*Research Article*

# Anonymous Biometric Access Control

**Shuiming Ye, Ying Luo, Jian Zhao, and Sen-Ching S. Cheung**

*Center for Visualization and Virtual Environments, University of Kentucky, Lexington, KY 40507-1464, USA*

Correspondence should be addressed to Sen-Ching S. Cheung, cheung@engr.uky.edu

Access control systems using the latest biometric technologies can offer a higher level of security than conventional password-based systems. Their widespread deployments, however, can severely undermine individuals' rights of privacy. Biometric signals are immutable and can be exploited to associate individuals' identities to sensitive personal records across disparate databases. In this paper, we propose the Anonymous Biometric Access Control (ABAC) system to protect user anonymity. The ABAC system uses novel Homomorphic Encryption (HE) based protocols to verify membership of a user without knowing his/her true identity. To make HE-based protocols scalable to large biometric databases, we propose the $k$-Anonymous Quantization (kAQ) framework that provides an effective and secure tradeoff of privacy and complexity. kAQ limits server's knowledge of the user to $k$ maximally dissimilar candidates in the database, where k controls the amount of complexity-privacy tradeoff. kAQ is realized by a constant-time table lookup to identity the $k$ candidates followed by a HE-based matching protocol applied only on these candidates. The maximal dissimilarity protects privacy by destroying any similarity patterns among the returned candidates. Experimental results on iris biometrics demonstrate the validity of our framework and illustrate a practical implementation of an anonymous biometric system.

## 1. Introduction

In the last thirty years, advances in computing technologies have brought dramatic improvements in collecting, storing, and sharing personal information among government agencies and private sectors. At the same time, new forms of privacy invasion begin to enter the public consciousness. From sale of personal information to identity theft, from credit card fraud to YouTube surrendering user data [1], the number of ways that our privacy can be violated increases rapidly.

One important area of growing concern is the protection of sensitive information in various access control systems. Access control in a distributed client-server system can generally be implemented by requesting digital credentials of the user wanting to access the system. Credentials are composed of attributes that contain identifiable information about a given user. Such information can be very sensitive and uncontrolled disclosure of such attributes can result in many forms of privacy breaches. It is unsurprising that privacy protection has been a central concern in widespread deployment of access control systems, especially in many of the e-commerce applications [2].

Among the different types of access control systems, Biometric Access Control (BAC) systems pose the most direct threat to privacy. BAC systems control allocation of resources based on highlydiscriminative physical characteristics of the user such as fingerprints, iris images, voice patterns, or even DNA sequences. As a biometric signal is based on "who you are" rather than "what you have," BAC systems excel in authenticating a user's identity. While the use of biometrics enhances system security and alleviates users from carrying identity cards or remembering passwords, it creates a conundrum for privacy advocates as the knowledge of the identity makes it much harder to keep users anonymous. A curious system operator or a parasitic hacker can infer the identity of a user based on his/her biometric probe. Furthermore, as biometrics is immutable from systems to systems, it can be used by attackers to cross-correlate disparate databases and cause damages far beyond the coverage of any protection schemes for individual database systems.

A moment of thought reveals that many access control systems do not need the true identity of the user but simply require a confirmation that the user is a legitimate member. For example, an online movie vendor may have a category of "VIP" members who pay a flat monthly membership fee and can enjoy an unlimited number of movies download. While it is important to verify the VIP status of a candidate user, it is unnecessary to precisely identify who the user is. In fact, it will be appeasing to customers if the vendor can provide a guarantee that it can never track their movie selections. Entry control of a large office building that hosts many companies can also benefit from such an anonymous access control system. While it is essential to restrict entry only to authorized personnel, individual companies may be reluctant to turn over sensitive identity information to the building management. Thus a system that can validate the tenant status of a person entering the building without knowing the true identity will be valuable. Another example is a community electronic message board. Only the members of the community can sign in to the system. Once their member status are verified, they can anonymously post messages and complaints to the entire community. All the aforementioned examples can benefit from an access control system that can verify the membership status using biometric signals while keeping the identity anonymous.

In this paper, we introduce Anonymous Biometric Access Control (ABAC) to provide anonymity and access control in such a way that the system server (Bob) can authenticate the membership status of a user (Alice) but cannot differentiate Alice from any other authorized users in his database. Our scheme differs from other work in privacy protection of biometric systems which focus primarily on the security of the biometric data from improper access. Our goal is to guarantee user's anonymity while providing the safeguard of the system resources similar to other access control systems.

In this paper, we consider two technical challenges in developing an ABAC system. First, to cope with the variability of the input probe, any biometric access system needs to perform a signal matching process between the probe and all the records in the database. The challenge here lies in making the process secure so that Bob can confirm the membership status of Alice without knowing any additional information about Alice's probe. We cast this process as a secure multiparty computation problem and develop a novel protocol based on homomorphic encryption. Such a procedure prevents Bob from extracting any knowledge about Alice's probe and its similarity distances with any records in Bob's database. On the other hand, Bob can compare the distances to a similarity threshold in the encrypted domain and the comparison results are aggregated into two secret numbers shared between Bob and Alice. The secret share held by Bob prevents Alice from cheating and Alice's membership status can be verified by Bob without knowing her identity.

Second, we consider the complexity challenge posed by scaling the matching process in encrypted domain to large databases. The high complexity of cryptographic primitives is often cited as the major obstacle of their widespread deployment in realistic systems. This is particularly true for biometric applications that require matching a large number of high-dimensional feature vectors in real time. In this paper, we propose a novel framework to provide a controllable trade-off between privacy and complexity. We call the framework $k$-anonymous ABAC system ($k$-ABAC) which keeps Alice anonymous from $k$, rather than the entire database of, authorized members in the database. This is similar to the well-known $k$-anonymity model [3] in that $k$ is a controllable parameter of anonymity. However, the two approaches are fundamentally different—the $k$-anonymity model is a data disclosure protocol where Bob anonymizes the database for public release by grouping all the data into $k$-member clusters. In a $k$-ABAC system, the goal is to *prevent Bob from obtaining information about the similarity relationship between his data and the query probe from Alice*. In order to minimize the knowledge revealed by any $k$-member cluster, we propose a novel grouping scheme called $k$-Anonymous Quantization (kAQ) that optimizes the *dissimilarity* among members in the same group. kAQ forbids similar patterns to be in the same group which might be a result of multiple registrations of the same person or from family members with similar biometric features. The kAQ process is carried out mostly in plaintext and is computationally efficient. Using kAQ as a preprocessing step, the subsequent encrypted-domain matching can be efficiently realized within the real-time constraint.

The rest of the paper is organized as follows. After reviewing related work in Section 2, we provide the necessary background in the security models for anonymous biometric matching, homomorphic encryption, and dimension reduction in Section 3. We first provide an overview of the entire system in Section 4. The design of ABAC using homomorphic encryption is presented in Section 5. In Section 6, we introduce the concepts of kABAC and $k$-Anonymous Quantization. We also describe a greedy algorithm to realize kAQ and show a secure procedure to perform quantization without revealing private information. To demonstrate the viability of our approach, we have tested our system using a large collection of iris patterns. The details of the experiments and the results are presented in Section 7. We conclude the paper and discuss future work in Section 8.

## 2. Related Work

The main contributions of our paper are the introduction of the ABAC system concept and a practical design of such a system using iris biometrics. There are other work that deal with the privacy and security issues in biometric systems but their focus is different from this paper. A privacy-protecting technology called "Cancelable Biometrics" has been proposed in [4]. To protect the security of the raw biometric signals, a cancelable biometric system distorts a biometric signal using a specially designed noninvertible transform so that similarity comparison can still be performed after distortion. Biometric Encryption (BE) described in [5] possesses all the functionality of Cancelable Biometrics, and is immune against the substitution attack because it outputs a key which is securely bound to a biometric. The BE

templates stored in the gallery have been shown to protect both the biometrics themselves and the keys. The stored BE template is also called "helper data". "Helper data" is also used in [6] to assist in aligning a probe with the template that is available only in the transformed domain and does not reveal any information about the fingerprint.

All the above technologies focus on the security and privacy of the biometric signals in the gallery. Instead of storing the original biometric signal, they keep only the transformed and noninvertible feature or helper data extracted from the original signal that do not compromise the security of the system even if they are stolen. In these systems, the identity of the user is always recognized by the system after the biometric matching is performed. To the best of our knowledge, there are no other biometric access systems that can provide access control and yet keep the user anonymous. Though our focus is on user anonymity, our design is complementary to cancelable biometrics and it is conceivable to combine features from both types of systems to achieve both data security and user anonymity.

Anonymity in biometric features like faces is considered in [7]. Face images are obfuscated by a face deidentification algorithm in such a way that any face recognition softwares will not be able to reliably recognize deidentified faces. The model used in [7] is the celebrated $k$-anonymity model which states that any pattern matching algorithm cannot differentiate an entry in a large dataset from at least $k - 1$ other entries [3, 8]. The $k$-anonymity model is designed for data disclosure protocols and cannot be used for biometric matching for a number of reasons. First, despite the goal of keeping the user anonymous, it is very important of an ABAC system to verify that a user is indeed in the system. Face de-identification techniques provide no guarantee that only faces in the original database will match the de-identified ones. As such, an imposter may gain access by sending an image that is close to an de-identified face. Second, de-identification techniques group similar faces together to facilitate the public disclosure of the data. This is detrimental to anonymity as face clusters may reveal important identity traits like skin color, facial structure, and so forth.

Another key difference between anonymity in data disclosure and biometric matching is the need for secure collaboration between two parties—the biometric server and the user. The formal study of such a problem is Secure Multiparty Computation (SMC). SMC is one of the most active research areas in cryptography and has wide applications in electronic voting, online bidding, keyword search, and anonymous routing. While there are no previous work that use SMC for biometric matching, many of the basic components in a BAC system can be made secure under this paradigm. They include inner product [9, 10], polynomial evaluation [11–13], thresholding [14–16], median [17], matrix computation [18, 19], logical manipulation [20], $k$-means clustering [21, 22], decision tree [23–25] and other classifiers [12, 26–28]. A recent tutorial in SMC for signal processing community can be found in [29].

The main hurdle in applying computationally-secure SMC protocols to biometric matching is their high computational complexity. For example, the classical solution to the thresholding problem (this problem is commonly referred to as the Secure Millionaire Problem in SMC literature), or comparing two private numbers $a$ and $b$, is to use Oblivious Transfer (OT) [30]. OT is an SMC protocol for joint table lookup. The privacy of the function is guaranteed by having the entire table encrypted by a precomputed set of public keys and transmitted to the other party. The privacy of the selection of the table entry is protected based on obfuscating the correct public key among the dummy ones. Even with recent advances in reducing the computational and communication complexity [13, 17, 31–34], the large table size, the intensive encryption, and decryption operations render OT difficult for pixel or sample-level signal processing operations.

A faster but less general approach is to use Homomorphic Encryption (HE) which preserves certain operations in the encrypted domain [35]. Recently, the homomorphic encryption scheme is proposed by IBM and Stanford researcher C. Gentry has generated a great deal of excitement in using HE for encrypted domain processing [36]. He proposed using Ideal Lattices to develop a homomorphic encryption system that can preserve both addition and multiplication operations. This solves an open problem on whether there exists a semanticallysecure homomorphic encryption system that can preserve both addition and multiplication. On the other hand, his construction is based on protecting the simplest boolean circuit and its generalization to realistic application is questionable. In an interview, Gentry estimates that performing a Google search with encrypted keywords would increase the amount of computing time by about a trillion [37] and even this claim is already challenged by others to be too conservative [38].

More practical homomorphic encryptions such as Paillier cryptosystem can only support addition between two encrypted numbers, but do so over a much larger additive plaintext group, thus providing a wide dynamic range for computation [39]. Furthermore, as illustrated in Section 3, multiplication between encrypted numbers can be accomplished by randomization and interaction between parties. Recently, Paillier encryption is being applied in a number of fundamental signal processing building blocks [40] including basic classifiers [27] and Discrete Cosine Transform [41] in encrypted domain. Nevertheless, the public-key encryption and decryption processes in any homomorphic encryption still pose a formidable complexity hurdle to overcome. For example, the fastest thresholding result takes around 5 seconds to compare two 32-bit numbers using a modified Paillier encryption system with a key size of 1024 bits [14]. One of the goals of this paper is to utilize homomorphic encryption to construct a realistic biometric matching system that can tradeoff computation complexity with user anonymity in a provably secure fashion.

## 3. Background

We model any biometric signal $\mathbf{x} = (x_1, \ldots, x_n)^T$ as an $n$-dimensional vector from a feature space $F^n$ where $F$ is a finite field. We also assume the existence of a commutative

distance function $d : F^n \times F^n \to \mathfrak{R}^+ \cup \{0\}$ that measures the dissimilarity between two biometric signals. In order for the distance to be computable using the operators in the field, we assume $F$ to be a subfield of $\mathfrak{R}$ so that the components of the constituent vectors will be treated as real numbers in the distance computation. The most commonly used distance is the Euclidean distance:

$$d(\mathbf{x}, \mathbf{y})^2 := \|\mathbf{x} - \mathbf{y}\|_2^2 = \sum_{i=1}^{n} (x_i - y_i)^2. \qquad (1)$$

For the iris patterns used in our experiments, $F$ is the binary field $Z_2 = \{0, 1\}$ and $d(\cdot, \cdot)$ is a modified hamming distance defined below [42]:

$$d_H(\mathbf{x}, \mathbf{y})^2$$

$$:= \frac{\left\|(\mathbf{x} \otimes \mathbf{y}) \cap \mathrm{mask_x} \cap \mathrm{mask_y}\right\|_2^2 \left\|\mathrm{mask_x} \cap \mathrm{mask_y}\right\|_2^2}{\left\|\mathrm{mask_x} \cap \mathrm{mask_y}\right\|_2^2},$$

$$(2)$$

where $\otimes$ denotes the XOR operation and $\cap$ denote the bitwise AND. $\mathrm{mask_x}$ and $\mathrm{mask_y}$ are the corresponding mask binary vectors that mask the unusable portion of the irises due to occlusion by eyelids and eyelash, specular reflections, boundary artifacts of lenses, or poor signal-to-noise ratio. As the mask has substantial variation even among feature vectors captured from the same eye, we assume that the mask vectors do not disclose any identity information.

The special distance function and the high dimension of many feature spaces make them less amenable to statistical analysis. There exist mapping functions that can project the feature space $F^n$ into a lower-dimensional space $\mathfrak{R}^m$ such that the original distance can be approximated by the distance, usually Euclidean, in $\mathfrak{R}^m$. The most well-known technique is Principal Component Analysis (PCA) which is optimal if the original distance is Euclidean [43]. For general distances, mapping functions can be derived by two different approaches—the first approach is Multidimensional Scaling (MDS) in which an optimal mapping is derived based on minimizing the differences between the two distances over a finite dataset [44]. The second approach is based on distance relationship with random sets of points and include techniques such as Fastmap [45], Lipschitz Embedding [46], and Local Sensitivity Hashing [47]. In our system, we use both PCA and Fastmap for their low computational complexity and good performance. Here we provide a brief review of the Fastmap procedure and will discuss its secure implementation in Section 6. Fastmap is an iterative procedure in which each step selects two random pivot objects $\mathbf{x}_A$ and $\mathbf{x}_B$ and computes the projection $x'$ for any data point $\mathbf{x}$ as follows:

$$x' := \frac{d(\mathbf{x}, \mathbf{x}_A)^2 + d(\mathbf{x}_A, \mathbf{x}_B)^2 - d(\mathbf{x}, \mathbf{x}_B)^2}{2d(\mathbf{x}_A, \mathbf{x}_B)}. \qquad (3)$$

The projection in (3) requires only distance relationships. A new distance is then computed by taking into account the existing projection:

$$d'(\mathbf{x}, \mathbf{y})^2 := d(\mathbf{x}, \mathbf{y})^2 - (x' - y')^2, \qquad (4)$$

where $x'$ and $y'$ are the projections of $\mathbf{x}$ and $\mathbf{y}$, respectively. The same procedure can now be repeated using the new distance $d'(\cdot, \cdot)$. It has been demonstrated in [45] that using pivot objects that are far apart, the Euclidean distance in the projected space produces a reasonable approximation of the original distance of many different feature spaces.

Using a dissimilarity metric, we can now define the function of a biometric access control system. It is a computational process that involves two parties: a biometric server (Bob) and a user (Alice). Bob is assumed to have a database of $M$ biometric signals DB $= \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, where $\mathbf{x}_i = (x_1^i, \ldots, x_n^i)^T$ is the biometric signal of member $i$. Alice provides a probe $\mathbf{q}$ and requests access from the server. Armed with these notations, we first provide a functional definition of a Biometric Access Control system.

*Definition 3.1.* A *Biometric Access Control* (BAC) system is a computational protocol between two parties, Bob with a biometric database DB and Alice with a probe $\mathbf{q}$, such that at the end of the protocol, Alice and Bob can jointly compute the following value:

$$y_{\mathrm{BAC}} := \begin{cases} 1, & \text{if } d(\mathbf{q}, \mathbf{x_i})^2 < \varepsilon \text{ for some } \mathbf{x_i} \in \mathrm{DB} \\ 0, & \text{otherwise.} \end{cases} \qquad (5)$$

Adding user anonymity to a BAC system results in the following definition:

*Definition 3.2.* An *Anonymous* BAC (ABAC) system is a BAC system on DB and $\mathbf{q}$ with the following properties at the end of the protocol.

(1) Except for the value $y_{\mathrm{BAC}}$, Bob has negligible knowledge about $\mathbf{q}$, $d(\mathbf{q}, \mathbf{x})$, and the comparison results between $d(\mathbf{q}, \mathbf{x})^2$ and $\epsilon$ for all $\mathbf{x} \in \mathrm{DB}$.

(2) Except for the value $y_{\mathrm{BAC}}$, Alice has negligible knowledge about $\epsilon$, $\mathbf{x}$, $d(\mathbf{q}, \mathbf{x})$, and the comparison results between $d(\mathbf{q}, \mathbf{x})^2$ and $\epsilon$ for all $\mathbf{x} \in \mathrm{DB}$.

Like any other computationally secure protocols, "negligible knowledge" used in the above definition should be interpreted as, given the available information to a party, the distribution of all possible values of the private input from the other party is computationally indistinguishable from the uniformly random distribution [48]. The first property in Definition 3.2 defines the concept of user anonymity, that is, Bob knows nothing about Alice except whether her probe matches one or more biometric signals in DB. As it has been demonstrated that even the distance values $d(\mathbf{q}, \mathbf{x}_i)$ are sufficient for an attacker to recreate DB [49], the second property is designed to disclose the least amount of information to Alice.

It is impossible to design a secure system without considering the possible adversarial behaviors from both parties. Adversarial behaviors are broadly classified into two types: semihonest and malicious. A dishonest party is called semihonest if he follows the protocol faithfully but attempts to find out about others' private data through the communication. A malicious party, on the other hand, will change

private inputs or even disrupt the protocol by premature termination. Making the proposed system robust against a wide range of malicious behaviors is beyond the scope of this paper. Here, we assume Bob to be semihonest but allow certain malicious behaviors from Alice—we assume that Alice will engage in malicious behaviors only if those behaviors can increase her chance of gaining access, that is turning $y_{BAC}$ into 1, from using a purely random probe. This is a restricted model because, for example, Alice will not prematurely terminate before Bob reaches the final step in computing $y_{BAC}$. Also, Alice will not randomly modify any private input unless such modification will increase her chance of success.

In Section 5, we shall provide an implementation of an ABAC system on iris biometrics that is robust under the above security model. The procedure is based on repeated use of a homomorphic encryption system. An encryption system $Enc(x)$ is homomorphic with respect to an operation $f_1(\cdot, \cdot)$ in the plaintext domain if there exists another operator $f_2(\cdot, \cdot)$ in the ciphertext domain such that

$$Enc(f_1(x, y)) = f_2(Enc(x), Enc(y)). \tag{6}$$

In our system, we choose the Paillier encryption system as it is homomorphic over a large additive plaitext group and thus providing a wide dynamic range for computation. Given a plaintext number $x \in Z_N$, the Paillier encryption process is given as follows:

$$Enc_{pk}(x) = \left[(1 + N)^x \cdot r^N \bmod N^2\right], \tag{7}$$

where $N$ is a product of two equal-length secret primes and $r$ is a random number in $Z_N$ to ensure semantic security. The public key $pk$ consists of only $N$. The decryption function $Dec_{sk}(c)$ with $c \in Z_{N^2}$ and the secret key $sk$ being the Euler-phi function $\phi(N)$ is defined by the following two steps:

(1) Compute $\hat{m} = [(c^{\phi(N)} \bmod N^2) - 1]/N$ over the integer field;

(2) $Dec_{sk}(c) = \hat{m} \cdot \phi(N)^{-1} \bmod N$.

The Paillier system is secure under the decisional composite residuosity assumption and we refer interested readers to [50, Chapter 11], for details. Paillier is homomorphic over addition in $Z_N$ and the corresponding function is multiplication over the ciphertext field $Z_{N^2}$. We can also carry out multiplication with a known plaintext in the encrypted domain. These properties are summarized in:

$$Enc_{pk}(x + y) = Enc_{pk}(x) \cdot Enc_{pk}(y),$$
$$Enc_{pk}(xy) = Enc_{pk}(x)^y. \tag{8}$$

Multiplication with a number to which only the ciphertext is known can also be accomplished with a simple communication protocol. Assume that Bob wants to compute $Enc_{pk}(xy)$ based on the ciphertexts $Enc_{pk}(x)$ and $Enc_{pk}(y)$. Alice has the secret key $sk$ but Bob wants to keep $x$, $y$ and $xy$ hidden from Alice. $MULT(Enc_{pk}(x), Enc_{pk}(y))$ (Protocol 1) is a secure protocol that can accomplish this task. It is secure
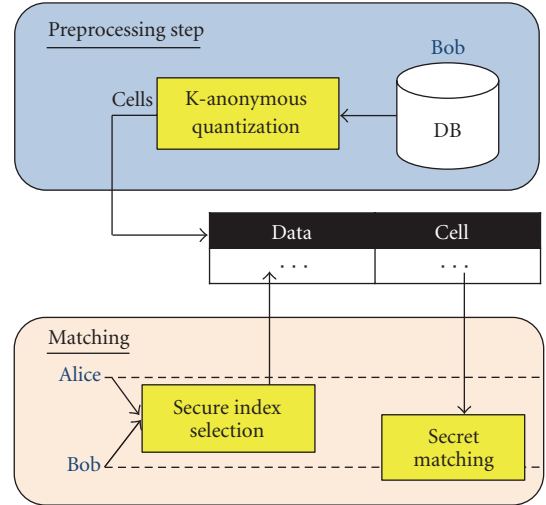


FIGURE 1: ABAC system overview.

because Alice can gain no knowledge about $x$ and $y$ from the uniformly random $x - r$ and $y - s$ where $r$ and $s$ are two random numbers generated by Bob, and Bob is never exposed to any plaintext related to $x$ and $y$. The complexities of $MULT(Enc_{pk}(x), Enc_{pk}(y))$ are three encryptions and seven encrypted-domain operations, (multiplication and exponentiation) on Bob side, as well as two decryptions and one encryption on Alice side. The communication costs are three encrypted numbers. The homomorphic properties and this protocol will be used extensively throughout this paper.

## 4. System Overview

In this section, we provide an overview of the entire design of our efficient anonymous biometric access control system. Again, we will use Bob and Alice to denote the biometric system owner and the user, respectively. The overall framework of our proposed system is shown in Figure 1. There are two main processing components in our systems: the preprocessing step and the matching step. While the matching step is executed for every probe, the preprocessing step is executed only once by Bob to compute a *publiclyavailable* quantization table based on a process called $k$-Anonymous Quantization. The purpose of the public table is that, based on a joint secure-index selection of the table entry between Alice and Bob, Bob can significantly reduce the scope of the similarity search from the entire database DB to approximately $k$ candidates. The $k$-Anonymous Quantization guarantees that (1) if there is an entry in Bob's database that matches Alice's probe, this entry must be among these candidates, (2) all the candidates are maximally dissimilar so as to provide the least amount information about Alice's probe, and (3) the public table discloses no information about Bob's database. The details of the $k$-Anonymous Quantization and the secure-index selection will be discussed in Section 6.

---

**Require**: Bob: $\text{Enc}_{pk}(x)$, $\text{Enc}_{pk}(y)$; Alice: $sk$
**Ensure**: Bob computes $\text{Enc}_{pk}(xy)$
   (1)  Bob sends $\text{Enc}_{pk}(x - r) = \text{Enc}_{pk}(x) \cdot \text{Enc}_{pk}(-r)$ and $\text{Enc}_{pk}(y - s) = \text{Enc}_{pk}(y) \cdot \text{Enc}_{pk}(-s)$
       to Alice where $r$ and $s$ are uniformly random numbers generated by Bob.
   (2)  Alice decrypts $\text{Enc}_{pk}(x - r)$ and $\text{Enc}_{pk}(y - s)$, computes $\text{Enc}_{pk}[(x - r)(y - s)]$ and send
       it to Bob.
   (3)  Bob computes $\text{Enc}_{pk}(xy)$ in the encrypted domain as follows:

$$\begin{aligned} \text{Enc}_{pk}(xy) &= \text{Enc}_{pk}[(x - r)(y - s) + xs + yr - rs] \\ &= \text{Enc}_{pk}[(x - r)(y - s)] \cdot \text{Enc}_{pk}(x)^s \cdot \text{Enc}_{pk}(y)^r \cdot \text{Enc}_{pk}(-rs) \end{aligned}$$

PROTOCOL 1: Private multiplication MULT($\text{Enc}_{pk}(x)$,$\text{Enc}_{pk}(y)$).

---

After computing the proper quantization cell index from the public table, Bob identifies all the candidates and then engages with Alice in a joint secret matching process to determine if Alice's probe resembles any one of the candidates. This process is conducted in a multiparty computation and communication protocol between Alice and Bob based on Paillier homomorphic encryption. We assume that there is an open network between Bob and Alice that will guarantee message integrity. Since only encrypted content is exchanged, there is no need for any protection against eavesdroppers. For each session, Alice will be responsible for generating the private and public keys for the encryption and sharing the public key with Bob. In other words, a different set of keys will be used for each different user. Furthermore this protocol demands comparable computational capabilities from both parties. Thus it is imperative to use the preprocessing step to reduce the computational complexity of this matching step. As the secret matching utilizes all the fundamental processing blocks for the entire system, we will first explain this component in the following section.

## 5. Homomorphic Encryption-Based ABAC

In this section, we describe the implementation of an ABAC system on iris features using homomorphic encryption. The system consists of three main steps: distance computation, bit extraction, and secure comparison. Except for the first step of distance computation which is specific towards iris comparison, the remaining two steps and the overall protocol are general enough for other types of biometric features and similarity search. We shall follow a bottom-up approach by first describing individual components and demonstrating their safety before assembling them together as an ABAC system.

*5.1. Hamming Distance.* The modified Hamming distance $d_H(\mathbf{x}, \mathbf{y})$ described in (2) is used to measure the dissimilarity between iris patterns $\mathbf{x}$ and $\mathbf{y}$ which are both 9600 bits long [51]. As the division in (2) may introduce floating point numbers, we focus on the following distance and roll the denominator into the similarity threshold during the later stage of comparison

$$\widehat{d_H}(\mathbf{x}, \mathbf{y})^2 := \left\| (\mathbf{x} \otimes \mathbf{y}) \cap \text{mask}_{\mathbf{x}} \cap \text{mask}_{\mathbf{y}} \right\|_2^2. \qquad (9)$$

DIST (Protocol 2) provides a secure computation of the modified Hamming distances between Alice's probe q and Bob's DB. Alice needs to provide the encryption of individual bits $\mathbf{q} = (q_1, q_2, \ldots, q_n)^T$ and their negation to Bob. Even though Bob can compute the negation in the encryption domain by performing $\text{Enc}_{pk}(\neg q_i) = \text{Enc}_{pk}(1 - q_i) = \text{Enc}_{pk}(1) \cdot \text{Enc}_{pk}(q_i)^{-1}$, it is computationally more efficient for Alice to compute them in plaintext as demonstrated in Section 7. In step 1(a), Bob computes the XOR between each bit of the query and the corresponding bit in each record $\mathbf{x}_i$. $\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)$ can then be computed by summing all the XOR results in the encrypted domain. Bob cannot derive any information about Alice's probe as the operations are all performed in the encrypted domain. Alice does not participate in this protocol at all. The complexity of DIST includes $O(Mn)$ encrypted-domain operations where $M$ is the size of DB and $n$ is the number of bits for each feature vector.

*5.2. Bit Extraction.* The next step is to compare the calculated encrypted distance with a plaintext threshold. As comparison cannot be expressed in terms of summation and multiplication of the two numbers, we need to first extract individual bits from the encrypted distance. EXTRACT($\text{Enc}_{pk}(x)$) (Protocol 3) is a secure protocol between Bob and Alice to extract individual encrypted bits $\text{Enc}_{pk}(x_k)$ for $k = 1, \ldots, l$ from $\text{Enc}_{pk}(x)$, where $x$ is a $l$-bit number. The idea is for Bob to ask Alice's assistance in decrypting the numbers and extracting the bits. To protect Alice from knowing anything about $x$, Bob sends $\text{Enc}_{pk}(x + r)$ to Alice who then extracts and encrypts individual bits $\text{Enc}_{pk}[(x + r)_k]$. Except for the least significant bit (LSB), Bob cannot undo the randomization in $\text{Enc}_{pk}[(x + r)_k]$ by carrying out an XOR operation with the bits of $r$ due to the carry bits. To rectify this problem, step 2(d) in EXTRACT zeros out the lower-order bits after they have been extracted and stores the intermediate result in $y$, thus guaranteeing the absence of any carry bits from the lower order bits during the randomization. Alice cannot learn any information about $y$ because the bit to be extracted, $(y + r)_k$, is uniformly distributed between 0 and 1. Plaintexts obtained by Alice in different iterations are also uncorrelated as a different random number is used by Bob in each iteration. Even though Alice wants to make $x$ as small as possible to pass the comparison test, there is no advantage

---

**Require**: Bob: $\mathbf{x}_i$ for $i = 1, \ldots, M$, $\mathrm{Enc}_{pk}(q_j)$ and $\mathrm{Enc}_{pk}(\neg q_j)$ for $j = 1, \ldots, n$

**Ensure**: Bob computes $\mathrm{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2]$ for $i = 1, \ldots, M$.

(1) For $i = 1, \ldots, M$, Bob repeats the following two steps:

    (a) For $k = 1, \ldots, n$, compute

$$\mathrm{Enc}_{pk}(q_k \otimes x_k^i) = \begin{cases} \mathrm{Enc}_{pk}(q_k) & \text{if } x_k^i = 0, \\ \mathrm{Enc}_{pk}(\neg q_k) & \text{otherwise} \end{cases}$$

    (b) Compute

$$\mathrm{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2] = \mathrm{Enc}_{pk}\Big(\sum_{k:[\mathrm{mask}_\mathbf{q} \cap \mathrm{mask}_{\mathbf{x}_i}]_i = 1} q_k \otimes x_k^i\Big)$$
$$= \prod_{k:[\mathrm{mask}_\mathbf{q} \cap \mathrm{mask}_{\mathbf{x}_i}]_i = 1} \mathrm{Enc}_{pk}(q_k \otimes x_k^i)$$

Protocol 2: Secure computation of distances DIST(DB, $\mathrm{Enc}_{pk}(\mathbf{q_j})$), $\mathrm{Enc}_{pk}(\mathbf{q_j})$ for $\mathbf{j = 1, \ldots, n}$.

of replacing her replies to Bob with any other value. Bob is not able to obtain any information about $x$ either as all operations are performed in the encrypted domain. Based on the security model introduced in Section 3, this protocol is secure. The complexities of EXTRACT are $l$ encryptions and $O(l)$ encrypted-domain operation for Bob, as well as $l$ decryptions and $l$ encryptions for Alice. The communication costs are $2l$ encrypted numbers.

*5.3. Threshold Comparison.* Based on the encrypted bit representations of the distances, we can carry out the actual threshold comparison. COMPARE($\mathrm{Enc}_{pk}(x_k), y_k$ for $k = 1, \ldots, l$) (Protocol 4) is based on the secure comparison protocol developed in [14]. Step 2(a) accumulates the differences between the two numbers starting from the most significant bits. The state variable $w = 0$ at the $k$th step implies that the bits at order $k$ and higher between $x$ and $y$ match perfectly with each other. Step 2(b) then computes $\mathrm{Enc}_{pk}(c_k)$ where $c_k = 0$ if and only if $w = 0$, $x_k = 0$, and $y_k = 1$. This implies that $x < y$. In other words, $x < y$ is true if and only if there exists $c_k = 0$. In the last step, we invoke the secure multiplication as described in Protocol 1 to combine all $c_k$ together into $c$ which is the desired output. Bob gains no knowledge in this protocol as he never handles any plaintext data. The only step that Alice involves in is in the secure multiplication. The adversarial intention of Alice is to make $c$ zero so as to pass the comparison test. However, the randomization step in Protocol 1 provides no additional knowledge nor advantage for Alice to change her input. Thus, this protocol is secure. The complexities of COMPARE are $3l$ encryptions and $O(l)$ encrypted-domain operations on Bob side, as well as $2l$ decryptions and $l$ encryptions on Alice side. The communication costs are $3l$ encrypted numbers.

*5.4. Overall Algorithm.* Protocol 5 defines the overall ABAC system. Steps 1 and 2 show that Alice first sends Bob her public key and the encrypted bits of her probe. Steps 3 and 4 use secure distance computation DIST (Protocol 2) and secure bit extraction EXTRACT (Protocol 3) to compute the encrypted bit representations of all the distances. Steps 4 and 5 then use secure comparison COMPARE (Protocol 4) and accumulate the results into $\mathrm{Enc}_{pk}(u)$ where $u = 0$ if and

only if $\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2 < \epsilon \cdot \|\mathrm{mask}_\mathbf{q} \cap \mathrm{mask}_{\mathbf{x}_i}\|_2^2$ for some $i$. To determine if Alice's probe produces a match, Bob cannot simply send Alice $\mathrm{Enc}_{pk}(u)$ for decryption as she will simply returns a zero to gain access. Instead, Bob adds a random share $r$ and sends $\mathrm{Enc}_{pk}(u + r)$ to Alice. The decrypted value $u + r$ cannot be sent directly to Bob for him to compute $u$. Unless $u = 0$, the actual value of $u$ should not be disclosed to Bob in plaintext as it may disclose some information about the distance computations. Instead, we assume the existence of a Collision-Resistant Hash Function HASH to which Bob and Alice share the same key $pk_H$ [50, Chapter 4]. Alice and Bob compute $\mathrm{HASH}_{pk_H}(u + r)$ and $\mathrm{HASH}_{pk_H}(r)$, respectively. As the hash function is collision resistant, their equality implies that $u = 0$ and Bob can verify that Alice's probe matches one of the entries in DB without knowing the actual value of the probe. Since Alice knows nothing about $r$, she cannot cheat by sending a fake hash value. The complexities of Protocol 5 are $O(M \log_2 n)$ encryptions and $O(Mn)$ encrypted-domain operations for Bob, as well as $O(M \log_2 n)$ encryptions and decryptions for Alice. The communication costs are $O(M \log_2 n)$ encrypted numbers.

## 6. $k$-Anonymous BAC

In Section 5, we show that both the complexities and the communication costs of the ABAC depend linearly on the size of the database, making ABAC difficult to scale to large databases. Inspired by the $k$-anonymity model, a simple approach is to tradeoff complexity with privacy by quickly narrowing Alice's query into a small group of $k$ candidates and then performing the full cryptographic search only on this small group. $k$ will serve as a parameter to balance between the complexity and the privacy needed by Alice. This is the idea behind the $k$-Anonymous Biometric Access Control system.

*Definition 6.1.* A $k$-*Anonymous* BAC ($k$-ABAC) system is a BAC system on Bob's database DB and Alice's probe $\mathbf{q}$ with the following properties at the end of the protocol.

(1) There exists a subset $S \subset \mathrm{DB}$ with $|S| \geq k$ such that for all $\mathbf{x} \in \mathrm{DB} \setminus S$, Bob knows $d(\mathbf{q}, \mathbf{x})^2 \geq \epsilon$.

**Require**: Bob: $Enc_{pk}(x)$ where $x$ is a $l$-bit number; Alice $sk$.
**Ensure**: Bob computes $Enc_{pk}(x_k)$ for $k = 1,\ldots,l$ with $k = 1$ being the LSB.
  (1) Bob creates a temporary variable $Enc_{pk}(y) := Enc_{pk}(x)$.
  (2) For $k = 1,\ldots,l$, the following steps are repeated
    (a) Bob generates a random number $r$ and sends $Enc_{pk}(y + r)$ to Alice.
    (b) Alice decrypts $y + r$, extracts the $k$th bit $(y + r)_k$ and sends $Enc_{pk}[(y + r)_k]$
      back to Bob.
    (c) Bob computes $Enc_{pk}(x_k) := Enc_{pk}[(y + r)_k \otimes r_k]$.
    (d) Bob updates $Enc_{pk}(y) := Enc_{pk}(y - x_k 2^{k-1}) = Enc_{pk}(y) \cdot Enc_{pk}(x_k)^{-2^{k-1}}$

PROTOCOL 3: Bit extraction EXTRACT($Enc_{pk}(\mathbf{x})$).

**Require** Bob: $Enc_{pk}(x_k), Enc_{pk}(y_k)$ and $y_k$ for $k = 1,\ldots,l$; Alice: $sk$
**Ensure** Bob computes $Enc_{pk}(c)$ such that $c = 0$ if $x < y$.
  (1) Bob sets $Enc_{pk}(c) := Enc_{pk}(1)$, $Enc_{pk}(w) := Enc_{pk}(0)$.
  (2) For $k = l,\ldots,1$ starting from the MSB, Bob and Alice compute
    (a) $Enc_{pk}(w) := Enc_{pk}[w + (x_k \otimes y_k)] = Enc_{pk}(w) \cdot Enc_{pk}(x_k \otimes y_k)$
    (b) $Enc_{pk}(c_k) := Enc_{pk}(x_k - y_k + 1 + w) = Enc_{pk}(x_k) \cdot Enc_{pk}(y_k)^{-1} \cdot Enc_{pk}(1) \cdot$
    $Enc_{pk}(w)$
    (c) $Enc_{pk}(c) := MULT(Enc_{pk}(c), Enc_{pk}(c_k))$.

PROTOCOL 4: Secure comparison COMPARE($Enc_{pk}(\mathbf{x_k}), \mathbf{y_k}$ for $\mathbf{k = 1,\ldots,l}$).

(2) Except for the value $y_{\mathrm{BAC}}$ as defined in Definition 3.1, Bob has negligible knowledge about $\mathbf{q}$ and $d(\mathbf{q}, \mathbf{x})$, for all $\mathbf{x} \in$ DB, as well as the comparison results between $d(\mathbf{q}, \mathbf{x})^2$ and $\epsilon$ for all $\mathbf{x} \in S$.

(3) Except for the value $y_{\mathrm{BAC}}$, Alice has negligible knowledge about $\epsilon$, $\mathbf{x}$, $d(\mathbf{q}, \mathbf{x})$, and the comparison results between $d(\mathbf{q}, \mathbf{x})^2$ and $\epsilon$ for all $\mathbf{x} \in$ DB.

The definition of $k$-ABAC system is similar to that of ABAC except that Bob can prematurely exclude DB $\setminus S$ from the comparison. Even though Alice may be aware of such a narrowing process, the $k$-ABAC has the same restriction on Alice's knowledge about DB as the regular ABAC. There are two challenges in designing a $k$-ABAC system.

(1) How do we find $S$ so that the process will disclose as little information as possible about $\mathbf{q}$ to Bob?

(2) How can Alice choose $S$ that contains the element that is close to $\mathbf{q}$ without learning anything about DB?

Sections 6.1 and 6.2 describe our approaches in solving these problems in the context of iris matching.

*6.1. k-Anonymous Quantization.* A direct consequence of Definition 6.1 is that if there exists an $\mathbf{x} \in$ DB such that $d(\mathbf{q}, \mathbf{x})^2 < \epsilon$, $\mathbf{x}$ must be in $S$. In order to achieve the goal of complexity reduction, our approach is to devise a static quantization scheme of the feature space $F^n$ and publish it in a scrambled form so that Alice can select the right group on her own. To explain this scheme, let us start with the

definition of a $\epsilon$-ball $k$-quantization. Define $B_\epsilon(\mathbf{x})$ or the $\epsilon$-ball of $\mathbf{x}$ to be the smallest subset of $F^n$ that contains all $\mathbf{y} \in F^n$ with $d(\mathbf{y}, \mathbf{x})^2 < \epsilon$. An $\epsilon$-ball $k$-quantization of DB is defined below.

*Definition 6.2.* An *$\epsilon$-ball $k$-quantization* (eBkQ) of DB is a partition $\Gamma = \{P_1,\ldots,P_N\}$ of $F^n$ with the following properties:

(1) $\bigcup_{i=1}^{N} P_i = F^n$ and $P_i \cap P_j = \phi$ for $i \neq j$,

(2) For all $\mathbf{x} \in$ DB, $B_\epsilon(\mathbf{x}) \cap P_j = B_\epsilon(\mathbf{x})$ or $\phi$ for $j = 1,\ldots,N$,

(3) $|\mathrm{DB} \cap P_j| \geq k$ for $j = 1,\ldots,N$.

Property 1 of Definition 6.2 ensures that $\Gamma$ is a partition while property 2 ensures that no $\epsilon$-ball centered at a data point straddles two cells. The last property ensures that each cell must at least contain $k$ elements from DB. The importance of using an eBkQ $\Gamma$ is that if $\Gamma$ is a shared knowledge between Alice and Bob, Alice can select $P_j \ni \mathbf{q}$ and communicate the *cell index* $j$ to Bob. Then Bob can compute $S := \mathrm{DB} \cap P_j$ which must contain, if exists, any $\mathbf{x}$ where $d(\mathbf{q}, \mathbf{x})^2 < \epsilon$.

While a typical vector quantization of DB will satisfy the $\epsilon$-ball preserving criteria, the requirement of preserving the anonymity of $\mathbf{q}$ imposes a very different constraint. Specifically, we would like all the data points in $S$ to be *maximally dissimilar* so that no common traits can be learned from $S$. This leads to our definition of $k$-Anonymous Quantization (kAQ).

---

**Require**: Bob: $\mathbf{x}_i, i = 1, \ldots, M$ and $\epsilon$; Alice: $\mathbf{q}$
  **Ensure** : Bob computes $y = 1$ if $\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2 < \epsilon$ for some $i$ and 0 otherwise
    (1)  Alice sends $pk$ to Bob.
    (2)  Alice computes $\text{Enc}_{pk}(q_j)$ and $\text{Enc}_{pk}(\neg q_j)$ for $j = 1, \ldots, n$ and sends them to Bob.
    (3)  Bob executes $\text{DIST}(\text{DB}, \text{Enc}_{pk}(q_j), \text{Enc}_{pk}(q_j)$ for $j = 1, \ldots, n)$ to obtain
        $\text{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2]$ for $i = 1, \ldots, M$.
    (4)  For $i = 1, \ldots, M$, Bob and Alice execute $\text{EXTRACT}(\text{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)^2])$ to obtain the
        binary representations $\text{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)_k^2]$ for $k = 1, \ldots, \lceil \log_2 n \rceil$.
    (5)  Bob sets $\text{Enc}_{pk}(u) := \text{Enc}_{pk}(1)$.
    (6)  For $i = 1, \ldots, M$, Bob and Alice computes
        (a)  $\text{Enc}_{pk}(c) := \text{COMPARE}(\text{Enc}_{pk}[\widehat{d_H}(\mathbf{q}, \mathbf{x}_i)_k^2], (\epsilon \| \text{mask}_\mathbf{q} \cap \text{mask}_{\mathbf{x}_i} \|_2^2)_k$ for $k =$
            $1, \ldots, \lceil \log_2 n \rceil)$
        (b)  $\text{Enc}_{pk}(u) := \text{MULT}(\text{Enc}_{pk}(u), \text{Enc}_{pk}(c))$.
    (7)  Bob generates a random number $r$, computes $\text{HASH}_{pk_H}(r)$ and sends Alice $\text{Enc}_{pk}(u + r)$.
    (8)  Alice decrypts $\text{Enc}_{pk}(u + r)$, computes $\text{HASH}_{pk_H}(u + r)$ and sends it back to Bob.
    (9)  Bob sets $y = 1$ if $\text{HASH}_{pk_H}(r) = \text{HASH}_{pk_H}(u + r)$ and 0 otherwise.

PROTOCOL 5: ABAC(DB, q).

*Definition 6.3.* An *optimal k-anonymous quantization* $\Gamma^*$ is an eBkQ of DB that maximizes the following utility function among all possible eBkQ $\Gamma$:

$$\min_{P \in \Gamma} \sum_{\mathbf{x}, \mathbf{y} \in P \cap \text{DB}} d(\mathbf{x}, \mathbf{y})^2. \qquad (10)$$

The utility function (10) can be interpreted as the total dissimilarity of the most homogeneous cell $P$ in the partition. The utility function also depends on the number of data points in a cell—adding a new point to an existing cell will always increase its utility. Thus finding the partition that maximizes this utility function not only can ensure the minimal amount of dissimilarity within a cell, but also can promotes equal distribution of data points among different cells. Given a fixed number of cells, it is important to minimize the variation in the number of data points among different cells so that the computational complexities of the encrypted-domain matching in different cells would be comparable.

It is challenging to solve for the optimal kAQ for the iris matching problem due to the high dimension, 9600 to be exact, and the uncommon distance used. Our first step is to project this high-dimensional space into a lower-dimensional Euclidean space $\mathfrak{R}^m$ by using Fastmap followed by PCA. The Fastmap is used to embed the native geometry of the feature space into an Euclidean space while the PCA optimally minimizes the dimension of the resulting space. Even in this lower-dimensional space, the structure of a quantization, namely, the boundary of individual cells, can still be difficult to specify. To approximate the boundary with a compact representation, we first use a simple uniform lattice quantization to partition $\mathfrak{R}^m$ into a rectilinear grid $\Omega$ consisting of $L$ bins $\{B_1, \ldots, B_L\}$. Then, we maximize the utility function (10) but force the cell boundary to be along those of the bins. This turns an optimal partitioning problem in continuous space into a discrete knapsack problem in

assigning bins to cells through a mapping function $f$ to optimize the utility function. The process is described in Figure 2. We denote the resulting approximated $k$-quantization as $\widehat{\Gamma^*}$.

As the utility function (10) is based on individual data points, a bin containing multiple $\epsilon$-balls may present in multiple cells. As such, $\widehat{\Gamma^*}$ is no longer a true partition and the mapping function $f$ is a multivalued function. A probe falling in these "overlapped" bins will invoke multiple cells, resulting in a larger candidate set $S$. Two examples of such overlapped bins are shown in Figure 2. This increases computational complexity and as such, it is important to minimize the amount of overlap. Due to the uneven distribution of data points in the feature space, a global $\epsilon$ can inflate the size of balls in some area of the feature space resulting in significant overlap problems. In our implementation, we do not use $\epsilon$ balls but estimate the local similarity structure by using multiple similar feature vectors from each iris, and creating a "bounding box" which is the smallest rectilinear box along the bin boundaries that encloses all the bins containing these similar feature vectors. If any bin in a bounding box is assigned to cell $i$, all the bins in the bounding box will have an assignment of cell $i$.

Protocol 6 (KAQ) describes a greedy algorithm that computes a suboptimized $k$-anonymous quantization mapping function from the data. Step 1 of KAQ sets the number of cells to be the maximum and the protocol will gradually decrease it until each cell has more than $k$ data points. The initialization steps in 2 and 3 randomly assign a bounding box into each cell. Step 4 identifies the cells that have the minimum utility. Among these cells, steps 5 and 6 identify the cell $P_{i*}$ and the bounding box $BB^*$ which together produce the maximum gain in utility. The bins inside $BB^*$ are then added to $P_{i*}$ and the whole process repeats. This update not only provides a greedy maximization of the overall utility function but also has the tendency to produce an even distribution of data points among different cells. A newly updated cell will have a much lower chance of being
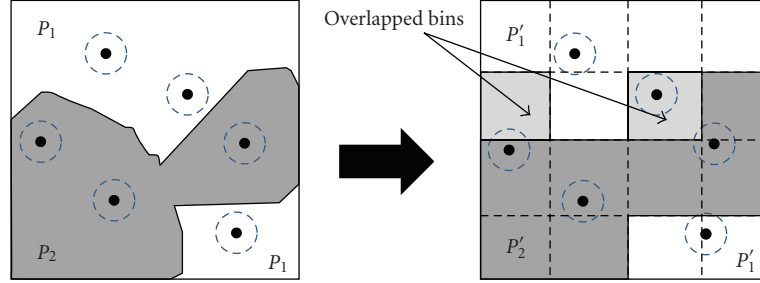
FIGURE 2: Approximation of the quantization boundary (a) along the bins (b). The number of bins $k$ here is 3. There are also two bins that are present in both cells.

updated again as it has a higher utility than others. The final step checks to see if any one cell has less than $k$ elements and, if yes, restarts the process with fewer target number of cells. For a fixed target number of cells, the complexity of this greedy algorithm is $O(M^2)$ where $M$ is the size of DB. It is important to point out that the output mapping $f$ only contains entries of bins that belong to at least one bounding box.

*6.2. Secure Index Selection.* Let us first describe how Alice and Bob can jointly compute the projection of Alice's probe $\mathbf{q}$ into the lower-dimensional space formed by Fastmap and PCA. The projection needs to be performed in encrypted domain so that Alice does not reveal anything about her probe and Bob does not reveal any information about his database, the Fastmap pivot points and the PCA basis vectors. Note that the need for encrypted-domain processing does not affect the scalability of our system as the computation complexity depends only on the dimension of the feature space but not on the size of the database.

The Fastmap projection in (3) involves a floating point division. The typical approach of premultiplying both sides by the divisor to ensure that integer-domain computation does not work. As the Fastmap update (4) needs to square the projection, recursive computation into higher dimensions will lead to a blowup in the dynamic range. To ensure all the computations are performed within in a fixed dynamic range, Alice and Bob need to agree on a predefined scaling factor $\alpha$ and rounding will be performed at each iteration of the Fastmap calculation. Specifically, given the encrypted probe $\text{Enc}_{pk}(\mathbf{q})$, Bob approximates the first projection $q'$ in encrypted domain based on the following formula derived from (3):

$$\alpha\widetilde{q'} := \text{round}\left(\frac{\alpha}{2ad}\right)\widehat{d_H}(\mathbf{q}, \mathbf{x}_A)^2 + \text{round}\left(\frac{\alpha}{2cd}\right)\widehat{d_H}(\mathbf{x}_A, \mathbf{x}_B)^2$$
$$- \text{round}\left(\frac{\alpha}{2bd}\right)\widehat{d_H}(\mathbf{q}, \mathbf{x}_B)^2,$$
$$(11)$$

where $a = \|\text{mask}_{\mathbf{q}} \cap \text{mask}_{\mathbf{x}_A}\|_2^2$, $b = \|\text{mask}_{\mathbf{q}} \cap \text{mask}_{\mathbf{x}_B}\|_2^2$, $c = \|\text{mask}_{\mathbf{x}_A} \cap \text{mask}_{\mathbf{x}_B}\|_2^2$, and $d = d_H(\mathbf{x}_A, \mathbf{x}_B)$. All the multipliers on the right-hand side of (11) are known to Bob in plaintext and the distances can be computed in the encrypted domain

using Procedure 2. Since rounding is involved, $\widetilde{q'}$ is just an approximation of $q'$ as computed with in the original Fastmap formula (3). Based on the computed encrypted values of $aq'$ from the probe and $ax'$ from a data point, the update (4) is executed as follows:

$$\alpha^2\widetilde{d_H'}(\mathbf{x}, \mathbf{q})^2 := \text{round}\left(\frac{\alpha^2}{\|\text{mask}_{\mathbf{x}} \cap \text{mask}_{\mathbf{q}}\|_2^2}\right)\widehat{d_H}(\mathbf{x}, \mathbf{q})^2$$
$$- \left(\alpha\widetilde{x'} - \alpha\widetilde{q'}\right)^2.$$
$$(12)$$

Bob again can compute the right-hand side of (12) entirely in encryption domain, with the square in the second term computed using Procedure 1. The value $\widetilde{d_H'}(\mathbf{x}, \mathbf{q})^2$ is again approximated due to the rounding of the coefficient. Note that the left-hand side has an extra factor of $\alpha$ which needs to be removed so as to prevent a blowup in the dynamic range. To accomplish that, Bob computes $\text{Enc}_{pk}(\alpha^2\widetilde{d_H'}(\mathbf{x}, \mathbf{q})^2 + r\alpha)$ where $r$ is a random number, and sends the result to Alice. Alice decrypts it, divides it by $\alpha$, and rounds it to obtain $\text{round}(\alpha^2\widetilde{d_H'}(\mathbf{x}, \mathbf{q})^2) + r$. Alice encrypts the result and sends it back to Bob who will then remove the random number $r$.

Bob can now use the new distances to project the probe along the second pair of pivot objects $\mathbf{x}_{A'}$ and $\mathbf{y}_{A'}$ as follows:

$$\alpha^2\widetilde{q''} := \text{round}\left(\frac{\alpha}{2d'}\right)\alpha\widetilde{d_H'}(\mathbf{q}, \mathbf{x}_{A'})^2 + \text{round}\left(\frac{\alpha^2}{2}\right)$$
$$- \text{round}\left(\frac{\alpha}{2d'}\right)\alpha\widetilde{d_H'}(\mathbf{q}, \mathbf{x}_{B'})^2,$$
$$(13)$$

where $d' = \widetilde{d_H'}(\mathbf{x}_{A'}, \mathbf{x}_{B'})^2$ can be computed by Bob in plaintext. The extra factor of $\alpha$ on the left-hand side of (13) can be removed with the help of Alice using a similar approach as previously discussed. As the iteration continues, the deviation of the rounded projection and the original projection will grow as the rounding error accumulates. However, the new distance computed at each iteration absorbs the rounding error from the previous projection. As a result, the distance in the projected space will approach the underlying distance in a similar manner as the original projection.

---

**Require** Bob: Projection of DB into $\mathfrak{R}^m$ or $\{P(x_i)$ for $i = 1, \ldots, M\}$; Bin and bounding box structures in $\Omega$;

**Ensure** Bob computes the multi-valued mapping $f : \Omega \rightarrow \{1, \ldots, N\}$ that defines the cell membership of each bin.

(1) Set the initial number of cells $N := \lfloor M/k \rfloor$.

(2) Let $L :=$ the list of bounding boxes in $\Omega$

(3) Random initialization of cells: for $i = 1, \ldots, N$,

    (a) Randomly remove a bounding box $BB$ from $L$.

    (b) Set $f^{-1}(i) := \{$bins in BB$\}$.

(4) Identify the collection of cells $E$ with the lowest utility, that is,

$$E := \arg\min_{i=1,\ldots,N} \sum_{\mathbf{x},\mathbf{y} \in A_i \cap \mathrm{DB}} d(\mathbf{x}, \mathbf{y})^2$$

    where $A_i = \bigcup_{B \in f^{-1}(i)} B$ contains all the bins in cell $i$.

(5) For each cell $j$ in $E$, identify the bounding box $BB_j^* \in L$ that maximizes the utility of cell $j$ after adding $BB_j^*$ to it and denote the resulting utility as $u_j^*$, that is,

$$BB_j^* := \mathrm{argmax}_{BB \in L} \sum_{\mathbf{x},\mathbf{y} \in (A_j \cup BB) \cap \mathrm{DB}} d(\mathbf{x}, \mathbf{y})^2$$

$$u_j^* := \sum_{\mathbf{x},\mathbf{y} \in (A_j \cup BB_j^*) \cap \mathrm{DB}} d(\mathbf{x}, \mathbf{y})^2$$

(6) Given $j^* = \arg\max_{j \in E} u_j^*$, identify the bounding box $BB^* := BB_{j*}^*$ and cell $P_{j*}$ that give rise to the maximum gain of utility from step 5.

(7) Set $f^{-1}(j^*) := f^{-1}(j^*) \cup \{$bins in $BB^*\}$ and remove $BB^*$ from $L$.

(8) Go back to Step 4 until $L$ is empty.

(9) For $i = 1, \ldots, N$, ensure that $|\bigcup_{B \in f^{-1}(i)} B \cap \mathrm{DB}| \geq k$. If not, set $N := N - 1$ and go back to step 2.

PROTOCOL 6: Greedy $k$-anonymous quantization KAQ.

---

In the computation of PCA projection, we scale each basis vector with a large enough multiplier not only to absorb the fractional parts of the basis vector but also the scalar $\alpha$ used in Fastmap. Let the $i$th basis vector of PCA be $\mathbf{p}_i = \eta(p_1^i, p_2^i, \ldots, p_{m_1}^i)^T$ where $i = 1, \ldots, m_2$ with $m_2$ being the target PCA dimension. The encrypted-domain PCA projection of the Fastmap projection of $\mathbf{q}$ can be computed as follows:

$$\mathrm{Enc}_{pk}\left[P_{pca}\left(P_{fm}(\mathbf{q})\right)_i\right] := \mathrm{Enc}_{pk}\left[P_{fm}(\mathbf{q})^T \mathbf{p}_i\right]$$

$$= \mathrm{Enc}_{pk}\left[\sum_{j=1}^{m_1} \alpha P_{fm}(\mathbf{q})_j \frac{\eta p_j^i}{\alpha}\right]$$

$$= \prod_{j=1}^{m_1} \mathrm{Enc}_{pk}\left[\alpha P_{fm}(\mathbf{q})_j\right]^{(\eta p_j^i/\alpha)}$$

$$\approx \prod_{j=1}^{m_1} \mathrm{Enc}_{pk}\left[\alpha P_{fm}(\mathbf{q})_j\right]^{\mathrm{round}(\eta p_j^i/\alpha)}.$$

$$(14)$$

The scalar $\eta$ is selected so that the loss of precision due to rounding is sufficiently small.

The last step of the process is to quantize the projection $P_{pca}(P_{fm}(\mathbf{q}))$. We only consider the quantization step size in powers of two so that the quantization process can be performed in the encrypted domain. First, we use the secure bit extraction routine EXTRACT to compute the binary representation of $\mathrm{Enc}_{pk}[P_{pca}(P_{fm}(\mathbf{q}))]$. Then, we drop the lower order bits based on the chosen stepsize. The resulting bits are recombined to form the binary representation to the encrypted bin index $\mathrm{Enc}_{pk}(B)$.

In order to obtain the cell index $f(B)$, we need an additional cryptographic tool: a homomorphic collision-resistant hash function $h_{PK_h}(\cdot)$ with the following homomorphic property [52, 53]:

$$h_{pk_h}(x + y) = h_{pk_h}(x) \cdot h_{pk_h}(y). \quad (15)$$

Our implementation is based on [52]. Bob generates both the public key $pk_h$ and the secret key for this hash function and shares the public key with Alice. Instead of directly publishing the mapping $f(\cdot)$ between the bin index and the corresponding cell indices, Bob publishes an obfuscated mapping $f'(\cdot)$ such that $f(B) = f'(h_{pk_h}(B))$. The hash function sufficiently scrambles all the bin indices so that the distribution of Bob's data among all the bins classified in the KAQ algorithm is disguised as random sampling in the range of the hash function. To prevent Alice from launching a dictionary attack on the table, the length of the bin index must be large enough. This can be accomplished, for example, by padding random projections of the query to make the bin index longer. The cell indices will be published without any obfuscation—little information is leaked through them as it is shared knowledge between Alice and Bob that there are roughly $N/k$ distinct cell indices, each of them occurring around $k$ times.

The reason behind why we need the homomorphic property (15) is to help Alice in computing $h_{pk_h}(B)$. After Bob finishes the computation of $\mathrm{Enc}_{pk}(B)$, he picks a random $r$, computes $h_{pk_h}(r)$ and $\mathrm{Enc}_{pk}(B - r)$, and sends them to Alice. Alice then decrypts $\mathrm{Enc}_{pk}(B-r)$, computes $h_{pk_h}(B-r)$,

**Require** Alice: Probe $\mathbf{q}$; Bob: Fastmap pivot objects, PCA basis, and quantization step-size in PCA space, $\{2^{q_i}$ for $i = 1, \ldots, m_2\}$; Public: Scrambled Mapping $\tilde{f}$, Deterministic homomorphic cipher with unknown secret key $\text{Enc}_{pk*, r*}(\cdot)$

**Ensure** Bob gets $f(B)$ where $B \in \Omega$ contains $\mathbf{q}$

  (1)   Alice and Bob computes $\text{Enc}_{pk}[P_{pca}(P_{fm}(\mathbf{q}))_i]$ for $i = 1, \ldots, m_2$.
  (2)   Bob creates an empty list $G := \phi$.
  (3)   Quantization of the projection: for $i = 1, \ldots, m_2$,
      (a)   Bob and Alice execute $R := \text{EXTRACT}[\text{Enc}_{pk}(P_{pca}(P_{fm}(\mathbf{q}))_i)]$ to get the encrypted binary representation of the $i$th dimension of the projection of $\mathbf{q}$.
      (b)   Bob discards $q_i$ lower order encrypted bits from $R$ and add the remaining bits to the set $G$.
  (4)   Bob recombines individual encrypted bits in $G$ to create a single encrypted $\text{Enc}_{pk}(B)$.
  (5)   Bob generates a random number $r$, compute and sends Alice $\text{Enc}_{pk}(B - r)$ and $h_{pk_h}(r)$.
  (6)   Alice decrypts $\text{Enc}_{pk}(B - r)$, computes $h_{pk_h}(B) = h_{pk_h}(B - r) \cdot h_{pk_h}(r)$ and uses it look up the cell indices $f(B) = f'(h_{pk_h}(B))$.
  (7)   If $f(B)$ has multiple cell indices, Alice will send the first one to Bob, wait for a random amount of time, re-execute this entire procedure, and sends the second cell index. The process is repeated until all cell indices in $f(B)$ are exhausted or a match occurs.

PROTOCOL 7: Secure cell index selection SELECT.

and uses the homomorphic property to compute $h_{pk_h}(B) = h_{pk_h}(B - r) \cdot h_{pk_h}(r)$. After that, Alice performs a table lookup to find $f'(h_{pk_h}(B)) = f(B)$. If there are multiple cell indices in $f(B)$, Alice should not send all of them to Bob because he may use this information to significantly reduce the possible choices of $B$ as overlapped bins are rare. Instead, Alice should send one cell index first. Then, she re-encrypts her probe and reruns the entire dimension reduction and index selection process as if she was a different user. The same $f(B)$ will be computed and Alice sends Bob the second index. The whole process is repeated until all the cell indices in $f(B)$ are exhausted or a match occurs.

SELECT (Protocol 7) summarizes the above process on how Bob can identify the cell to which $\mathbf{q}$ belongs. As for the security of Protocol 7, steps 1 through 4 are processing in encrypted domain and thus reveal no secrets to either parties. Steps 5 and 6 allow Bob to identify the cell indices to which $\mathbf{q}$ belongs. As we assume Bob to be semihonest, Bob will not deviate from the protocol by adding any identifiable information to the public table $f'(\cdot)$. Alice has no incentive to deviate from this protocol as a wrong cell index will erase any chance of success in the subsequent encrypted-domain matching with the elements in the cell. The complexities of Protocol 7 are $O(m_1 m_2 + m_2 l)$ on Bob side and $O(m_2 l)$ on Alice side, where $m_1$ is the Fastmap dimension, $m_2$ is the PCA dimension, and $l$ is the bit length of the scaled PCA coordinates. The communication costs are $O(m_1 + m_2 l)$ encrypted numbers.

## 7. Experiments and Discussions

For our experiments, we use the CASIA Iris database from the Chinese Academy of Sciences Institute of Automation (CASIA) [54], a common benchmark for evaluating the performance of iris recognition systems. For the iris feature extraction, we use the MATLAB code from [51] to generate both the iris feature vectors and the masks. Each iris feature vector is 9600 bit long. The similarity threshold $\epsilon$ is set to be 0.35. We select 1948 samples from CASIA based on the following criteria: the distances are smaller than 0.35 between any two samples from the same eye, and larger than 0.40 between any two samples from different eyes. Furthermore, each eye contains at least six good samples and one sample is set aside for testing. A total of 160 individuals are included in our dataset. Our Paillier implementation is based on the Paillier Library developed by J Bethencourt [55]. The key length of the Paillier cipher is set to be 1024 bit which results in 2048-bit ciphertexts.

*7.1. Encrypted Domain Processing.* In this subsection, we summarize the complexity and communication costs of various encrypted-domain processes discussed in this paper. The communication cost is measured based on total amount of information exchanged between Bob and Alice without any overhead from the network stack. The computation time excludes networking time and is computed based on averaging 100 trials. All of them are implemented in C language on a Linux machine with a 2.4 GHz AMD Athlon 64 CPU and 2 GB memory. Table 1 summarizes the results. Encrypted-domain addition and multiplication with plaintext are relatively lightweight, except when the plaintext multiplier is negative (i.e., a large positive number in modular arithmetic). Multiplication between two encrypted numbers (MULT) takes the longest and requires information exchange between Bob and Alice. Hamming distance (DIST) is fast as there are no encryption or decryption. Bit extraction (EXTRACT) takes longer and threshold comparison (COMPARE) takes the longest due to the repeated use of negative numbers, encryption and decryption processes. The long computation time for Query preparation is primarily due the high dimension of the iris feature. The overall computation of an ABAC system consists of a fixed setup
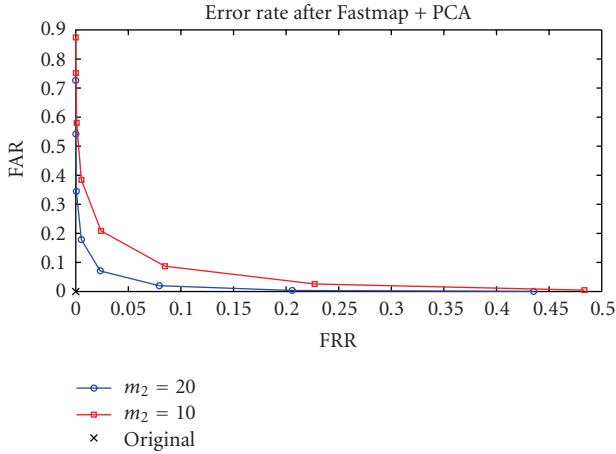
FIGURE 3: FRR versus FAR for using (a) the original feature space, (b) 100*d* Fastmap and then $m_2 = 20$ dimensional PCA, and (c) 100d Fastmap and then $m_2 = 2$ dimensional PCA.
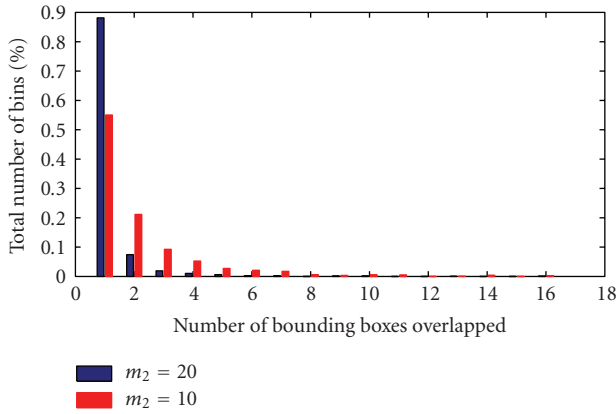


FIGURE 4: Histogram of overlapped bins.



FIGURE 5: Tradeoff between complexity and utility (privacy).

time of query preparation followed by the time taken for the remaining steps scaled by the size of the database. For a database of 10000 iris, our ABAC system is estimated to take 41,490 seconds or 11.5 hours and 120 MBytes of network bandwidth. On the other hand, in a *k*-anonymous ABAC system, the fixed setup time is the Query Preparation and the SELECT process. The matching complexity depends only on *k* but not on the size of the database, except for the rare cases in which the probe falls into an overlapped bin. We shall study the effect of the quantization on the number of overlapped bins in details in Section 7.2. Apart from these exceptions, for the same database of 10000 iris patterns using a *k*-ABAC system with $k = 50$, the time required is only 650 seconds and the bandwidth is 1.3 MBytes.

### 7.2. k-Anonymous Quantization.

In the *k*-ABAC system, we first use Fastmap to reduce the original 9600-bits iris code into 100-dimension Euclidean space. Then we use PCA again to further reduce the dimension. Two PCA dimensions, 10 and 20, are tested in our experiments. These steps were performed on a machine running Windows XP Pro. with
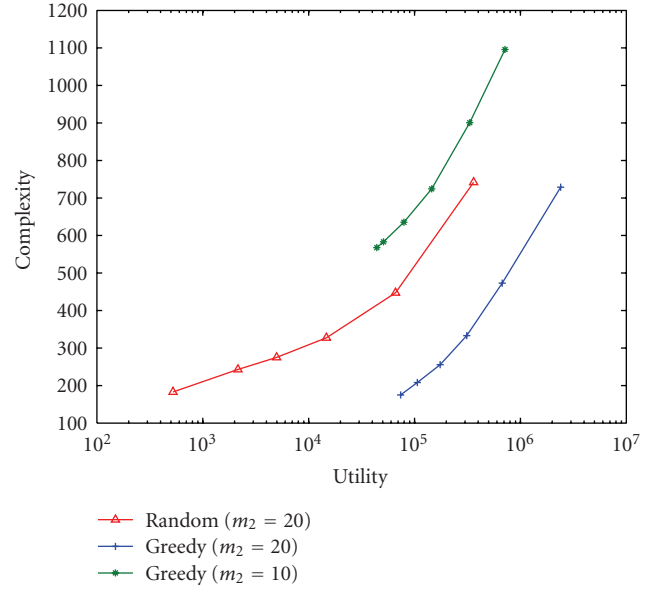
3.4 GHz Intel Pentium 4 CPU and 2 GB of RAM. The rum times for Fastmap and PCA are 36.24 and 0.274 seconds. There is a loss in performance in each step of projection as the distances cannot be represented as accurately. The plots of False Accept Rates (FAR) versus False Reject Rate (FRR) for the original space and the two projected cases are shown in Figure 3. The performance clearly declines as the dimension decreases from 20 to 10. The consequence of dimension reduction is that the similarity structure cannot be well approximated in low dimensions. In defining the *k*-Anonymous quantization, we rely on a uniform quantization grid and similarity within a single iris is estimated based on a bounding box of similar features. If the similarity structure is poorly represented, bounding boxes begin to overlap. Probe falling in overlapped areas may need to invoke multiple cells, and thus increase the computational complexities. Figure 4 shows the histogram of the fraction of bins that overlap different numbers of bounding boxes. For $m_2 = 20$, 88% of the bins are contained in only one bounding box and 96% in at most two bounding boxes. When the dimension is reduced to $m_2 = 10$, these numbers reduce to 55% and 76%. Even though overlapped bins are not necessarily classified into different cells by the KAQ algorithm, their total number serves as the upper bound of bins with multiple cell affiliations.

Next, we consider the performance of KAQ. This algorithm, programmed in C language, was run on a machine running Windows XP Pro. with 2.0 GHz AMD Athlon 64 CPU and 1 GB of RAM. The execution time is a function of the size of the database and takes less than 2 milliseconds to complete regardless of the parameters we used. We have tested the algorithms for various values of *k* and for $m_2 = 10$ and 20 dimensions. Table 2 summarizes the outputs of the KAQ algorithm at $m_2 = 20$. The first column shows the input parameter *k*. The second column shows the average

TABLE 1: Time and communication complexities of encrypted-domain processing.

| Process | Bob's time in seconds | Alice's time in seconds | Communication (Kbits) |
| --- | --- | --- | --- |
| Encryption $\text{Enc}_{pk}(x)$ | $17.3 \times 10^{-3}$ | — | — |
| Decryption $\text{Dec}_{sk}(c)$ | $12.8 \times 10^{-3}$ | — | — |
| Addition $\text{Enc}_{pk}(x) \cdot \text{Enc}_{pk}(y)$ | $13 \times 10^{-6}$ | — | — |
| Multiplication $\text{Enc}_{pk}(x)^y, y \geq 0$ | $0.143 \times 10^{-3}$ | — | — |
| Multiplication $\text{Enc}_{pk}(x)^y, y < 0$ | $30.1 \times 10^{-3}$ | — | — |
| MULT | $47.9 \times 10^{-3}$ | $43.0 \times 10^{-3}$ | 3 |
| DIST[a] | $98 \times 10^{-3}$ | — | — |
| EXTRACT[b] | 0.845 | 0.421 | 56 |
| COMPARE[b] | 2.06 | 0.602 | 42 |
| Query preparation (Step 2 in ABAC) | — | 290 | — |
| Remaining steps in ABAC[a] | 3.05 | 1.07 | 98 |
| SELECT[c] | 2149.842 | 3.455 | 5522 |

[a]Average running time for each entry in DB amortized over 100 entries, with the dimension of each entry equal to 9600.
[b]14 bits operand are used as they are sufficient for the Hamming distance.
[c]Fastmap dimension $m_1 = 100$; PCA dimension $m_2 = 20$ and $l = 64$.

TABLE 2: Output statistics of the KAQ algorithm at $m_2 = 20$.

| $k$ | Cell size | Utility | Cell utility | Complexity |
| --- | --- | --- | --- | --- |
| 100 | $106.5 \pm 4.0$ | 73856 | $80262 \pm 4885$ | $160.2 \pm 146$ |
| 120 | $127.2 \pm 4.7$ | 106438 | $115881 \pm 5532$ | $189.5 \pm 165$ |
| 150 | $157.7 \pm 5.8$ | 174855 | $179855 \pm 5818$ | $232.1 \pm 191$ |
| 200 | $207.9 \pm 5.1$ | 311756 | $315252 \pm 3016$ | $300.1 \pm 226$ |
| 300 | $303.5 \pm 5.0$ | 673085 | $679503 \pm 8149$ | $423.7 \pm 275$ |

TABLE 3: Output statistics of the KAQ algorithm at $m_2 = 10$.

| $k$ | Cell size | Utility | Cell utility | Complexity |
| --- | --- | --- | --- | --- |
| 100 | $153.9 \pm 52$ | 44074 | $87421 \pm 67533$ | $567.7 \pm 354$ |
| 120 | $162.4 \pm 49$ | 50965 | $95583 \pm 65760$ | $582.9 \pm 355$ |
| 150 | $182.5 \pm 41$ | 79450 | $118268 \pm 59472$ | $635.3 \pm 377$ |
| 200 | $224.1 \pm 26$ | 145441 | $176631 \pm 42509$ | $724.2 \pm 404$ |
| 300 | $315.3 \pm 7.9$ | 332649 | $358721 \pm 12955$ | $900.8 \pm 436$ |

and standard deviation of the number of data points in each cell. $k$ is the lower bound of the cell size and KAQ manages to produce consistent cell sizes with small variance. The third column shows the utility function as defined in (10) which measures the minimum level of privacy among all the cells. The fourth column considers the average utility function and its standard deviation over all the cells. Again, the standard deviations are generally very small demonstrating the consistency across different cells. The utility increases with $k$ as the bigger the $k$ is, the more data points are grouped into the same cell. On the other hand, neither the cell size nor $k$ is reliable metrics of complexity as they do not take the overlapping among cells into consideration. To provide a more realistic measure, we hold back one data point per individual iris during the quantization construction and use them to test the true complexity. Specifically, we measure complexity based on the actual number of data points in the union of cells that contains the testing probe. The results are tabulated in the last column. The complexity number will be larger than the cell size if the probe falls into a bin that overlaps more than one cell and the number of data points will at least double. The quantized increase in the number of cells accounts for the large standard deviation. In general, the complexity is roughly 1.5 times that of the average cell size.

Table 3 summarizes the results for KAQ $m_2 = 10$. While showing a similar trend as Table 2, there are a number of major differences. All the measurements show a much higher level of noise as compared with the previous experiments.

This is due to the significant amount of overlapping among bounding boxes. Thus, even when the KAQ algorithm tries to evenly spread the data points, the overlapping forces bounding boxes to be in many cells at the same time. As a consequence, the complexity numbers are much higher than those from KAQ at $m_2 = 20$. The utility numbers also decrease from before as the distance measurements are not as well preserved.

As there are no comparable quantization schemes in the literature for maximizing privacy, we have chosen, as a reference scheme, random cell assignment for each bounding box at a target number of cells. We call this scheme RANDOM and it is a sensible choice for ensuring individuals with similar iris features to be grouped at a random manner. The testing methodology is that we would first run the KAQ algorithm approach for a specific $k$, and then use the same number of cells for RANDOM. Ten random trials of RANDOM are run at each operating point. The results for $m_2 = 20$ are summarized in Table 4. As expected, RANDOM shows a significant drop in utility as no explicit optimization mechanism is used. The complexity numbers are comparable to those of KAQ as they are mostly a function of the geometry of the data distribution which dictates the overlapping of the bounding boxes.

We finally present the idea of trading off complexities with privacy, as measured by the utility function. We plot the complexity versus utility for all the three schemes in Figure 5. We have left out the error bars as the standard deviation

TABLE 4: Output statistics of the RANDOM algorithm at $m_2 = 20$.

| Cell size | Utility | Cell Utility | Complexity |
|---|---|---|---|
| $102.5 \pm 46$ | $963.0 \pm 764$ | $21620.7 \pm 18805$ | $183.0 \pm 155$ |
| $121.8 \pm 50$ | $2104.8 \pm 694$ | $29927.7 \pm 23258$ | $242.9 \pm 230$ |
| $150.8 \pm 57$ | $7732.1 \pm 3192$ | $45517.4 \pm 34276$ | $275.1 \pm 237$ |
| $196.9 \pm 65$ | $11747.6 \pm 5714$ | $76586.2 \pm 48475$ | $327.4 \pm 259$ |
| $285.2 \pm 71$ | $50150.8 \pm 17737$ | $156620.9 \pm 71238$ | $447.3 \pm 335$ |

for the complexity numbers is not meaningful due to the quantized effect of cell increase. This figure demonstrates that the KAQ algorithm provides a good level of privacy protection as the curves for both dimension reside on the high end of utility. While KAQ at $m_2 = 10$ does not scale well when a high level of privacy is needed, KAQ at $m_2 = 20$ stays relatively linear. RANDOM is not able to offer much privacy protection.

## 8. Conclusions

In this paper, we have proposed a design for the Anonymous Biometric Control System (ABAC) which allows a biometric server to verify the membership status of a user without knowing his/her identity. The system is composed of various secure multiparty protocols including Hamming distance computation, bit extraction, comparison and result aggregation, all implemented with a homomorphic cipher. To reduce the computational and communication complexities of such a system, we have proposed a framework called the $k$-Anonymous ABAC system that tradeoffs privacy and complexity by quantizing the search space into cells, each of which contains at least $k$ members. Complexity is reduced by restricting the encrypted domain search process to a small number of cells. Privacy is measured by the dissimilarity of the smallest cell. A greedy quantization scheme on a reduced-dimensional space called $k$-Anonymous Quantization has been devised to derive the optimal quantization that maximizes privacy. Secure procedures have been proposed to perform the dimensional reduction and cell lookup. Experimental results on a dataset of iris patterns demonstrate the effectiveness of our techniques in terms of balancing privacy and computational costs. We are currently investigating the extension of the proposed systems to handle a broader class of malicious behaviors. Also, we are interested in improving the efficiency of the homomorphic cipher, particularly in the case when small plaintext numbers are used. Another topic under investigation is the scalability of the $k$-Anonymous Quantization to a much larger dataset.

## References

[1] A. Jesdanun, *Youtube, Vacom Agree to Mask Viewer Data*, Associated Press, 2007.

[2] W. Hassan and L. Logrippo, "Governance policies for privacy access control and their interactions," in *Feature Interactions in Telecommunication and Software Systems VIII*, D. Amyot and L. Logrippo, Eds., pp. 114–130, IOS Press, 2005.

[3] L. Sweeney, "$k$-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[4] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[5] S. Hoque, M. Fairhurst, G. Howells, and F. Deravi, "Feasibility of generating biometric encryption keys," *Electronics Letters*, vol. 41, no. 6, pp. 309–311, 2005.

[6] U. Uludag and A. Jain, "Securing fingerprint template: fuzzy vault with helper data," in *Proceedings of the Computer Vision and Pattern Recognition Workshops (CVPR '06)*, p. 163, June 2006.

[7] E. M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, 2005.

[8] V. Ciriani, S. D. C. di Vimercati, S. Foresti, and P. Samarati, "$k$-anonymity," in *Secure Data Management in Decentralized Systems*, vol. 33, pp. 323–353, Springer, New York, NY, USA, 2007.

[9] O. Goldreich, *Foundations of Cryptography: Volume II Basic Applications*, Cambridge University Press, Cambridge, UK, 2004.

[10] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen, "On private scalar product computation for privacy-preserving data mining," in *Proceedings of the 7th Annual International Conference in Information Security and Cryptology (ICISC '04)*, vol. 3506, pp. 104–120, Seoul, South Korea, December 2005.

[11] M. Naor and B. Pinkas, "Oblivious polynomial evaluation," *SIAM Journal on Computing*, vol. 35, no. 5, pp. 1254–1281, 2006.

[12] Y.-C. Chang and C.-J. Lu, "Oblivious polynomial evaluation and oblivious neural learning," *Theoretical Computer Science*, vol. 341, no. 1–3, pp. 39–54, 2005.

[13] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC '99)*, pp. 245–254, Atlanta, Ga, USA, May 1999.

[14] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic encryption and secure comparison," *International Journal of Applied Cryptography*, vol. 1, no. 1, pp. 22–31, 2008.

[15] M. Fischlin, "A cost-effective pay-per-multiplication comparison method for millionaires," in *Proceedings of the Conference on Topics in Cryptology: The Cryptographer's Track at RSA (CT-RSA '2001)*, vol. 2020 of *Lecture Notes in Computer Science*, pp. 457–472, San Francisco, Calif, USA, April 2001.

[16] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (FOCS '82)*, pp. 160–164, 1982.

[17] G. Aggarwal, N. Mishra, and B. Pinkas, "Secure computation of the $k^{th}$-ranked element," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '04)*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 40–55, 2004.

[18] E. Kiltz, P. Mohassel, E. Weinreb, and M. Franklin, "Secure linear algebra using linearly recurrent sequences," in *Proceedings of the 4th Theory of Cryptography Conference (TCC '07)*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 291–310, Amsterdam, The Netherlands, February 2007.

[19] R. Cramer and I. Damgaard, "Secure distributed linear algebra in constant number of rounds," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in*

*Cryptology (IACR CRYPTO '01)*, vol. 2139 of *Lecture Notes In Computer Science*, pp. 119–136, Springer, 2001.

[20] B. Schoenmakers and P. Tuyls, "Efficient binary conversion for Paillier encrypted values," in *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '06)*, vol. 4004 of *Lecture Notes in Computer Science*, pp. 522–537, St. Petersburg, Russia, May-June 2006.

[21] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A new privacy-preserving distributed *k*-clustering algorithm," in *Proceedings of the 6th SIAM International Conference on Data Mining (SDM '06)*, pp. 494–498, 2006.

[22] M. C. Doganay, T. B. Pedersen, Y. Saygin, E. Savaş, and A. Levi, "Distributed privacy preserving *k*-means clustering with additive secret sharing," in *Proceedings of the International Workshop on Privacy and Anonymity in Information Society (PAIS '08)*, vol. 331, pp. 3–11, Nantes, France, 2008.

[23] S. Samet and A. Miri, "Privacy preserving ID3 using gini index over horizontally partitioned data," in *Proceedings of the 6th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '08)*, pp. 645–651, March-April 2008.

[24] J. Zhan, "Privacy-preserving decision tree classification in horizontal collaboration," in *Proceedings of the 1st International Conference on Security of Information and Networks (Sin '07)*, 2007.

[25] Y. Lindell and B. Pinkas, "Privacy preserving data mining," *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

[26] J. Vaidya, H. Yu, and X. Jiang, "Privacy-preserving SVM classification," *Knowledge and Information Systems*, vol. 14, no. 2, pp. 161–178, 2008.

[27] C. Orlandi, A. Piva, and M. Barni, "Oblivious neural network computing via homomorphic encryption," *EURASIP Journal on Information Security*, vol. 2007, Article ID 37343, 11 pages, 2007.

[28] R. Wright and Z. Yang, "Privacy-preserving Bayesian network structure computation on distributed heterogeneous data," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pp. 713–718, Seattle, Wash, USA, 2004.

[29] S.-C. Cheung and T. Nguyen, "Secure signal processing between distrusted network terminals," *EURASIP Journal on Information Security*, vol. 2007, Article ID 051368, 10 pages, 2007.

[30] M. O. Rabin, "How to exchange secrets by oblivious transfer," Tech. Rep. TR-81, Harvar Aiken Computation Laboratory, 1981.

[31] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Theory of Cryptography Conference (TCC '05)*, J. Killian, Ed., vol. 3378 of *Lecture Notes in Computer Science*, pp. 325–342, Springer, 2005.

[32] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '01)*, pp. 448–457, Washington, DC, USA, January 2001.

[33] M. Naor and K. Nissim, "Communication complexity and secure function evaluation," in *Proceedings of the Electronic Colloquium on Computational Complexity (ECCC '01)*, vol. 8, 2001.

[34] C. Cachin, J. Camenisch, J. Kilian, and J. Muller, "One-round secure computation and secure autonomous mobile agents," in *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP '00)*, pp. 512–523, Geneva, Switzerland, July 2000.

[35] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, Article ID 13801, 10 pages, 2007.

[36] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*, pp. 169–179, Bethesda, Md, USA, 2009.

[37] M. Cooney, "Ibm touts encryption innovation," Computer World, June 2009.

[38] B. Schneier, "Homomoprhic encryption breakthrough," in *Schneier on Security*, 2009.

[39] P. Pailler, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '99)*, vol. 1592, pp. 223–238, May 1999.

[40] Z. Erkin, A. Piva, S. Katzenbeisser, et al., "Protection and retrieval of encrypted multimedia content: when cryptography meets signal processing," *EURASIP Journal on Information Security*, vol. 2007, Article ID 78943, 20 pages, 2007.

[41] T. Bianchi, A. Piva, and M. Barni, "Discrete cosine transform of encrypted images," in *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP '08)*, pp. 1668–1671, October 2008.

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

[43] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[44] T. F. Cox and M. A. Cox, *Multidimensional Scaling*, Chapman & Hall, Boca Raton, Fla, USA, 2nd edition, 2001.

[45] C. Faloutsos and K.-I. Lin, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD '95)*, pp. 163–174, San Jose, Calif, USA, May 1995.

[46] J. Bourgain, "On lipschitz embedding of finite metric spaces in Hilbert space," *Israel Journal of Mathematics*, vol. 52, no. 1-2, pp. 46–52, 1985.

[47] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimneions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*, pp. 518–529, September 1999.

[48] O. Goldreich, *Foundations of Cryptography: Volume 1, Basic Tools*, Cambridge University Press, Cambridge, UK, 2007.

[49] P. Mohanty, S. Sarkar, and R. Kasturi, "Privacy & security issues related to match scores," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '06)*, pp. 162–165, June 2006.

[50] J. Katz and Y. Lindell, *Introduction To Modern Cryptography*, Chapman & Hall, Boca Raton, Fla, USA, 2008.

[51] L. Masek and P. Kovesi, "Matlab source code for a biometric identification system based on iris patterns," Tech. Rep., The School of Computer Science and Software Engineering, The University of Western Australia, Perth, Australia, 2003.

[52] D. Filho and P. Barreto, "Demonstrating data possession and uncheatable data transfer," Cryptology ePrint Archive, Report 2206/150, 2006.

[53] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '04)*, vol. 2004, pp. 226–240, Berkeley, Calif, USA, May 2004.

[54] T. Tan and Z. Sun, "Casia-irisv3," Tech. Rep., Chinese Academy of Sciences Institute of Automation, 2005, http://www.cbsr .ia.ac.cn/IrisDatabase.htm.

[55] J. Bethencourt, Paillier Library, UC Berkeley, http://acsc .cs.utexas.edu/.