Impacts on Database Performance in a Privacy-Preserving Biometric Authentication Scenario

Veit Köppen, Christian Krätzer
Jana Dittmann, Gunter Saake

Faculty of Computer Science
Otto-von-Guericke University
Email: [vkoeppen|kraetzer|
jana.dittmann|gunter.saake]@ovgu.de

Claus Vielhauer

Department of Informatics and Media
Brandenburg University of Applied Sciences
Email: vielhauer@fh-brandenburg.de

***Abstract*—Nowadays, biometric data are more and more used within authentication processes. Such data are usually stored in databases and underlie inherent privacy concerns. Therefore, special attention should be paid to their handling. We propose an extension to an existing privacy preserving similarity verification system. The Paillier scheme, being an asymmetric as well as additive homomorphic cryptography approach, enables signal processing in the encrypted domain operations. Amongst other modifications, we introduce a padding approach to increase entropy for better filling the co-domain. As a result, we combine the benefits of signal processing in the encrypted domain with the advantages of salting. The concept of verification of encrypted biometric data comes at the cost of increased computational effort in contrast to already available biometric systems. Nevertheless, this additional cost is in many scenarios justified by addressing that most currently available biometric authentication systems lack sufficient privacy protection. In our evaluation, we focus on performance issues of the privacy-preserving biometric authentication scheme with respect to database response time. The results presented for different evaluations on the influence of numbers of users, template sizes, and cryptographic key lengths show that the increase in effort required caused by our extensions is negligible. Furthermore, our improved scheme lowers the error rates attached as well as it reduces the amount of data that is disclosed in an authentication attempt. Our work highlights that user- and privacy-centric approaches to authentication have become feasible in the last few years. Modern schemes, as the one discussed in this paper, are not only efficient but also make the usage of data mining techniques in the domain of user tracking much more difficult.**

***Index Terms*—Database Security; Homomorphic Encryption; Privacy; Multi-Computer Scenarios; Database Performance; Biometric Authentication**

I. Motivation

Biometric data are more and more used in daily life. However, these data underlie privacy concerns by design, because these data are directly related to individuals. As a result, this may potentially be misused, e.g., by means of replay attacks, once accessible by malicious parties. Therefore, biometric data require protection mechanisms to take advantage of positive aspects of an authentication scheme. So, privacy-preserving biometric authentication is a requirement that comes into focus of databases, which form the core of any biometric system. In [1], the original conference article that is extended in this paper, we present a new approach for user authentication based on the assumption that encrypted data have to be stored and at the same time there is no logging information available.

Although data might be deleted from a database, it is possible to restore the information partly or even completely. Grebhahn et al. [2] present an approach for deleting data in a database whereas at the same time information could be completely recovered. Although new approaches exist to cover this information or even to improve the system for secure deletion [3], an overall security of traditional database management systems with respect to such information leakage cannot be guaranteed.

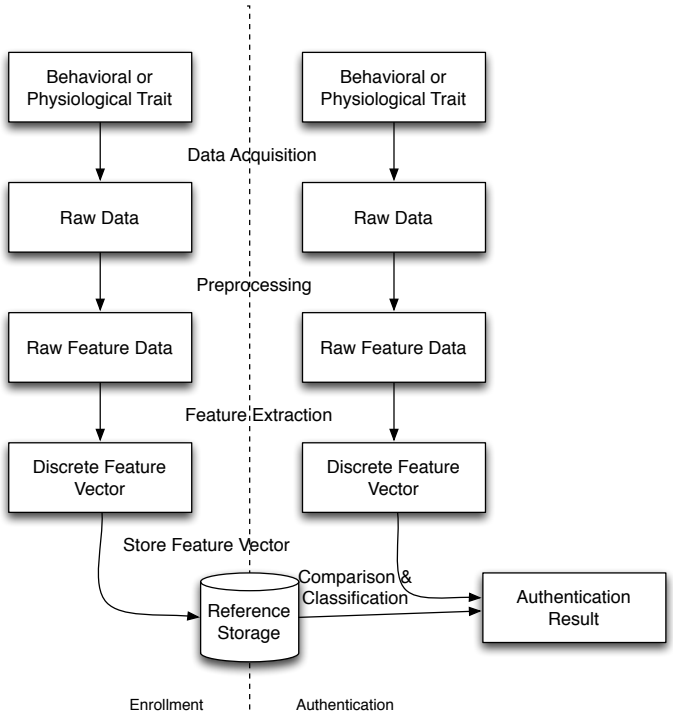

Figure 1. Enrollment and Authentication Pipeline

In a biometric authentication system, two phases are differentiated [4]. Firstly, a user has to create a specific biometric template. In practice, these templates are typically stored in a database. In order to store only required information, the data acquisition (e.g., by using sensors) is followed by a data preprocessing to filter out noise and non-related information of the raw data. Note that required information is often depicted in a feature space. Secondly, a feature extraction is applied, which is followed by a discretization of the feature values. Finally, the feature vector is stored. This phase is called **enrollment**. We show the basic steps in Figure 1 on the left side.

The second phase is called **authentication**, where a classification is required to declare an identity of the biometric features. We depict this pipeline on the right side of Figure 1. The first steps from data acquisition to the discrete feature vector should be applied in the same manners as in the enrollment phase. Otherwise, it cannot be guaranteed that the same properties are compared. However, the data for authentication are not stored. In the comparison step, if a one-to-one matching is performed, we call the authentication **verification** [4]. Another classification schema is **identification**, where a biometric discrete feature vector is compared to a set of templates from the database. In both schemes, usually a threshold is used to decide on the success of the authentication.

In case the threshold does not influence the comparison of templates, the result set of an identification can be the closest match, all, k-nearest, or ϵ-distance-neighbors. With these result-sets, further analyzes are possible, e.g., data mining or forensic investigations. Due to complexity, there are several optimization approaches possible. For instance, it is possible to use index structures within the database system for an enhanced data access. However, such index structures need to be carefully optimized for a multi-dimensional feature space, see for further details [5]. Another approach is to preserve privacy in the context of deletion in database index structures as described in [3].

Data mining enables users to detect patterns that are hidden in complex data. With the use of computational techniques, it is also possible to observe and identify relations in the context of privacy preserving scenarios, see for instance [6], [7], or [8].

The work presented in this paper is based on the paper [1] and extends the work as well as summarizes the main results. We present a methodology based on the Paillier cryptosystem [9] to improve user preferences with respect to authentication systems.

We present a cross-evaluation of the impact of homomorphic encryption for biometric authentication using a database within our evaluation section. The Paillier system is an asymmetric cryptographic scheme with additive homomorphic properties. With our new approach, both unique identifiers in our scheme (UID and FID, see Figure 5) need to be decrypted for every message. A disclosure of either the key is more unlikely, user-tracing becomes less likely, and the pad do not immediately reveal user content data.

The remainder of this paper is structured as follows: In Section II, we briefly describe the current state of the art regarding our new approach. In Section III, we present the architectural requirements for the application scenario of multi-computer involvement. Our extension of the secure similarity verification is given in Section IV. The evaluation of our approach regarding performance is part in Section V, where we show that response times are accompanied with a small computational effort for privacy preserving aspects. These findings are in line with theoretical considerations and assumptions. Finally, we conclude our results and give a short outlook in Section VI.

II. Background and Related Work

In this section, we present related work for preserving privacy in a biometric authentication context. As important factors, we concentrate on homomorphic encryption as well as deletion in database systems. The reason to focus here on homomorphic encryption instead of any other alternative cryptographic concept (see, e.g., [10]) is that this concept allows neglecting the crucial question of key provisioning.

With the majority of the established cryptographic schemes, the client either has to disclose a key to the database system (DBS) or, if such a disclosure is not allowed, has to perform the cryptographic functions itself. Both alternatives result in the transfer or registration of sensitive data items (either the keys or the data itself). With homomorphic encryption this is not necessary, because certain operations on the encrypted data can be performed by the DBS without possession of keys (see [10] and [11] for details).

Data security requirements target at properties of a system to protect data in a sufficient way. The main properties regarding data security are [12]:

- **Confidentiality** addresses the secrecy or prevention of unauthorized resources disclosure. In most practical cases, it refers to information, which needs to be treated secret from unauthorized entities.
- **Authenticity** is divided into two distinct aspects: Data origin authenticity and entity authenticity. Data origin authenticity is the proof of the data origin, genuineness, originality, truth, and realness. Entity authenticity is the proof that an entity has been correctly identified as originator, sender or receiver; it can be ensured that an entity is the one it claims to be.
- **Integrity** is the quality or condition of data objects being whole and unaltered, and it refers to their consistency, accuracy, and correctness.
- Given a set of entities and a resource, the resource has the property of **availability** if all entities of the set can successfully use the resource.
- **Non-repudiation** proves involved and third parties whether or not a particular event or a particular action occurred. The event or action can be, e.g., generation or sending of a message, receipt of a message, and submission or transport of a message.

The general security requirements for a biometric authentication system are summarized in [13]. Here, it is shown that all security aspects summarized in [12] become relevant for all enrollment and verification/identification related components as well as all data transitions between these. Privacy issues are mainly related to confidentiality, but require integrity, authenticity, availability, and non-repudiation of privacy related data. For each security aspect, a security level can also be introduced, e.g., ranging from non, low, up to high.

Within the domain of biometric authentication, data signals are often erroneous. The data are error prone due to noise within the acquisition process. This is the reason, why fault tolerance has to be carefully respected, too. Currently, only the One-Time-Pad approach can be considered as information-theoretically secure as long as the key is distributed securely.

Security plays a vital role due to different scenarios, in which an attack of personal data is imaginable. A differentiation of attacks can be made on a first level regarding passive or active attacks. The data stream between sender and recipient is not influenced in passive attacks. Therefore, only the reading of data is target for such attacks. Besides just reading data, a specialization is frequency analysis, where for instance for a substitution cipher an analysis of letter frequency is used to identify a mapping. Different extensions are applicable, e.g., frequency attacks or domain attacks [14].

Data mining and big data enable a high variety of data analytic techniques. In our biometric scenario, we hide user information to avoid a user tracking. However, it is possible to identify users with the help of log-files [15] or use pattern identification to even track anonymized users [16]. Furthermore, it is not necessary to to use as much information as possible, because a reduction of the multi-dimensional data spaces also reveals good patterns and deliver interpretable models [17].

In the concept of database performance, it has been shown that procedural extensions of modern database systems, such as Oracle PL/SQL and PostgreSQL PL/pgSQL, are very well suited for integrating cryptographic and steganographic functionality, e.g., [18]. This comes with a minimal performance overhead [19]. However, the authors have also shown that this approach comes with a large implementation and testing overhead. To this end, we consider this integration into databases as main focus.

In the following subsections, we first present background on the issue of template protection and secure deletion in databases, we also look into homomorphic encryption, which is followed by efficient biometric comparison in the encrypted domain.

A. Biometric Template Protection

A first idea to preserve privacy in the domain of biometric authentication is to store no direct data corresponding to personal information. In such a scenario, the templates are exchanged with a one-way hash function that is applied on the feature vectors.

As a result of the noise characteristics of biometric signals mentioned above, no cryptographic hash function can be applied directly for this task. Instead biometric hashes (or BioHashes, see [4]) are used. Those algorithms generate hash objects and supporting data required for the quantization operations used to stabilize the biometric input.

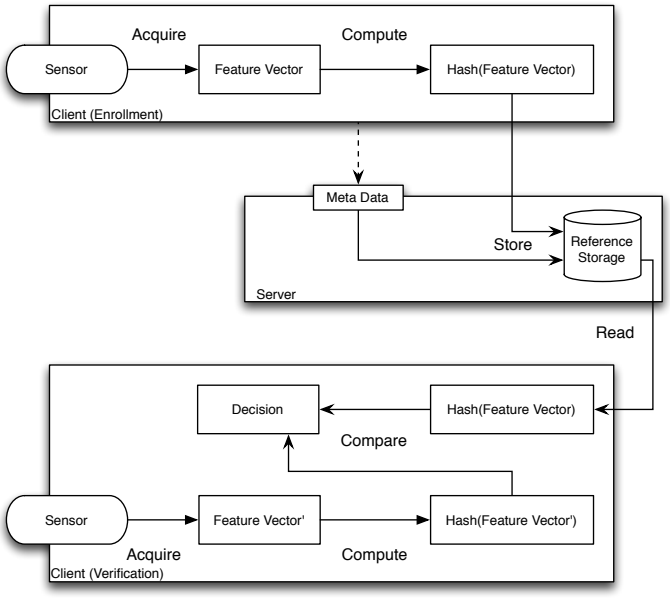

Figure 2. A Biometric Template Protection Example

In Figure 2, we present this approach for the verification of a user. In the database, only the hash values and user specific information such as the required interval matrix are stored. Meta data is also stored, to guarantee that future constructions perform in a comparable manner. The verification takes place at the client side, where the new biometric feature vector is handled as the comparable one with the same hash function. From the database the stored hash value and the corresponding user specific data are obtained and used for comparison.

Protection mechanisms for such Biometric reference systems exist since more than a decade; prominent examples are BioHashes [4], Fuzzy Commitment Scheme [20], and Fuzzy Vault [21]. For an overview on challenges for biometric template protection and further current protection schemes see [22]. All these established protection schemes require data to be compared in an unencrypted form, which leads to the threat of information leakage as discussed in Section I. Therefore, these mechanisms are not relevant for the work presented in this paper.

B. Secure Deletion in Databases

Databases can often reveal more information than intended. If an entry is deleted from the data collection, it is a mandatory step to avoid the data reconstruction afterward. Stahlberg et al. [23] and Grebhahn et al. [2] explain how data can be reconstructed from metadata or system copies. Furthermore, DBS specific data, such as index structures, can also be used for reconstruction of deleted data. This means, even if no data are left, the system inherent data structure can be used

to gain information from fully deleted data tuples. Therefore, privacy awareness for database tunings, as described in [3], is required for biometric DBS to guarantee data privacy, which is especially challenging for multi-dimensional data [24].

Apart from a possible reconstruction of previously erased data, saved data can reveal additional information. For instance, the amount of queries for a data tuple can give an idea about who that tuple belongs to. This kind of vulnerabilities of the confidentiality needs to be addressed early at the stage of the database layout. Not all security risks can be solved at this stage of the design, but a good database layout can indeed be the foundation of a secure system.

Our proposal here is to solve a prominent part of these confidentiality issues by not storing plain text items to the DBS and using homomorphic encryption to solve the key provisioning issue (i.e., the need for the system to have access to crypto keys).

C. Homomorphic Encryption

Homomorphic encryption is used to perform for asymmetric encryption schemes data operations on the cipher text, which have a corresponding operation on plain text data. In homomorphic encryption, operations op^* can be performed on encrypted data that are equivalent to operations op on the plain text. This means that the following formula holds:

$$op(x) = decryption\left(op^*(encryption\left(x\right)\right). \quad (1)$$

In such a case, the mapping is structure preserving. The operations op and op^* depend on the cryptosystem. There exist additive and multiplicative homomorphic cryptosystems. Gentry [25] proves the existence of a fully homomorphic encryption scheme having additive as well as multiplicative properties. So, it is possible to perform certain operations on data without possessing a decryption key. However, such systems require high computational effort as well as a translation of required operations on the functions provided by the homomorphic encryption scheme at hand (here the Paillier scheme [9]. In this paper, we make use of homomorphic encryption to perform operations for authentication in an encrypted domain. The basic idea for our work is derived from the work of Rane et al. as summarized in the next subsection.

D. Verification of Homomorphic Encrypted Signals

Rane et al. [11] [26] developed an authentication scheme with adjustable fault tolerance. This is especially important for noisy sensor data. Due to error correction and similarity verification, Rane's method can be applied for a wide range of biometric traits.

In their application, three participants are involved for a multi-computer scenario. Whereas the first user provides the biometric signals, the second involved user acts as the central storage server for all biometric templates. The third user is responsible for verification. However, this user is seen as vulnerable and therefore, she is not allowed to query the database system (DBS). Despite the fact that we also use this three participant setup for our evaluations, we present alternative scenarios in Section III.

III. Architecture for Privacy-Preserving Authentication

In a general authentication setup, there are two instances that have to share information with each other. There is a participant using a sensor to authenticate a claimed identity on the one side. On the other side, there is a reference DBS containing all enrolled data of all registered users. The DBS is considered to be semi-trustworthy, which means the data in this system shall never be available to the database holder without any kind of restriction or encryption. For that reason, a system allowing database authentication without revealing any information to the database holder needs to be applied. Furthermore, it has to be impossible to decrypt data without having the secret key. The solution used in this paper to address this issue is the use of homomorphic encryption.

Here, we use the Paillier crypto system as described in [9]. We slightly extend this scheme with the inclusion of user-definable key lengths for the purpose of the performance evaluations presented in Section V.

In Figure 3, we present a simplified pipeline of a verification process. For this paper, we consider this process layout as a standard pipeline. Note, in this scenario, a compromised DBS administrator could keep track of the order of enrolled employees and therefore, a sequential ID has to be avoided. This is also conceivable for timestamps and other metadata. So, it is inevitable to disable any logging of enrollment steps.

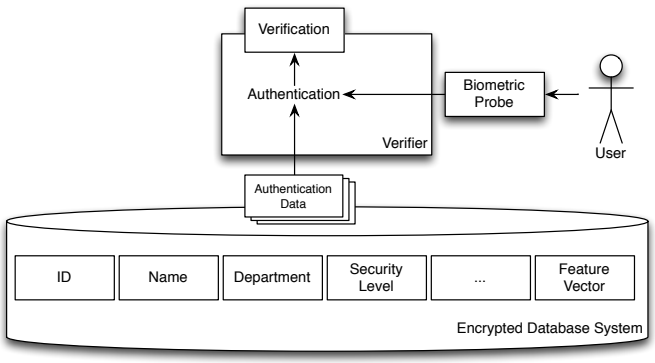

Figure 3. Authentication Process with Encrypted Database, adapted from [27]

In general, there are three major approaches with a different number of participants to be considered for a setup. First, there is a setup consisting of **two participants**. A participant holds sensor and private key and a non-trusted domain holding the data that are encrypted with the public key. The data can never be decrypted on the server side and therefore, need to be sent to sensor side for authentication. The two participants approach requires a secure layer (prospective two+ participants approach) to become trustworthy. This layer would be able to perform black box operations without revealing any information to the database holder or the user.

The second approach is the **three participants** approach from [11], [26], which is summarized in Figure 3. This approach consults a third member, called the verifier, which is

deemed semi-trustworthy as well. The new member shall gain as little information as possible. For that reason, the Paillier cryptosystem [9] is used for the instantiation of this approach within this paper.

The last major setup considered here is a multi-party setup consisting of at least **four participants**. A major advantage is the possible performance boost, since there can be more than one server that handles the computational effort, which is possibly very high, especially when using a key length of 2,048 bits or more. An obvious disadvantage is that more members need to be entrusted with private data. Even though data always remain encrypted, there are vulnerabilities nonetheless. For example, a corrupt administrator could try to track the amount of successful authentication attempts for each enrolled sample and use this information and their domain knowledge to match the samples to actual persons.

The multi-party setup allows every member to be in the setup more than once, which can be of interest for locally distributed systems. For example, a verifier appears multiple times and so, the database-holding participant, which implies that data can be saved either redundantly or distinctly. If, in a decentralized biometric access system, the servers keep their data distinct, every verifier has to keep on searching on the next server until every server has been checked or the data collection has been found. The case in which data are saved redundantly implies that there are as many participants possessing the whole data collection as there are servers. This does not only result in a performance boost. Each copy of the data collection adds a potential corrupt database holder, but makes it harder to keep track users. Furthermore, if a user has to be removed from the system, every trace has to be deleted, too. This is due to prevent reconstruction or information leakage, see also Section II.

A forensically secure deletion, see for instance [2], [3], becomes more complicated the more copies exist, especially if they are distributed on different servers. There can be multiple sensors in the system. It is obviously insecure, if all of them have access to the secret keys. Only if necessary security requirements are met and if the client is fully trusted, the access to the secret key can be granted. Actually, a sensor does not need the secret key to authenticate or enroll a user. Only when it comes to obtaining further information, for example the biometric sample itself as plain, the secret key is required.

Current approaches enable data mining techniques for user tracking. An adequate consideration of multi-participants is another open challenge for authentication. In the next section, we present our padding approach for an enhanced security similarity verification.

IV. Extending Secure Similarity Verification

There exist many biometric authentication systems, which use quite different biometric modalities. Another aspect in this domain is the quality of systems with respect to accuracy and security. To some extent, both properties rely on the trait itself. So, a system that uses only a small set of features with low quality is expected to have overlapping features for different users, which means an encryption of the same value with the same public key.

Due to the fact that systems often have more than one server and are using different key pairs, user tracking is not possible. Additionally, the order of users can be mixed within different systems. We introduce the padded biometrics approach, which allows user authentications in a multiple participant scenario with respect to privacy-preservation. Additionally, we present performance impacts and a brief security impact discussion.

A. The Padded Biometrics Approach

In Figure 4, we depict a scenario for user tracking with two database systems (DBS). We assume, an attacker has read access to both databases. The differences between both DBS are key pairs and user IDs. Assume, with some knowledge, the attacker identifies in DBS_1 User 1. The DBS uses an unsalted asymmetric encryption, which results for a given key and plain text value always in the same cipher value. Within DBS_1, the attacker finds the exact same value for another user (User 5). With the help of this knowledge, both users can be identified in DBS_2, see User 11 and User 31 in Figure 4. Due to the fact that the feature vectors are not shuffled, the attacker needs to identify a match between two users in DBS_2 with an overlap of the same two features.

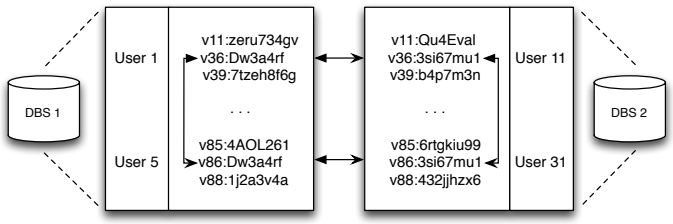

Figure 4. User Tracking in a Multiple DBS Setup, adapted from [27]

In practice, for a proper biometric trait with an appropriate resolution this scenario is implausible. As an example, we take the iris codes with 2,048 bit representation for the iris features; there exist theoretically more than 10^{74} different codes. However, the Euclidean vector space is very sparsely populated due to cluster of iris codes. Such clustering occurs in many biometric modalities. Therefore, our example, given in Figure 4, is a result from exact matches for different feature vectors. Correlations of biometric features are the main reason for such clusters. For instance, [28] examines different approaches in spatial domain iris data.

Daugman [29] identifies the iris phase code to be 0 or 1. This results in a Hamming distance with a very small variance. Daugman uses 249 different features and obtains $\mu = 0.499$ and $\sigma = 0.0317$. There exist several other analogous examples, e.g., in face recognition for the distribution of eyes, nose, and mouth that are quite similar for every person. We conclude that it is very likely that the data in the feature space are not equally distributed.

With these insights or domain knowledge, it is possible to link users or even track users as in our example in Figure 4. An

inclusion of the metadata of the database also enables further possibilities for an information gain, e.g., in the case that an index structure relates similar values, as the R-tree [30] or the Pyramid technique [31].

We propose a padding approach. This is comparable to salting [32]. In Figure 5, we show the idea. Every user receives a specific ID (UID). This ID is encrypted together with the template, e.g., by concatenating ID and biometric feature. This approach also allows including the feature index (FID) in the pad, which avoids intra-user overlapping.

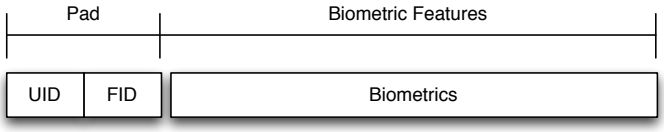

Figure 5. Lead-Pad for Biometric Features, adapted from [27]

The resulting value of a pad and a biometric feature has to be encrypted. A leading pad avoids any inter- and intra-user redundancies. At the same time, the possibility of the above described attack is close to zero. The padding, seen as a security layer, can be either maintained by the user or operated by an additional participant who has paddings and IDs.

This proposal comes at the cost that identification is expected to be more difficult. The pad shifts features semantically away from others. Therefore, the Euclidean measurements for similarity cannot be used, but the complete set of pads for each person has to be processed. We concentrate on performance of our proposed approach in the following.

B. Performance of the Padding Approach in the DBS

Index methods are widely used in DBMS to increase performance [33]. In relational databases, the B-tree [34] [35] and variants, such as the B+-tree, are used to achieve a logarithmic lookup performance. A similarity search using B+-trees results on average in a linear performance overhead additionally. Including a verifier, as proposed in an encrypted data domain, influences the processing time due to transportation effort. We discuss pros and cons in the following.

TABLE I. Performance in a Database System and compared to the Padding Approach

Query Type	DBS with B+-tree	Padding DBS
Exact Match	$O(log(n))$	$O(n)$
Similarity Search	$O(n)$	$O(n)$

Sorting and the use of metadata, which can improve query response times, should be avoided for security reasons. This requirement is in contrast to typically used index structures in relational data management systems. Therefore, the identification within the authentication process requires linear computational effort. Depending on the size and the application scenario, different metadata, such as gender, can be utilized to limit this effort. Note, if small subsets can be created from this metadata, it is necessary to separate these from biometrics. Alternatively, the padding approach can be applied to non-biometrics, too. In Table I, we summarize the computational efforts for a relational database and also for a database with encryption using our padding approach. Due to several other possible performance impacts, such as database size, feature size, thresholds, or key bit-length, we present in Section V a short evaluation study.

1) Implementation Issues: We propose to use a distance result from the verifier instead of a binary decision of acceptance or decline of an authentication attempt. Besides a reasonable attack scenario, where learning from accepted authentications and repeated authentication queries is possible in the later scenario, this risk can be reduced by disabling repeated authentication. In our approach, the quality of the similarity can be computed in an evaluation step. We apply the following formula:

$$d\left(X,Y\right) = \frac{\sum_{i=1}^{dim} |x_i - y_i|^a}{\tau^a \cdot dim} \tag{2}$$

with threshold τ, $a \geq 1$ as degree of freedom, and dim as dimensionality of the feature vector. These parameters are important for adjusting quality regarding sensor accuracy, error rates, and the biometric trait. The better the quality, the lower can be τ and the larger a.

We use a dictionary to maintain all pads for all enrolled users. The pads are delivered via a secure channel for each authentication process. The pads are concatenated before encryption. Due to the non-existence of relations to personal data, the pads can be generated randomly. The necessary step before enrollment or authentication is adding the pad. Note, it is not necessary to add the pad before the signal. Within an identification process, it is necessary to lookup the dictionary for the pad of a user. If outsourcing the dictionary to an external server, a processing time increase has to be respected.

In the following, we consider the three participant approach, for other system architectures from Section III. We measure the influence of computation time regarding all three involved participants. Note, if participants are embedded, as described in Section III, special security requirements have to be met.

The Three Participants scenario, as the default scheme considered in this paper, consists of a user, a verifier, and the DBS, which maintains the encrypted templates. Biometrics are taken by a sensor at user side. The verifier is responsible for authentication. Note, communication channels can be realized in different ways, such as insecure or with encryption. In the case, that only the user stores all pads to corresponding IDs, verifier and DBS do not need to be fully trusted. Hill-climbing should be avoided and therefore, a repeated authentication from single users has to be disabled. As a result, we can sum up that applying our approach to this scenario, only the user and partly the server gain information on the claimed identity. The ability to learn from the results can only be realized on user or verifier side. There is no plain information, due to encryption at user side within the complete process.

For the alternative scenarios discussed in Section III, we briefly present the alternatives in the following.

Two, respectively Two+ Participants: This system is comparable to the above described system. The difference is that the DBS is embedded at user side via a secure layer. We assume, users can never immediately access data by themselves, but only via strict protocols. Therefore, authentication is scheduled by the verifier. Whereas the embedding requires one participant less, secure embedding and sand-boxing of the DBS is necessary.

Three+ Participants (as an extension to the Three Participants scenario): Analogous to the Two+ Participants scenario, a secure layer is introduced. This connects user side and disguise. This requires either centralization or synchronization again. In this scenario, user side gains full information on IDs, but the DBS gains less information on their users. A trace of users is not possible due to random queries.

Four Participants: although the verifier learns in the three participants scenario the results from matching, this information is not important. However, to further decrease such a risk, a participant for disguising claimed identities is introduced. The function of this participant is to reduce the information gain for all participants. Therefore, the disguise blurs requested IDs by fake queries in undetermined intervals. It could also use a dictionary to reduce information gain at user side. However, the user has to learn a name or pseudo-identity to realize identity claiming. With a disguise participant neither the user nor the server can gain full information on claimed identities, but the disguise can learn this information. Verifier and user can learn from the results of the authentication.

In the following section, we show some impacts of our approach regarding security issues.

C. Security Impacts

Our experiments in this paper are separated into two groups: The experiments in the first group examine the security of the authentication system the second group focuses on performance issues. The objective of the security related experiments is to spot when and where data leakage can occur. Especially the changes to the original system proposed by Rane et al. in [11], [26] are of interest, because they affect the security of the system. An entire system analysis is not possible and a cryptanalysis would exceed the scope of this paper. Instead, a selection of conceivable attacks on the system is investigated. In these investigations the setup that causes additional risks for the confidentiality and privacy of the system are detailed.

Since many sensitive data sets are kept in DBS, this is a promising attack vector to gain information. A careful design, see Section IV-B1 and a proper security concept are mandatory. Implementation can cause vulnerabilities to the protocol that can lead to information leakage. There are some attacks, which do not immediately address the protocol. For instance, there are attacks on availability and the endpoint should be carefully considered. An attacker can try to take advantage of vulnerabilities that originated from poor system design. For example, a system designer decides to embed the verifier at user side, but does not meet all steps to guarantee confidentiality. If an unauthorized user is able to listen to the verifier, an information leakage occurs.

In the case the padding approach is implemented inappropriate, e.g., without secure separation from unauthorized users, and an attacker gains access to the pads, the confidentiality is at risk. With access to the pads and the encrypted signal, known-plain-text attacks [36] are possible.

Assume, the system design consists of a traditional DBS. This results in multiple instances of a dataset. Even though, all datasets are marked as deleted, it must be guaranteed on all DBS that there are no cached tables or backups available. So, every additional DBS requires a check that no information is left that can be used to recreate the data set. MySQL, for instance, only marks data with a certain bit, if data are deleted. The data are available in data slacks until they are overwritten. Furthermore, new data do not have the same length. If old data have not been overwritten by, for instance, NULL values, parts of old data can still remain. Accordingly, the data must be erased manually. Grebhahn [2] discusses this example in more detail.

The asymmetric Paillier cryptosystem as well as our padding approach are not information-theoretically secure compared to the symmetric approach, e.g., One-Time-Pad approach.

Thus, there are threats, such as the known-plain-text attack [36], leading to leakage of the biometric templates in the DBS. We introduce a padding approach to avoid opportunity of such attacks. Note, a secure dictionary is mandatory. The implementation of a system can enable various security vulnerabilities. These enable an attacker to gain trusted information. It is mandatory to implement a proper pseudonymization approach in combination with a secure dictionary.

The configuration of a system is presumably the most promising path for an attacker. The DBS amount and type of meta-information can be a threat to security. For instance, time stamps and logging information can be used to compromise security. An attacker can match users to datasets and therefore, trace users. So, system designers have to carefully consider meta-information. Additionally, backups play an important role. With access to both, DB and backup, an attacker subtracts users from backup and current state for user tracking.

Acceptance threshold and quality classes influence *false acceptance* and *false rejection rates*. The threshold decides on size of error patterns. There are many additional factors: level of information confidentiality, quality of the signals, access frequency, expectations regarding response times, and combined biometrics.

If the authentication protocol uses web communication, a denial of service attack (DoS) can disturb the protocol from functioning and harms availability. Even without using the web, there are other possible attacks that are not only taking advantage of communication. For instance, using malware to prevent participants from following the protocol is an imaginable attack on availability. Assuming that a biometric authentication scheme applies the Four Participants scenario,

a DoS attack on the disguise would prevent the system's functioning. It is possible to reduce the threat, but impossible to prevent it completely.

Endpoint security is crucial to provide confidentiality, especially if users have access to secret keys. Assuming the secret key is not as easily accessible, an attacker can try to read parts of communications. This includes plain and encrypted data such as pads. Assessing these data, follow-up attacks like known-plain or known cipher text attacks [36] are possible. For a restriction, basic security steps, including anti-virus software and firewalls, should be implemented.

V. Evaluation

In this section, we present evaluation results on performance for our approach. We focus on performance issues regarding our pad approach. Furthermore, we evaluate processing time as performance metric.

For our evaluation, we present experiments regarding different influence factors, such as enrolled users, key length, feature vector dimension, and thresholds. Firstly, we explain the evaluation setting. Secondly, we show results of our performance evaluation with respect to enrolled users, key length, feature dimensions, and threshold by studying with and without-padding approaches and encrypted versus non-encrypted scenarios.

A. Experimental Layout

For our evaluation, we use a MySQL database, version 5.5.27. We restrict our evaluation to a two table layout with index structures as follows:

- *Person*(Name, Security level, Department, ID)
- *Biometrics*(Feature, ID, BID).

Every enrolled person in the system has some attributes, i.e., a name, a security level, and a department. These attributes can be exchanged or extended by any property. In addition, every person has an ID to find a data tuple unambiguously. All properties like name, security level and department are encrypted with the public key. Biometrics are divided by the count of dimensions of the Euclidean vector. Every feature is identified by a biometric ID (BID), while biometrics are assigned to the corresponding person by an ID.

We make the following assumptions: The DBS is designed that it can be used for most common discrete biometric features. The resolution or the quality of the feature has no influence on the operative readiness of the biometric system itself. How accurate the resolution has to be is a question of acceptable error rates and needs to be adjusted by the corresponding use case. A forensic comparison of found biometrics on a crime scene, for example, needs to be well adjusted and requires a high quality of signals, while an attendance check must not be as accurate. Features are saved in feature vectors and have a minimum of at least one dimension and can have as many dimensions as needed. Everything that depends on the dimension of the feature vector grows corresponding to its size. For example, the codebooks are depending on the size of the feature vector.

B. Performance Evaluation

We perform all experiments on an AMD Phenom II X6 1055T Processor, an SSD, and 8GB RAM. In our evaluation, we focus on response time as crucial performance factor. We apply 10 replications per evaluation run for validity. We use artificial data that we $i.i.d.$ generated from Gaussian distribution. Note, there might be different parameters or measures. However, for simplicity, we exclude more complex influence parameters, such as skewness or correlation within our data. This does not simplifies our evaluations, but enables an easier identification of impacts.

First, we test for size of enrolled users. Note, for simplicity, the feature length is eleven dimensions, the key size is 64bit, and the threshold is set to three.

TABLE II. Performance for Users

Users	Database Processing Time per Users in ms	Identification in ms	Verification in ms
20	17	35	87
1,000	26	63	107
100,000	105	354	354

In Table II, we present arithmetic means for identification and verification for our padding approach. Our results indicate that the overall processing increases with a higher amount of enrolled users. This growth seems linear compared to the database processing time per users. With an increase of the database size, the processing time increases, too. For a sound comparison, we use a standardization of processing time per user. Memory management and thread scheduling or configuration and running the DBS cause this increase. Since verification only requires data of one person, the increase is not similar to identification. Due to B+-trees in MySQL, there is an increasing impact according to the size of enrolled users.

In Table III, we present results regarding key length. Note, we use 1,000 enrolled users in the DBS and a feature dimensionality of 11. As expected, an exponential growth with an increase of the key length is obvious. Due to our experimental setup (using one machine for all tasks), this growth might be influenced in our experimental setup. However, using a private key only increases the processing time in a small amount. A fast feedback is a user requirement for user acceptance of biometric authentication.

TABLE III. Performance for Key Length

Key Length	Identification in ms	Verification in ms
64	63	107
128	112	87
512	1001	400
1,024	6933	2188

We test different feature vector sizes (11, 69, 100, 250, and 2,048) and present the results in Table IV. Adding new features to the feature vectors requires more comparisons, which result in higher response times. Note, with an increase of the feature

TABLE IV. FEATURE DIMENSIONS PERFORMANCE

Feature Dimensions	Identification in ms	Verification in ms
11	63	56
69	239	81
100	354	321
250	693	571
2,048	1,065	860

vector the codebooks also increase. Due to this, the growth in smaller feature vectors can be explained.

As a last evaluation parameter, we vary the threshold from 3 to 1,000 and present our results in Table V. The threshold parameter is used for quality reasons, see also Section IV.

Compared to [11], increasing the threshold by 1 means that two additional comparisons have to be computed. Therefore, the increase is linear with the number of enrolled users. Signals with a higher fluctuation, which require a larger range of validity, require more processing time. This has to be examined for each application and evaluated regarding hardware, requirements, and accuracy.

Nevertheless, our experiment results show this linear relation in both settings, identification and verification. Note, the verification is slightly faster than the identification, which is comparable to the used feature dimensions.

TABLE V. PERFORMANCE FOR THRESHOLD

Threshold	Identification in ms	Verification in ms
3	125	99
5	199	104
10	216	114
100	280	208
1,000	1,572	1,311

As a concluding remark, we present our evaluation results regarding our approach compared to the approach presented in [11], which is implemented without a salting scheme. Note once again, salting increases privacy.

TABLE VI. PERFORMANCE FOR THE PADDING APPROACH

System Parameters	Padding Approach in ms	Without Pad in ms
1,000 users, 2,048 features, 64 bit	25,546	26,014
1,000 users, 11 features, 1,024 bit	28,033	27,197
100,000 users, 11 features, 64 bit	35,009	35,403

In Table VI, we show three different parameter scenarios exemplary. This table shows unexpected results. In the first and third experiment, the response times for the padding approach are slightly lower than without padding. This might be a result from caching and optimizations that take place in the experiments. Note, we conducted the experiment ten times to average execution overhead fluctuations. However, our results show that the influences of our approach are negligible.

In the last setting, we show differences between encrypted and unencrypted identification in Table VII. We use again a key length of 64bit and 11 feature dimensions. The threshold is set to 3. The results show the cost for encryption. Note, we only use a very small computation effort regarding encryption due to a very short key length. With an increase of the key length the difference for both scenarios increases dramatically.

TABLE VII. COMPARISON OF SECURE IDENTIFICATION

Enrolled Users	Encrypted Identification	Unencrypted Identification
20	35 ms	26 ms
1,000	63 ms	47 ms
100,000	354 ms	310 ms

Summarizing our evaluation, we state that a privacy-preserving encryption strategy is not only possible, but the processing overhead is acceptable. We provide a solution that is applicable to existing database systems. Furthermore, we show the impact on performance, which has to be considered in applying this approach.

VI. SUMMARY AND OUTLOOK

In this paper, we present an extension to the secure and similarity verification between homomorphically encrypted signals by Rane [11], [26]. Tracing users is possible in the original scenario. We present a padding approach, to overcome this challenge. We extend the original contribution to search on encrypted values and to use a padding concept. Furthermore, we develop a evaluation study of our conceptual design to evaluate our approach.

With the padding approach, an advanced search in an encrypted domain is possible. However, if repeated authentication attempts are possible, it is already possible to gain information regarding the template. One can avoid such template reproduction by disabling repeated authentications. Our approach improves data security. We name some security requirements for this purpose, but many attack scenarios are getting unlikely if information separation as well as signal processing in the encrypted domain are applied.

Processing times in our evaluation reveal that our padding approach comes at very low additional cost compared to [11]. This is an important aspect for user acceptance of such a system. Whereas the size of enrolled users has logarithmic impact on computational effort, the key length impacts with an exponential scheme. The dimensions of the feature vector have logarithmic influence as well and the threshold is linear in the computational effort. All these parameters do not drastically influence the system of Rane [11]. Due to simple operations, such as summation and amount computation, computational overhead is negligible. However, the concept of privacy-preserving authentication, discussed in this paper, has a strong influence on computational effort compared to plain-text biometric authentication systems.

In future work, our approach can be adapted for other domains that fulfill the same requirements on operations that need to be performed in the encrypted domain. We propose to semantically shift data to complicate unauthorized decryption

attempts, which makes user tracing via duplicate identification unlikely. Particularly, this becomes important, if the co-domain of the biometric feature is smaller than the co-domain of the key. The approach presented in [37] verifies users in the encrypted domain. It is imaginable that the extensions are of interest, too, for this approach, which bases on the homomorphic cryptosystem RSA.

ACKNOWLEDGMENTS

This work is partly based on the master thesis by Martin Leuckert [27]. We thank Martin Schäler for fruitful discussions on an earlier version of this paper. The work in this paper has been funded in part by the German Federal Ministry of Education and Research (BMBF) through the Research Program "DigiDak+ Sicherheits-Forschungskolleg Digitale Formspuren" under Contract No. FKZ: 13N10816 and 13N10818.

The work presented in this paper is based on the paper [1], which was presented at SECURWARE 2014. It contains original text from it with extensions to related work and architectural concepts.

REFERENCES

[1] J. Dittmann, V. Köppen, C. Krätzer, M. Leuckert, G. Saake, and C. Vielhauer, "Performance impacts in database privacy-preserving biometric authentication," in SECURWARE 2014: The Eighth International Conference on Emerging Security Information, Systems and Technologies, R. Falk and C. B. Westphall, Eds. IARA, 2014, pp. 111–117.

[2] A. Grebhahn, M. Schäler, and V. Köppen, "Secure deletion: Towards tailor-made privacy in database systems," in BTW-Workshops. Köllen-Verlag, 2013, pp. 99–113.

[3] A. Grebhahn, M. Schäler, V. Köppen, and G. Saake, "Privacy-aware multidimensional indexing," in BTW. Köllen-Verlag, 2013, pp. 133–147.

[4] C. Vielhauer, Biometric User Authentication for IT Security, ser. Advances in Information Security. Springer, 2006, no. 18.

[5] M. Schäler, A. Grebhahn, R. Schröter, S. Schulze, V. Köppen, and G. Saake, "QuEval: Beyond high-dimensional indexing à la carte," PVLDB, vol. 6, no. 14, 2013, pp. 1654–1665.

[6] F. Emekci, O. Sahin, D. Agrawal, and A. E. Abbadi, "Privacy preserving decision tree learning over multiple parties," DKE, vol. 63, no. 2, 2007, pp. 348 – 361.

[7] A. Inan, Y. Saygyn, E. Savas, A. Hintoglu, and A. Levi, "Privacy preserving clustering on horizontally partitioned data," in Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on, 2006, pp. 95–95.

[8] D. Shah and S. Zhong, "Two methods for privacy preserving data mining with malicious participants," Information Sciences, vol. 177, no. 23, 2007, pp. 5468–5483.

[9] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in EUROCRYPT, ser. Lecture Notes in Computer Science, J. Stern, Ed., vol. 1592. Springer, 1999, pp. 223–238.

[10] M. Schott, C. Vielhauer, and C. Krätzer, "Using different encryption schemes for secure deletion while supporting queries," in Datenbanksysteme für Business, Technologie und Web (BTW 2015) Workshopband, ser. Lecture Notes in Informatics, no. 242, Hamburg, 2015, pp. 37–45.

[11] S. Rane, W. Sun, and A. Vetro, "Secure similarity verification between homomorphically encrypted signals," US Patent US8 249 250 B2, Sep. 30, 2012.

[12] S. Kiltz, A. Lang, and J. Dittmann, "Taxonomy for computer security incidents," in Cyber Warfare and Cyber Terrorism. IGI Global, 2008, pp. 412–417.

[13] C. Vielhauer, J. Dittmann, and S. Katzenbeisser, "Design aspects of secure biometric systems and biometrics in the encrypted domain," in Security and Privacy in Biometrics, P. Campisi, Ed. Springer, 2013, pp. 25–43.

[14] S. Hildenbrand, D. Kossmann, T. Sanamrad, C. Binnig, F. Faerber, and J. Woehler, "Query processing on encrypted data in the cloud," Systems Group, Department of Computer Science, ETH Zurich, Tech. Rep., 2011.

[15] S. T. Peddinti and N. Saxena, "On the effectiveness of anonymizing networks for web search privacy," in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 483–489.

[16] D. Herrmann, C. Banse, and H. Federrath, "Behavior-based tracking: Exploiting characteristic patterns inDNS traffic," Computers & Security, vol. 39, Part A, no. 0, 2013, pp. 17 – 33, 27th {IFIP} International Information Security Conference.

[17] V. Köppen, M. Hildebrandt, and M. Schäler, "On performance optimization potentials regarding data classification in forensics," in Datenbanksysteme für Business, Technologie und Web (BTW 2015) Workshopband, ser. Lecture Notes in Informatics, no. 242, Hamburg, 2015, pp. 21–35.

[18] M. Schäler, S. Schulze, R. Merkel, G. Saake, and J. Dittmann, "Reliable provenance information for multimedia data using invertible fragile watermarks," in 28th British National Conference on Databases (BNCOD), ser. LNCS, vol. 7051. Springer, 2011, pp. 3–17.

[19] M. Schäler, "Minimal-invasive provenance integration into data-intensive systems," Ph.D. dissertation, Otto-von-Guericke-University, Magdeburg, Germany, DEC 2014.

[20] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in 6th ACM Conference on Computer and Communications Security. New York, NY, USA: ACM, 1999, pp. 28–36.

[21] A. Juels and M. Sudan, "A fuzzy vault scheme," Designs, Codes and Cryptography, vol. 38, no. 2, 2006, pp. 237–257.

[22] A. K. Jain, A. Ross, and U. Uludag, "Biometric template security: Challenges and solutions," in In Proceedings of European Signal Processing Conference, 2005.

[23] P. Stahlberg, G. Miklau, and B. N. Levine, "Threats to privacy in the forensic analysis of database systems," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 91–102.

[24] A. Grebhahn, D. Broneske, M. Schäler, R. Schröter, V. Köppen, and G. Saake, "Challenges in finding an appropriate multi-dimensional index structure with respect to specific use cases," in Proceedings of the 24th GI-Workshop "Grundlagen von Datenbanken 2012", I. Schmitt, S. Saretz, and M. Zierenberg, Eds. CEUR-WS, 2012, pp. 77–82, urn:nbn:de:0074-850-4.

[25] C. Gentry, "Computing arbitrary functions of encrypted data," Commun. ACM, vol. 53, no. 3, 2010, pp. 97–105.

[26] S. Rane, W. Sun, and A. Vetro, "Secure similarity verification between encrypted signals," US Patent US20 100 246 812 A1, Sep. 30, 2010.

[27] M. Leuckert, "Evaluation and extension of secure similarity verification in multi-computer scenarios to sesecure store and communicate biometric data," Master's thesis, Otto-von-Guericke University, 2013.

[28] M. Negin, T. A. Chmielewski, M. Salganicoff, T. A. Camus, U. M. C. von Seelen, P. L. Venetianer, and G. G. Zhang, "An iris biometric system for public and personal use," Computer, vol. 33, no. 2, 2000, pp. 70–75.

[29] J. Daugman, "How iris recognition works," IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, no. 1, 2004, pp. 21–30.

[30] A. Guttman, "R-trees: A dynamic index structure for spatial searching," SIGMOD Rec., vol. 14, no. 2, 1984, pp. 47–57.

[31] S. Berchtold, C. Böhm, and H.-P. Kriegel, "The Pyramid-technique: Towards breaking the curse of dimensionality," SIGMOD Rec., vol. 27, no. 2, 1998, pp. 142–153.

[32] R. Morris and K. Thompson, "Password security: A case history," Commun. ACM, vol. 22, no. 11, Nov. 1979, pp. 594–597.

[33] V. Köppen, M. Schäler, and R. Schröter, "Toward variability management to tailor high dimensional index implementations," in RCIS. IEEE, 2014, pp. 452–457.

[34] R. Bayer and E. McCreight, "Organization and maintenance of large ordered indexes," Acta Informatica, vol. 1, 1972, pp. 173–189.

[35] D. Comer, "The Ubiquitous B-Tree," ACM Comput. Surv., vol. 11, no. 2, 1979, pp. 121–137.

[36] B. Schneier, Secrets & Lies: Digital Security in a Networked World. New York, NY, USA: John Wiley & Sons, Inc., 2000.

[37] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Efficient biometric verification in encrypted domain," in 3rd International Conference on Advances in Biometrics, 2009, pp. 899–908.