denver, CO, USA, 12–16 October 2015; pp. 591–602.
16. Pagnin, E.; Mitrokotsa, A. Privacy-preserving biometric authentication: Challenges and directions. *Secur. Commun. Netw.* **2017**, *2017*, 7129505. [[CrossRef](#)]
17. Blanton, M.; Gasti, P. Secure and Efficient Protocols for Iris and Fingerprint Identification. In Proceedings of the European Symposium on Research in Computer Security (ESORICS), Leuven, Belgium, 12–14 September 2011; pp. 190–209.
18. Barni, M.; Droandi, G.; Lazzeretti, R.; Pignata, T. SEMBA: SEcure Multi-Biometric Authentication. *IET Biometr.* **2019**, *8*, 411–421. [[CrossRef](#)]
19. Archer, D.W.; Bogdanov, D.; Kamm, L.; Lindell, Y.; Nielsen, K.; Pagter, J.I.; Smart, N.P.; Wright, R.N. From Keys to Databases—Real-World Applications of Secure Multi-Party Computation. *Comput. J.* **2018**, *61*, 1749–1771. [[CrossRef](#)]
20. Fălămaș, D.E.; Marton, K. Performance Impact Analysis of Rounds and Amounts of Communication in Secure Multiparty Computation Based on Secret Sharing. In Proceedings of the 18th RoEduNet Conference: Networking in Education and Research, Galați, Romania, 10–12 October 2019; pp. 190–195.
21. Nishide, T.; Ohta, K. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In *Public Key Cryptography—PKC 2007; Lecture Notes in Computer Science (LNCS)*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4450, pp. 343–360.
22. Boesgaard, M.; Vesterager, M.; Pedersen, T.; Christiansen, J.; Scavenius, O. Rabbit: A New High-Performance Stream Cipher. In Proceedings of the International Workshop on Fast Software Encryption (FSE), Lund, Sweden, 24–26 February 2003; Lecture Notes in Computer Science (LNCS); Volume 2887, pp. 307–329.
23. De Canniere, C. Trivium: A Stream Cipher Construction Inspired by Block Cipher Design Principles. In Proceedings of the ISC, Samos, Greece, 30 August–2 September 2006; Lecture Notes in Computer Science (LNCS); Volume 4176, pp. 171–186.
24. Turban, T. A Secure Multi-Party Computation Protocol Suite Inspired by Shamir's Secret Sharing Scheme. Master's Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2014.
25. Damgård, I.; Fitzi, M.; Kiltz, E.; Nielsen, J.B.; Toft, T. Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation. In Proceedings of the Theory of Cryptography Conference (TCC), New York, NY, USA, 4–7 March 2006; Lecture Notes in Computer Science (LNCS); Volume 3876, pp. 285–304.