This extended abstract appeared in
Proceedings of the 6th International Conference on Cryptology and Network Security (CANS '07)
December 8–10, 2007, Singapore – T. Okamoto Ed. Springer-Verlag, LNCS 4856, pages 175-193.

Extended Private Information Retrieval and its Application in Biometrics Authentications*

Julien Bringer[1], Hervé Chabanne[1], David Pointcheval[2], and Qiang Tang[2]

[1] Sagem Sécurité
[2] Departement d'Informatique, École Normale Supérieure
45 Rue d'Ulm, 75230 Paris Cedex 05, France

Abstract In this paper we generalize the concept of Private Information Retrieval (PIR) by formalizing a new cryptographic primitive, named Extended Private Information Retrieval (EPIR). Instead of enabling a user to retrieve a bit (or a block) from a database as in the case of PIR, an EPIR protocol enables a user to evaluate a function f which takes a string chosen by the user and a block from the database as input. Like PIR, EPIR can also be considered as a special case of the secure two-party computation problem (and more specifically the oblivious function evaluation problem). We propose two EPIR protocols, one for testing equality and the other for computing Hamming distance. As an important application, we show how to construct strong privacy-preserving biometric-based authentication schemes by employing these EPIR protocols.

1 Introduction

This paper describes a new primitive, Extended Private Information Retrieval (EPIR) which is a natural generalization of PIR, and two EPIR protocols, one for testing equality and the other for computing Hamming distance. This work is partially motivated by the growing privacy requirements in processing sensitive information such as biometrics.

1.1 Related Work

With respect to the functionality, an EPIR is indeed a combination of a PIR [10] and a general secure two-party computation protocol [26,51]. Next, we briefly review the literature in both areas.

The concept of PIR was proposed by Chor *et al.* [10]. A PIR protocol enables a user to retrieve a bit from a database which contains a bit string. Chor *et al.* defined user privacy for PIR in the information-theoretical setting, which captures the concept that the database (with unlimited resources) learns nothing about which bit the user has retrieved. They also proposed a number of multi-database protocols that are secure in the information-theoretical setting. Chor and Gilboa [9] proposed to construct multi-database PIR under computational assumptions. Kushilevitz and Ostrovsky [33] presented a definition of user privacy in computational setting, where a PIR protocol achieves user privacy if, for any query for i-th bit, the database learns nothing about the index i. They showed that one can achieve single-database PIR under the Quadratic Residuosity assumption with communication complexity $O(N^c)$ for any $c > 0$, where N is the database size throughout the paper. Cachin, Micali, and Stadler [7] proposed a single-database PIR scheme with poly-logarithmic communication complexity $O((\log N)^8)$ based on the Φ-hiding assumption.

Chor *et al.* [10] also proposed the notion of Private Block Retrieval (PBR), a natural extension to single-bit PIR, in which instead of retrieving only one bit, the user retrieves a d-bit block. They proposed an efficient method for the transformation from PIR to PBR. Lipmaa [35] proposed a PBR scheme with communication complexity $\Theta(\Omega((\log N)^{3-o(1)})(\log N)^2 + d \log N)$. Gentry and Ramzan [23] proposed a single-database PBR protocol based on the decision subgroup problem, with communication complexity $O(k + d)$ where $k \geq \log N$ is the security parameter.

* This work is partially supported by french ANR RNRT project BACH.

Gertner *et al.* [24] introduced the notion of data privacy in the computational setting, where a PIR protocol achieves database privacy if, for any query, the user cannot tell whether it is an ideal-world execution or a real-world execution. In an ideal-world execution the user interacts with a simulator which takes only a single bit from the database as input, while in a real-world execution the user interacts with the database. If a PIR protocol achieves both user privacy and data privacy, then it is said to be SPIR (symmetrically-private information retrieval) which is also referred to as one-out-of-N oblivious transfer [13]. Mishra and Sarkar [37] proposed a single-server SPIR protocol which can have communication complexity $O(N^{\epsilon})$ for any $\epsilon > 0$. The protocol is proven secure under the XOR assumption defined by Mishra and Sarkar.

Gasarch [22] provides a very detailed summary of PIR/PBR protocols and lower/upper bounds on communication complexity, and Ostrovsky and Skeith III [39] also provides a summary. To facilitate our discussion, we use the notation PIR to denote both PIR and PBR, and generalise the setting of PIR to be: a database $\mathcal{DB}$ contains a list of N blocks $\mathbf{R} = (R_1, R_2, \cdots, R_N)$, and a user $\mathcal{U}$ can run a PIR protocol to retrieve R_i from $\mathcal{DB}$, for any $1 \leq i \leq N$.

As a special case of secure two-party computation problem, the concept of EPIR is relevant to the oblivious function evaluation [8,20,38]. Canetti *et al.* [8] study the problem that a client privately evaluate a public function which takes inputs from one or more servers. Note that the client does not have any private input to the function. Naor and Pinkas [38] study the problem that a receiver privately evaluates a function $\mathsf{f}(a)$ by interacting with a sender, where f is a secret polynomial of the sender and a is a secret input of the receiver. Freedman *et al.* [20] study the keyword search problem that a client privately evaluates whether a keyword is contained in a database. EPIR can be considered to be a generalization of the these problems (in the single database case). Next, we briefly review some works which are related to equality test and hamming distance computation. In [11,19], the authors studied how to compare two commonly shared strings and determine whether they are the same. Freedman, Nissim, and Pinkas [21] studied two-party set-interaction problems and proposed a number of protocols. Du and Atallah [50,17] considered the secure computation in an environment similar to that of EPIR, and proposed protocols based on solutions to Yao's millionaire problem. Goethals *et al.* [25] showed the weakness in the private scalar product protocols [16,48] and proposed a new protocol based on homomorphic encryption schemes. Kiltz, Leander, and Malone-Lee [32] proposed some methods for a user to compute the mean (and other statistics) over the data in a database. However, they did not propose any specific security model for this type of computation, and their protocols either require a semi-trusted third party or are very inefficient in round and communication complexity. Note that Kiltz, Leander, and Malone-Lee [32] showed that some approach in [17] leaks information in some applications. Boneh, Goh, and Nissim [3] proposed an encryption scheme (referred to as the BGN encryption scheme) and used it for evaluating 2-DNF formulas. As an application, they showed how to construct efficient PIR protocols based on their encryption scheme.

1.2 Practical Motivation

Biometrics, such as fingerprint and iris, have been used to a high level of security in order to cope with the increasing demand for reliable and highly-usable information security systems, because they have many advantages over cryptographic credentials. However, there are some obstacles for a wide adoption of biometrics in practice. Among them, one is that biometric features are volatile over the time so that it cannot be integrated into most of the legacy systems. This means that approximate matching might be necessary for an identification or authentication. The other is that biometrics are usually considered to be sensitive, so that there is big privacy concern in using them. To address the volatility of biometrics, error-correction concept is widely used in the literature (e.g. [4,5,12,15,14,30,31,45,42]). Employing this concept, some public information is firstly generated based on a reference biometric template, and later, a newly-captured template could help to recover the reference template if their distance (in a certain space) is not too large. In [34,44,45,46,49], the authors attempted to enhance privacy protection in

biometric authentication schemes, where the privacy means that the compromise of the database will not enable the attacker to recover the biometric template. Ratha, Connell, and Bolle [2,41] introduced the concept of *cancelable biometrics* in an attempt to solve the revocation and privacy issues related to biometric information. More recently, Ratha *et al.* [40] intensively elaborated this concept in the case of fingerprint-based authentication systems. In addition, Atallah *et al.* [1] proposed a method, in which biometric templates are treated as bit strings and subsequently masked and permuted during the authentication process. Schoenmakers and Tuyls [43] proposed to use homomorphic encryption schemes for biometric authentication schemes by employing multi-party computation techniques. Practical concerns, security issues, and challenges about biometrics have been discussed in a number of papers (e.g. [2,29,36,41,47]).

Despite these efforts, there are still some concerns which require further investigation. The most important one is that privacy may mean much more than recovering the biometric template. For example, an application server may not be trusted to store biometric information, and, even if an independent database stores biometric information, the application server's access to the biometric information still need to be restricted. In addition, it is also desirable to simplify the storage requirements for the human users and the (communication) client. Bringer *et al.* [6] proposed a biometric-based authentication protocol which protects the sensitive relationship between a biometric feature and relevant pseudorandom identity. Their protocol makes use of the Goldwasser-Micali encryption scheme and is less efficient in communication than those described in Section 5.

1.3 Our Contributions

We generalize the concept of PIR by formalizing a new cryptographic primitive, named Extended Private Information Retrieval (EPIR). Instead of enabling a user to retrieve a block from a database as in the case of PIR, an EPIR protocol enables a user to evaluate a function f which takes a string chosen by the user and a block from the database as input[1]. If f is defined to be a function that simply returns the block from the database then the EPIR protocol is indeed a traditional PIR protocol. Analogous to the privacy properties of PIR, we define two privacy properties for EPIR, including (1) user privacy which captures the concept that, for any query, the database should know nothing about block index the user has queried and the user's input to f, (2) database privacy captures the concept that, from a single query, the user should obtain no more information than the output of function f. Note that we focus on the single-database computational setting in this paper.

We further propose two EPIR protocols: one for testing equality and the other for computing Hamming distance. The first protocol is based on a PIR protocol and the ElGamal encryption scheme (described in Appendix A)[18], and the second protocol is based on a PIR protocol and the BGN encryption scheme (described in Appendix B) [3]. In both EPIR protocols, in order to achieve database privacy, the PIR protocols employed do not need to achieve database privacy.

As an important application, we show a modular way to construct biometric-based authentication schemes by employing an EPIR protocol. Due to the privacy properties of EPIR, these schemes achieve strong privacy properties against a malicious server and a malicious database which will not collude. It is worth noting that our proposal is not focused on a specific biometric, but rather on a generalisation of biometrics which can be represented as binary strings in the Hamming space. Iris is such a type of biometric that can be easily encoded into a binary string [28].

1.4 Organization of the Paper

The remainder of the paper is organized as follows. In Section 2 we present the security definitions for EPIR. In Section 3 we describe an EPIR protocol for testing equality of two binary strings based on

[1] We assume that the index of the block from the database is also chosen by the user.

the ElGamal encryption scheme. In Section 4 we describe an EPIR protocol for computing Hamming distance of two binary strings based on the BGN encryption scheme. In Section 5 we propose two biometric-based authentication schemes by employing these two EPIR protocols. In Section 6 we conclude the paper.

2 Privacy Definitions for EPIR

Formally, a (single-database) EPIR protocol involves two principals: a database $\mathcal{DB}$ which holds a set of N blocks $\mathbf{R} = (R_1, R_2, \cdots, R_N)$ where $R_j \in \{0,1\}^{\ell_1}$ and ℓ_1 is an integer, a user $\mathcal{U}$ which retrieves the value of a function $\mathsf{f}(R_i, X)$ where $X \in \{0,1\}^{k_1}$ is chosen by the user, k_1 is an integer, and the index i is also chosen by the user. We assume that the description of f is public and N is a public constant integer. Without loss of generality, we further assume that the retrieval is through a $\mathsf{retrieve}(\mathsf{f}, i, X)$ query.

2.1 Notation

We first describe some conventions for writing probabilistic algorithms and experiments. The notation $x \stackrel{R}{\leftarrow} S$ means x is randomly chosen from the set S. If $\mathcal{A}$ is a probabilistic algorithm, then $\mathcal{A}(\mathsf{Alg}; \mathsf{Func})$ is the result of running $\mathcal{A}$, which can have any polynomial number of oracle queries to the functionality Func, interactively with Alg which answers the oracle query issued by $\mathcal{A}$. For clarity of description, if an algorithm $\mathcal{A}$ runs in a number of stages then we write $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \cdots)$. As a standard practice, the security of a protocol is evaluated by an experiment between an attacker and a challenger, where the challenger simulates the protocol executions and answers the attacker's oracle queries. Without specification, algorithms are always assumed to be polynomial-time.

Specifically, in our case, there is only one functionality, namely $\mathsf{retrieve}$. If the attacker is a malicious $\mathcal{DB}$, the challenger samples the index i and X from the distribution specified in the protocol and issues $\mathsf{retrieve}$ queries to the attacker. If the attacker is a malicious $\mathcal{U}$ then it can freely chooses the index i and X (that may derivate from the distribution specified in the protocol) and issues $\mathsf{retrieve}$ queries to the challenger.

In addition, we have the following definitions for negligible and overwhelming probabilities.

Definition 1. The function $P(\ell) : \mathbb{Z} \to \mathbb{R}$ is said to be negligible if, for every polynomial $f(\ell)$, there exists an integer N_f such that $P(\ell) \leq \frac{1}{f(\ell)}$ for all $\ell \geq N_f$. If $P(\ell)$ is negligible, then the probability $1 - P(\ell)$ is said to be overwhelming.

2.2 User Privacy

This property is an analog to the user privacy property in the case of PIR where user privacy captures the concept that $\mathcal{DB}$ knows nothing about block index that $\mathcal{U}$ has queried. However, in the case of EPIR, we wish user privacy to imply more than that $\mathcal{DB}$ knows nothing about the block index $\mathcal{U}$ has queried. Consider a toy example, in which an EPIR protocol is constructed as follows: $\mathcal{U}$ simply sends X to the database which computes $\mathsf{f}(R_j, X)$ $(1 \leq j \leq N)$, and $\mathcal{U}$ then runs a PIR to retrieve $\mathsf{f}(R_i, X)$. It is clear that, if the PIR protocol achieves user privacy then $\mathcal{DB}$ learns nothing about the index in the toy protocol. However, if $\mathsf{f}(R_j, X)$ $(1 \leq j \leq N)$ are equal then $\mathcal{DB}$ knows the result obtained by $\mathcal{U}$.

Informally, the user privacy for EPIR captures the concept that, for any $\mathsf{retrieve}(\mathsf{f}, i, X)$ query, $\mathcal{DB}$ knows nothing about the queried block index i and the user's string X. Formally, an EPIR protocol

achieves user privacy if any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$ has only a negligible advantage in the following game, where the attacker's advantage is $|\Pr[b' = b] - \frac{1}{2}|$.

$$
\begin{array}{|ll}
\multicolumn{2}{l}{\mathbf{Exp}_{\mathcal{A}}^{\text{user-privacy}}} \\
\mathbf{R} = (R_1, R_2, \cdots, R_N) & \leftarrow \mathcal{A}_1(1^\ell) \\
1 \le i_0, i_1 \le N; X_0, X_1 \in \{0,1\}^{k_1} & \leftarrow \mathcal{A}_2(\mathit{Challenger}; \mathsf{retrieve}) \\
b & \stackrel{R}{\leftarrow} \{0,1\} \\
\emptyset & \leftarrow \mathcal{A}_3(\mathit{Challenger}; \mathsf{retrieve}(\mathsf{f}, i_b, X_b)) \\
b' & \leftarrow \mathcal{A}_4(\mathit{Challenger}; \mathsf{retrieve})
\end{array}
$$

In this game, the attacker $\mathcal{A}$ is a malicious $\mathcal{DB}$. For the clarity, we rephrase the game as follows.

1. The attacker $\mathcal{A}_1$ generates N blocks $\mathbf{R} = (R_1, R_2, \cdots, R_N)$.
2. The attacker $\mathcal{A}_2$ can request the challenger to start any (polynomial) number of $\mathsf{retrieve}$ queries. At some point, $\mathcal{A}_2$ outputs (i_0, i_1, X_0, X_1) for a challenge.
3. The challenger randomly chooses $b \in \{0,1\}$ and issues a $\mathsf{retrieve}(\mathsf{f}, i_b, X_b)$ query to the attacker $\mathcal{A}_3$.
4. The attacker $\mathcal{A}_4$ can continue requesting the challenger to start any (polynomial) number of $\mathsf{retrieve}$ queries. At some point, $\mathcal{A}_4$ outputs a guess b'.

Note that the symbol $\emptyset$ means that the attacker $\mathcal{A}_3$ has no explicit output (besides the state information).

2.3 Database Privacy

This property is an analog to the database privacy property in the case of SPIR [24] and the formalization follows that for secure two-party computation [51,26]. Informally, database privacy captures the concept that, from a $\mathsf{retrieve}(\mathsf{f}, i, X)$ query, $\mathcal{U}$ obtains no more information than $\mathsf{f}(R_{i'}, X')$ for some $1 \le i' \le N$ and $X' \in \{0,1\}^{k_1}$. As in [24], we do not require that $i' = i$ and $X' = X$ because a malicious $\mathcal{U}$ may construct the query without following the specification. The concept can also be rephrased as follows: $\mathcal{U}$ cannot tell whether it is an ideal-world execution and a real-world execution. In an ideal-world execution $\mathcal{U}$ interacts with a simulator which takes $(i', \mathsf{f}(R_{i'}, X'))$ as input, while in a real-world execution $\mathcal{U}$ interacts with $\mathcal{DB}$.

For the clarity of formalization, let $\mathsf{simulator}_0$ denote $\mathcal{DB}$. Formally, an EPIR protocol achieves database privacy, if there exists a simulator $\mathsf{simulator}_1$ such that any attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has only a negligible advantage in the following game, where the attacker's advantage is $|\Pr[b' = b] - \frac{1}{2}|$. For every $\mathsf{retrieve}$ query, $\mathsf{simulator}_1$ has an auxiliary input from a hypothetical oracle $\mathcal{O}$, where the input is $(i', \mathsf{f}(R_{i'}, X'))$ for some $1 \le i' \le N$ and $X' \in \{0,1\}^{k_1}$.

$$
\begin{array}{|ll}
\multicolumn{2}{l}{\mathbf{Exp}_{\mathcal{A}}^{\text{database-privacy}}} \\
b & \stackrel{R}{\leftarrow} \{0,1\} \\
\mathbf{R} = (R_1, R_2, \cdots, R_N) & \leftarrow \mathcal{A}_1(1^\ell) \\
b' & \leftarrow \mathcal{A}_2(\mathsf{simulator}_b; \mathsf{retrieve})
\end{array}
$$

In this game, the attacker $\mathcal{A}$ is a malicious $\mathcal{U}$. For the clarity, we rephrase the game as follows.

1. The challenger randomly chooses $b \in \{0,1\}$. If $b = 0$ then $\mathsf{simulator}_0$ answers the $\mathsf{retrieve}$ queries from the attacker; otherwise $\mathsf{simulator}_1$ answers such queries.
2. The attacker $\mathcal{A}_1$ generates N blocks $\mathbf{R} = (R_1, R_2, \cdots, R_N)$.
3. The attacker $\mathcal{A}_2$ can start any (polynomial) number of $\mathsf{retrieve}$ queries. At some point, $\mathcal{A}_2$ outputs a guess b'.

We emphasize that the hypothetical oracle $\mathcal{O}$ may have unlimited computing resources. In an attack game, a malicious $\mathcal{U}$ may or may not generate a query by following the protocol specification, nonetheless, in order to answer the attacker's query, $\mathsf{simulator}_1$ only needs to obtain $\mathsf{f}(R_{i'}, X')$ for some $1 \leq i' \leq N$ and $X' \in \{0,1\}^{k_1}$. As a result, if the attacker cannot distinguish the interactions with $\mathsf{simulator}_0$ and $\mathsf{simulator}_1$, then, for each query, it obtains no more information about $\mathbf{R}$ than i' and $\mathsf{f}(R_{i'}, X')$, which is what $\mathsf{simulator}_1$ needs to answer the query.

2.4 Security of EPIR

Analogous to the case of other primitives, a (useful) EPIR protocol should be sound, which means that if both $\mathcal{U}$ and $\mathcal{DB}$ follow the protocol specification then $\mathsf{retrieve}(\mathsf{f}, i, X)$ always returns the correct value of $\mathsf{f}(R_i, X)$ with an overwhelming probability.

Definition 2. An EPIR protocol is said to be secure if any attacker has only negligible advantage in the attack games for user privacy and database privacy.

3 EPIR Protocol for testing equality

In this section we present an EPIR protocol which enables $\mathcal{U}$ to compare a string with a block from $\mathcal{DB}$. The function $\mathsf{f}(R_i, X)$ is defined to be 1 if $R_i = X$ and to be 0 otherwise. Suppose every block in $\mathcal{DB}$ has bit-length ℓ_1, X also has bit-length ℓ_1, and N has bit-length ℓ_2.

The construction is based on the ElGamal scheme and a PIR protocol. It is worth noting that, due to the randomization in step 3, the employed PIR protocol does not need to be SPIR (achieving database privacy) in order to guarantee the database privacy for the EPIR.

3.1 Description of the Protocol

The EPIR protocol is as follows.

1. $\mathcal{U}$ generates an ElGamal key pair (pk, sk), where $pk = (p, q, g, y)$, $y = g^x$, and $sk = x$ is randomly chosen from $\mathbb{Z}_q$. It is required that the bit-length of q is at least $\ell_1 + \ell_2 + 1$. Let "$||$" be the string concatenation operator.
2. To retrieve the value $\mathsf{f}(R_i, X)$, for any $1 \leq i \leq N$ and $X \in \{0,1\}^{\ell_1}$, $\mathcal{U}$ first sends pk and an ElGamal ciphertext $(g^r, y^r g^{i||X})$ to $\mathcal{DB}$, where r is randomly chosen from $\mathbb{Z}_q$.
3. After receiving pk and $(g^r, y^r g^{i||X})$ from $\mathcal{U}$, $\mathcal{DB}$ first checks that pk is a valid ElGamal public key[2] and $(g^r, y^r g^{i||X})$ is a valid ElGamal ciphertext. If the check succeeds, $\mathcal{DB}$ computes C_j for every $1 \leq j \leq N$, where r_j, r'_j are randomly chosen from $\mathbb{Z}_q$ and

$$C_j = (g^{r'_j}(g^r)^{r_j}, y^{r'_j}(y^r g^{i||X}(g^{j||R_j})^{-1})^{r_j})$$

4. $\mathcal{U}$ runs a PIR protocol to retrieve C_i from $\mathcal{DB}$. $\mathcal{U}$ then sets $\mathsf{f}(i, X) = 1$ if $\mathsf{Dec}(C_i, sk) = 1$ and sets $\mathsf{f}(i, X) = 0$ otherwise.

It is clear that, in our case, no encoding algorithm Ω is required to guarantee the semantic security of the ElGamal scheme. As to the performance, the communication complexity is dominated by that of the PIR protocol. The computational complexity is dominated by the computation of C_j $(1 \leq j \leq N)$, say $O(N)$ exponentiations for $\mathcal{DB}$. Moreover, it is straightforward to verify the following observation.

[2] In practice, the validity of pk can be certified by a TTP, and the same pk can be used by the user for all his queries.

Observation 1 *For every $1 \leq j \leq N$, if $g^{i||X}(g^{j||R_j})^{-1} \neq 1$, the components of $C_j = (C_{j1}, C_{j2})$ are uniformly and independently distributed over $\mathbb{G}$; otherwise C_{j1} is uniformly distributed over $\mathbb{G}$ and $C_{j2} = (C_{j1})^x$.*

Due to the bit-length requirement on q, if $\ell_1 + \ell_2 + 1$ is very large then the protocol may become impractical. Note that ℓ_2 will be bounded by a reasonably small integer (say 50), because it is hard to imagine that we have a database with 2^{50} records. As a result, in this situation, a simple solution is to work on the records $\mathbf{R}' = (R'_1, R'_2, \cdots, R'_N)$ instead of $\mathbf{R}$, where $R'_j = \mathsf{H}(R_j)$ $(1 \leq j \leq N)$ and H is a collision-resistant hash function with a reasonable output bit-length. Inherently, $\mathcal{U}$ issues a $\mathsf{retrieve}(\mathsf{f}, i, \mathsf{H}(X))$ query to retrieve the value of $\mathsf{f}(R_i, X)$. It is clear that $\mathcal{U}$ gets the correct answer with an overwhelming probability.

Instead of employing the ElGamal encryption scheme, other homomorphic encryption schemes may also be used here though we will need a different randomization method in step 3.

3.2 Security Analysis

It is straightforward to verify that if the PIR protocol is sound then the EPIR protocol for equality is also sound.

Lemma 3 (user privacy). *If the PIR protocol achieves user privacy, then the EPIR protocol for testing equality achieves user privacy based on the DDH assumption.*

Proof. If the proposed scheme does not achieve user privacy (as described in Section 2.2), then we can construct an algorithm $\mathcal{A}'$, which receives a public key pk from the ElGamal challenger and runs $\mathcal{A}$ as a subroutine to break the semantic security of the ElGamal scheme (or, DDH assumption). $\mathcal{A}'$ is defined as follows:

1. On receiving the output $\mathbf{R}$ from $\mathcal{A}_1$, $\mathcal{A}'$ sets its ElGamal public key as pk and faithfully answers the $\mathsf{retrieve}$ queries from $\mathcal{A}_2$.
2. On receiving $\{i_0, i_1, X_0, X_1\}$ from $\mathcal{A}_2$, $\mathcal{A}'$ sends $i_0||X_0$ and $i_1||X_1$ to the ElGamal challenger and obtains a challenge $c_b = \mathsf{Enc}(i_b||X_b, pk)$ where b is the coin toss of the challenger. Then $\mathcal{A}'$ sends pk and c_b to $\mathcal{A}_3$, and runs the PIR protocol to retrieve C_{i_e} where e is a coin toss of $\mathcal{A}'$.
3. $\mathcal{A}'$ faithfully answers $\mathsf{retrieve}$ queries issued by $\mathcal{A}_4$. If $\mathcal{A}$ finally outputs e', then $\mathcal{A}'$ terminates by outputting $b' = e'$.

Let E_1 be the event that $e = b$ in the game. Clearly, $\Pr[E_1] = \frac{1}{2}$. If E_1 occurs, then this is a valid attack game for $\mathcal{A}$ and its advantage is $Adv = |\Pr[e' = e|E_1] - \frac{1}{2}|$. It is straightforward to verify that the following equation holds

$$|\Pr[e' = e|E_1] + \Pr[e' = e|\neg E_1] - 1| = \epsilon$$

where ϵ is negligible. Otherwise, it is straightforward to construct an attacker for the PIR protocol. From this equation, we have the following probability relationships.

$$\begin{aligned}
\Pr[b = b'] &= \Pr[E_1]\Pr[e' = e|E_1] + \Pr[\neg E_1]\Pr[e' \neq e|\neg E_1] \\
&= \frac{1}{2}(\Pr[e' = e|E_1] + \Pr[e' \neq e|\neg E_1]) \\
&= \frac{1}{2} + \frac{1}{2}(\Pr[e' = e|E_1] - \Pr[e' = e|\neg E_1]) \\
&\geq \frac{1}{2} + \frac{1}{2}(\Pr[e' = e|E_1] - (1 - \Pr[e' = e|E_1] + \epsilon)) \\
&= \frac{1}{2} + (\Pr[e' = e|E_1] - \frac{1}{2}) - \frac{\epsilon}{2}
\end{aligned}$$

$$|\Pr[b=b'] - \frac{1}{2}| = |\frac{1}{2} + (\Pr[e'=e|E_1] - \frac{1}{2}) - \frac{\epsilon}{2} - \frac{1}{2}|$$
$$\geq Adv - \frac{\epsilon}{2}$$

Based on the assumption that ElGamal scheme is semantically secure, then we get a contradiction. The lemma now follows. □

Lemma 4 (database privacy). *The EPIR protocol for testing equality achieves database privacy (unconditionally).*

Proof. Recall that, in the attack game definition for database privacy (as described in Section 2.3), $\mathsf{simulator}_0$ is assumed to be $\mathcal{DB}$. We now define a simulator $\mathsf{simulator}_1$ which answers the attacker's query as follows.

1. On receiving pk and and (α_1, α_2), $\mathsf{simulator}_1$ first checks that they are well formed. If the check succeeds, go to next step; otherwise, abort.
2. Based on the auxiliary input $(i', \mathsf{f}(R_{i'}, X')))$ from the hypothetical oracle $\mathcal{O}$, $\mathsf{simulator}_1$ performs as follows.
   - If $\mathsf{f}(R_{i'}, X') = 1$, set $C_{i'} = (g^{s_1}, y^{s_1})$ where s_1 is randomly chosen from $\mathbb{Z}_q$, and for every $1 \leq j \leq N$ and $j \neq i'$, set $C_j = (\beta_{j1}, \beta_{j2})$ where β_{j1}, β_{j2} are randomly chosen from $\mathbb{G}$.
   - Otherwise, for every $1 \leq j \leq N$, set $C_j = (\beta_{j1}, \beta_{j2})$ where β_{j1}, β_{j2} are randomly chosen from $\mathbb{G}$.
3. Faithfully execute the PIR protocol with $\mathcal{U}$.

For the EPIR protocol for equality, the hypothetical oracle $\mathcal{O}$ generates its input to $\mathsf{simulator}_1$ as follows.

1. Compute $i||X$ satisfying $(\alpha_1, \alpha_2) = \mathsf{Enc}(g^{i||X}, pk)$.
2. If $i \notin \{1, 2, \cdots, N\}$, set the input to be $(1, 0)$. Otherwise, set the input to be $(i, \mathsf{f}(R_i, X))$.

It is clear that the only difference between $\mathsf{simulator}_0$ and $\mathsf{simulator}_1$ in answering $\mathcal{U}$'s query lies in the computation of C_j $(1 \leq j \leq N)$. From Observation 1, the distributions of C_j $(1 \leq j \leq N)$ are identical for $\mathsf{simulator}_0$ and $\mathsf{simulator}_1$, therefore, $\mathcal{A}$ can only have advantage 0. The lemma now follows. □

4 EPIR protocol for computing Hamming distance

In this section we present an EPIR protocol which enables $\mathcal{U}$ to compute Hamming distance between a string chosen by itself and a block from $\mathcal{DB}$. Especially, the protocol allows the user to assign a weight for every bit. For an ℓ_1-bit string S, let $S^{(k)}$ denote the k-th bit of S. Let the weight vector be $(w_1, w_2, \cdots, w_{\ell_1})$ where w_k $(1 \leq k \leq \ell_1)$ are integers. The function f is defined as follows.

$$\mathsf{f}(R_i, X) = \sum_{k=1}^{\ell_1} w_k (R_i^{(k)} \oplus X^{(k)})$$

The construction is based on the BGN encryption scheme [3], the GOS NIZK protocol [27] (described in Appendix D), and a PIR protocol. It is worth noting that, due to the randomization in step 3, the employed PIR protocol does not need to be SPIR (achieving database privacy) in order to guarantee the database privacy for the EPIR.

4.1 Description of the protocol

Suppose every block in $\mathcal{DB}$ has bit-length ℓ_1. The EPIR protocol is as follows.

1. $\mathcal{U}$ generates a key pair (pk, sk) for the BGN encryption scheme, where $pk = (n, \mathbb{G}, \mathbb{G}_1, \hat{e}, g, h)$, and $sk = q_1$.
2. To retrieve the value of $\mathsf{f}(R_i, X)$, for any $1 \le i \le N$ and $X \in \{0,1\}^{\ell_1}$, $\mathcal{U}$ first sends BGN ciphertexts c and c_k $(1 \le k \le \ell_1)$ to $\mathcal{DB}$, where $c = g^i h^r$, $c_k = g^{X^{(k)}} h^{s_k}$ $(1 \le k \le \ell_1)$, r and s_k $(1 \le k \le \ell_1)$ are randomly chosen from $\mathbb{Z}_n$. In addition, $\mathcal{U}$ also sends $proof_k$ $(1 \le k \le \ell_1)$ to $\mathcal{DB}$, where, for every $1 \le k \le \ell_1$, $proof_k$ is the GOS NIZK parameter for proving $X^{(k)} \in \{0,1\}$.
3. After receiving c, c_k $(1 \le k \le \ell_1)$, and $proof_k$ $(1 \le k \le \ell_1)$ from $\mathcal{U}$, $\mathcal{DB}$ first checks that pk is a valid BGN public key[3] and c, c_k $(1 \le k \le \ell_1)$ are valid BGN ciphertexts. If the check succeeds, $\mathcal{DB}$ verifies $proof_k$ $(1 \le k \le \ell_1)$. If the verification succeeds, $\mathcal{DB}$ computes C_j for every $1 \le j \le N$ as follows.
   (a) For every $1 \le k \le \ell_1$, compute $m_{j,k}$ where

$$\begin{aligned} m_{j,k} &= \frac{\hat{e}(c_k g^{R_j^{(k)}}, g)}{\hat{e}(c_k, g^{R_j^{(k)}})^2} \\ &= \frac{\hat{e}(g^{X^{(k)}} h^{s_k} g^{R_j^{(k)}}, g)}{\hat{e}(g^{X^{(k)}} h^{s_k}, g^{R_j^{(k)}})^2} \\ &= \frac{\hat{e}(g^{X^{(k)}} g^{R_j^{(k)}}, g)\hat{e}(h^{s_k}, g)}{\hat{e}(g^{X^{(k)}}, g^{R_j^{(k)}})^2 \hat{e}(h^{s_k}, g^{R_j^{(k)}})^2} \\ &= \hat{e}(g,g)^{X^{(k)} + R_j^{(k)} - 2X^{(k)}R_j^{(k)}} \hat{e}(h,g)^{s_k(1-2R_j^{(k)})} \\ &= \hat{e}(g,g)^{X^{(k)} \oplus R_j^{(k)}} \hat{e}(h,g)^{s_k(1-2R_j^{(k)})} \end{aligned}$$

   (b) Compute C_j, where r_j, r'_j are randomly chosen from $\mathbb{Z}_n$ and

$$\begin{aligned} C_j &= \hat{e}(cg^{-j}h^{r'_j}, g)^{r_j} \prod_{k=1}^{\ell_1} (m_{j,k})^{w_k} \\ &= \hat{e}(g^{i-j}h^{r+r'_j}, g)^{r_j} \prod_{k=1}^{\ell_1} \hat{e}(g,g)^{w_k(X^{(k)} \oplus R_j^{(k)})} \hat{e}(h,g)^{w_k s_k (1-2R_j^{(k)})} \\ &= \hat{e}(g,g)^{r_j(i-j) + \sum_{k=1}^{\ell_1} w_k (X^{(k)} \oplus R_j^{(k)})} \hat{e}(h,g)^{r_j(r+r'_j) + \sum_{k=1}^{\ell_1} w_k s_k (1-2R_j^{(k)})} \end{aligned}$$

   Otherwise, $\mathcal{DB}$ aborts the protocol execution.
4. $\mathcal{U}$ runs a PIR protocol to retrieve C_i from $\mathcal{DB}$, and sets $\mathsf{f}(i, X) = d$ if $C_i^{q_1} = \hat{e}(g^{q_1}, g)^d$.

As to the performance, the communication complexity is dominated by that of the PIR protocol and the transmission of $c_k, proof_k$ $(1 \le k \le \ell_1)$. For $\mathcal{U}$, the computational complexity is dominated by generating $c_k, proof_k$ $(1 \le k \le \ell_1)$: $O(\ell_1)$ exponentiations. For $\mathcal{DB}$, the computational complexity is dominated by checking the GOS NIZK proofs and the computation of C_j $(1 \le j \le N)$: $O(N + \ell_1)$ pairing computations and $O(N)$ exponentiations. Moreover, it is straightforward to verify the following observation.

[3] In practice, the validity of pk can be certified by a TTP, and the same pk can be used by the user for all his queries.

Observation 2 *For every $1 \leq j \leq N$, given that $i \neq j$, the components of $C_j = (C_{j1}, C_{j2})$, where*

$$C_{j1} = \hat{e}(g,g)^{r_j(i-j)+\sum_{k=1}^{\ell_1} w_k(X^{(k)} \oplus R_j^{(k)})},\ C_{j2} = \hat{e}(h,g)^{r_j(r+r'_j)+\sum_{k=1}^{\ell_1} w_k s_k(1-2R_j^{(k)})},$$

are uniformly and independently distributed over $\mathbb{G}_1$ and the subgroup of order q_1 of $\mathbb{G}_1$, respectively. If $i = j$, then $C_{j1} = \hat{e}(g,g)^{\sum_{k=1}^{\ell_1} w_k(X^{(k)} \oplus R_j^{(k)})}$ and C_{j2} is uniformly distributed over the subgroup of order q_1 of $\mathbb{G}_1$.

4.2 Security Analysis

It is straightforward to verify that if the PIR protocol is sound then the EPIR protocol is also sound. First, we prove the following lemma.

Lemma 5. *Given any $M \geq 1$, the attacker's advantage in the following game is negligible for the BGN encryption scheme.*

$\mathbf{Exp}_{\mathcal{A}}^{P\text{-}IND\text{-}CPA}$

$$\begin{array}{rl}
(pk, sk) & \leftarrow \mathsf{Gen}(1^\ell) \\
((m_{0,1}, \ldots, m_{0,M}), (m_{1,1}, \ldots, m_{1,M})) & \leftarrow \mathcal{A}_1(pk) \\
b & \leftarrow \{0,1\} \\
\mathbf{c} & \leftarrow (\mathsf{Enc}(m_{b,1}, pk), \ldots, \mathsf{Enc}(m_{b,M}, pk)) \\
b' & \leftarrow \mathcal{A}_2(\mathbf{c})
\end{array}$$

Proof. We only prove the case of $M = 2$ since a general result can be obtained by an induction on M. Suppose the attacker $\mathcal{A}$ succeeds in guessing b' with probability δ, then we construct an attacker $\mathcal{A}'$ for the BGN scheme as follows.

1. $\mathcal{A}'$ receives pk from the BGN challenger.
2. $\mathcal{A}'$ runs $\mathcal{A}_1$ with input pk.
3. After receiving $(m_{0,1}, m_{0,2})$ and $(m_{1,1}, m_{1,2})$ from $\mathcal{A}_1$, $\mathcal{A}'$ submits $m_{0,d}$ and $m_{1,d}$ to the BGN challenger for a challenge, where d is randomly chosen from $\{1,2\}$.
4. If $d = 1$, $\mathcal{A}'$ runs $\mathcal{A}_2$ with input $(c_{b,1}, c_{e,2})$, where $c_{b,1}$ is the BGN challenge, e is a random coin toss of $\mathcal{A}'$, and $c_{e,2} = \mathsf{Enc}(m_{e,2}, pk)$. If $d = 2$, $\mathcal{A}'$ runs $\mathcal{A}_2$ with input $(c_{e,1}, c_{b,2})$, where $c_{b,2}$ is the BGN challenge, e is a random coin toss of $\mathcal{A}'$, and $c_{e,1} = \mathsf{Enc}(m_{e,1}, pk)$.
5. After receiving b' from $\mathcal{A}_2$, $\mathcal{A}'$ outputs its guess b'.

We first discuss the probability that $b' = b$ when $d = 1$ and $d = 2$.

- Case $d = 1$: Let E_1 be the event that $b = e$. It is clear that $\Pr[E_1] = \frac{1}{2}$. If E_1 occurs, we have $\Pr[b = b'|E_1] = \delta$ and $\Pr[e = b'|E_1] = \delta$ since $\mathcal{A}'$ faithfully simulates the attack game for $\mathcal{A}$. Otherwise, let $\Pr[b = b'|\neg E_1] = \frac{1}{2} + \epsilon_1$ and $\Pr[e = b'|\neg E_1] = \frac{1}{2} - \epsilon_1$. In this case, the probability that $b' = b$ is $\frac{1}{2}\delta + \frac{1}{4} + \frac{1}{2}\epsilon_1$, namely $\Pr[b' = b|d = 1] = \frac{1}{2}\delta + \frac{1}{4} + \frac{1}{2}\epsilon_1$.
- Case $d = 2$: Let E_2 be the event that $b = e$. It is clear that $\Pr[E_2] = \frac{1}{2}$. If E_2 occurs, we have $\Pr[b = b'|E_2] = \delta$ and $\Pr[e = b'|E_2] = \delta$ since $\mathcal{A}'$ faithfully simulates the attack game for $\mathcal{A}$. Otherwise, let $\Pr[b = b'|\neg E_2] = \frac{1}{2} + \epsilon_2$ and $\Pr[e = b'|\neg E_2] = \frac{1}{2} - \epsilon_2$. In this case, the probability that $b' = b$ is $\frac{1}{2}\delta + \frac{1}{4} + \frac{1}{2}\epsilon_2$, namely $\Pr[b' = b|d = 2] = \frac{1}{2}\delta + \frac{1}{4} + \frac{1}{2}\epsilon_2$.

The overall probability that $\Pr[b' = b]$ is

$$\begin{aligned}
\Pr[b' = b] &= \frac{1}{2}(\Pr[b' = b|d = 1] + \Pr[b' = b|d = 2]) \\
&= \frac{1}{2}\delta + \frac{1}{4} + \frac{1}{4}(\epsilon_1 + \epsilon_2)
\end{aligned}$$

From the description of $\mathcal{A}'$, it is clear that the following observation is true.

Observation 3 *The game simulations for $\mathcal{A}$ are identical when $d = 1$ and $d = 2$, therefore,* $\Pr[b = b'|\neg E_1] = \Pr[e = b'|\neg E_2]$ *so that* $\epsilon_1 = -\epsilon_2$.

As a result, $\mathcal{A}'$ has the advantage $|\Pr[b' = b] - \frac{1}{2}| = \frac{1}{2}|\delta - \frac{1}{2}|$. If $\mathcal{A}$ wins the game with a non-negligible advantage (or $|\delta - \frac{1}{2}|$ is non-negligible), then we get a contradiction with the semantic security of BGN scheme (or the subgroup decision assumption). The lemma now follows. □

From Lemma 3 and Lemma 5, we immediately have the following lemma because the protocol for testing equality shares the same structure as the protocol for computing Hamming distance. Because the GOS NIZK proof protocol is perfectly sound, the proof of this lemma is similar to Lemma 3, therefore, we omit it here.

Lemma 6 (user privacy). *If the PIR protocol achieves user privacy, the EPIR protocol for computing Hamming distance achieves user privacy based on the subgroup decision assumption.*

Lemma 7 (database privacy). *The EPIR protocol for computing Hamming distance achieves database privacy (unconditionally).*

Proof. Recall that, in the attack game definition for database privacy (as described in Section 2.3), $\mathsf{simulator}_0$ is assumed to be $\mathcal{DB}$. We now define a simulator $\mathsf{simulator}_1$ which answers the attacker's query as follows.

1. On receiving pk, c, and c_k $(1 \le k \le \ell_1)$, $\mathsf{simulator}_1$ first checks that they are well formed. If the check succeeds, go to next step; otherwise, abort.
2. Based on the auxiliary input $(i', \mathsf{f}(R_{i'}, X')))$ from the hypothetical oracle $\mathcal{O}$, $\mathsf{simulator}_1$ performs as follows. For every $1 \le j \le N$ and $j \ne i'$, set $C_j = \hat{e}(cg^{-j}, g)^{r_j}\hat{e}(h, g)^{r'_j}$ where r_j, r'_j are randomly chosen from $\mathbb{Z}_n$. For $j = i'$, set $C_{i'}$ to be
$$\begin{aligned} C_{i'} &= \hat{e}(cg^{-i'}, g)^{r_{i'}}\hat{e}(g, g)^{\mathsf{f}(R_{i'}, X')}\hat{e}(h, g)^{r'_{i'}} \\ &= \hat{e}(g, g)^{r_{i'}(i-i')+\mathsf{f}(R_{i'}, X')}\hat{e}(h, g)^{rr_{i'}+r'_{i'}} \end{aligned}$$
where $c = g^i h^r$ and $r_{i'}, r'_{i'}$ are randomly chosen from $\mathbb{Z}_n$.
3. Faithfully execute the PIR protocol with $\mathcal{U}$.

For the EPIR protocol for computing Hamming distance, the hypothetical oracle $\mathcal{O}$ generates its input to $\mathsf{simulator}_1$ as follows.

1. Compute $i = \mathsf{Dec}(c, sk)$ and $X^{(k)} = \mathsf{Dec}(c_k, sk)$ $(1 \le k \le \ell_1)$.
2. If $i \notin \{1, 2, \cdots, N\}$, set the input to be $(1, 0)$. Otherwise, set the input to be $(i, \mathsf{f}(R_i, X)))$.

It is clear that the only difference between $\mathsf{simulator}_0$ and $\mathsf{simulator}_1$ in answering $\mathcal{U}$'s query lies in the computation of C_j $(1 \le j \le N)$. From observation 2, the distributions of C_j $(1 \le j \le N)$ are identical for $\mathsf{simulator}_0$ and $\mathsf{simulator}_1$, therefore, $\mathcal{A}$ can only have advantage 0. The lemma now follows. □

5 Authentication Schemes using Biometrics

5.1 Preliminaries

In our security model, besides human users, we assume that a biometric-based (remote) authentication system consists of the following types of components:

– Authentication client $\mathcal{C}$, which is responsible for extracting human user's biometric template using some biometric sensor and communicating with authentication server.

- Authentication server $\mathcal{S}$, which is responsible for dealing with the human user's authentication requests by querying the database which stores user's biometric template.
- Centralized database $\mathcal{DB}$, which stores the relevant biometric information for authentication[4].

Like most existing biometric-based systems (and many traditional cryptosystems), a biometric-based authentication scheme consists of two phases: an enrollment phase and a verification phase.

1. In the enrollment phase, user U_i registers his biometric template b_i at the database $\mathcal{DB}$ and his identity information ID_i at the authentication server $\mathcal{S}$.
2. In the verification phase, user U_i issues an authentication request to the authentication server $\mathcal{S}$ through a client $\mathcal{C}$. The authentication server $\mathcal{S}$ retrieves U_i's biometric information from the database $\mathcal{DB}$ and makes a decision.

Human users and $\mathcal{S}$ trust $\mathcal{C}$ to be honest, and $\mathcal{S}$ trusts $\mathcal{DB}$ to provide the correct biometric information. We further make the following assumptions on the system components: The communication links between any two components are authenticated and encrypted. In practice, the security links can be implemented using a standard protocol such as SSL or TLS. In addition, the following assumptions are indispensable for all biometrics-based systems.

1. Biometric Distribution assumption: Let H be the distance function in the Hamming space. We assume that, there is a threshold value λ, the probability that $\mathsf{H}(b_i, b_j) > \lambda$ is close to 1[5], where b_i is Alice's biometric template and b_j is Bob's biometric template, while the probability that $\mathsf{H}(b_i, b_i') \leq \lambda$ is close to 1, where b_i and b_i' are Alice's biometric templates in two measurements.
2. Liveness assumption: We assume that, with a high probability, the biometric template captured by the sensor is from a live human user. In other words, it is difficult to produce a faked biometric template that can be accepted by the sensor.

However, how to achieve these properties is beyond the scope of this paper.

For a biometric-based authentication scheme, two types of security properties are mainly concerned. One is the resistance to impersonation attacks, in which case we only consider outside adversaries by assuming that all the system components are honest. The other is preserving privacy properties, in which case we only consider malicious inside adversaries including a malicious $\mathcal{S}$ and a malicious $\mathcal{DB}$. But we assume that $\mathcal{S}$ and $\mathcal{DB}$ will not collude. In practice, many methods (for example, issuing a smart-card to every user) can be used to guarantee these properties against other kinds of adversaries, but we omit them in this paper since our main aim is to demonstrate the application of the EPIR protocols.

5.2 The First Biometric-based Authentication Scheme

This biometric-based authentication scheme is constructed based on the EPIR protocol for equality as described in Section 3.1. In this scheme, due to the secure sketch scheme, the user does not need to store any private information and the client $\mathcal{C}$ does not need to store any user specific information. The enrollment phase works as follows.

- $\mathcal{C}$ implements a (m, m', λ)-secure sketch $(\mathsf{SS}, \mathsf{Rec})$ (an example is described in Appendix C),where m' is the system security parameter.

[4] It is worth emphasizing that $\mathcal{DB}$ and $\mathcal{S}$ are two different principles and $\mathcal{DB}$ may serve as a trusted storage for a number of authentication servers. This is different from the conventional environment where we say a server has its own database for storing the authentication secrets.

[5] Note that this probability is related to the false accept and false reject rates of biometrics, but we omit a detailed discussion in this paper.

- $\mathcal{S}$ generates an ElGamal key pair (pk, sk).
- U_i generates his unique pseudorandom identifier ID_i and registers it at the server $\mathcal{S}$, and registers (ID_i, R_i) at the database $\mathcal{DB}$, where b_i is U_i's reference biometric template and

$$\begin{aligned} R_i &= \mathsf{Enc}(g^{ID_i||b_i}, pk) \\ &= (R_{i1}, R_{i2}). \end{aligned}$$

  In addition, U_i publicly stores a sketch $sketch_i = \mathsf{SS}(b_i)$.

If U_i wants to authenticate himself to the server $\mathcal{S}$ through the authentication client $\mathcal{C}$, then the procedure is as follows.

1. $\mathcal{C}$ extracts U_i's biometric template b_i^* and computes the adjusted template $b_i' = \mathsf{Rec}(b_i^*, sketch_i)$. Then $\mathcal{C}$ sends ID_i to S and sends X to $\mathcal{DB}$, where $X = \mathsf{Enc}(g^{ID_i||b_i'}, pk)$. Otherwise, $\mathcal{C}$ aborts the operation.
2. After receiving X, $\mathcal{DB}$ performs as in the EPIR protocol for equality as described in Section 3.1, where $\mathcal{DB}$ computes C_j for every $1 \leq j \leq N$, where r_j, r_j' are randomly chosen from $\mathbb{Z}_q$ and

$$C_j = (g^{r_j'}(g^r(R_{i1})^{-1})^{r_j}, y^{r_j'}(y^r g^{ID_i||X}(R_{i2})^{-1})^{r_j})$$

3. The server runs a PIR to retrieve C_i. If $\mathsf{Dec}(C_i, sk) = 1$, $\mathcal{S}$ accepts the request; otherwise rejects it.

It is easy to verify that impersonation attacks are prevented based on the biometric distribution assumption, i.e. an adversary can not force $\mathcal{C}$ to output U_j's template by letting $\mathcal{C}$ measure $U_i's$ biometric, given that U_i and U_j are different human users.

Every authentication is indeed an execution of the EPIR protocol for equality between $\mathcal{S}$ and $\mathcal{DB}$, though X is sent to $\mathcal{DB}$ by a trusted $\mathcal{C}$. From the user privacy property of the EPIR protocol, $\mathcal{DB}$ learns nothing about which user is authenticating himself and what is the authentication result. In addition, $\mathcal{DB}$ obtains nothing about the registered biometric templates because they are encrypted by $\mathcal{S}$'s public key. From the database privacy property of the EPIR protocol, $\mathcal{S}$ learns nothing about a user's biometric template. In fact, $\mathcal{S}$ only obtains the information whether the authentication request is made by the legitimate user or not.

5.3 The Second Biometric-based Authentication Scheme

This biometric-based authentication scheme is constructed based on the EPIR protocol for computing Hamming distance as described in Section 4.1. In this scheme, the user does not need to store any private or public information and the client $\mathcal{C}$ does not need to store any user specific information. The server $\mathcal{S}$ is enabled to make its decision based on an exact matching between a user's biometric templates. The overall matching result can be more accurate by allocating a score (or a weight) for the matching result of every single bit. The enrollment phase works as follows.

- $\mathcal{S}$ generates a BGN encryption key pair (pk, sk).
- U_i generates his pseudorandom identifier ID_i and registers it at the server $\mathcal{S}$, and registers $(ID_i, \alpha_i^{(k)}\ (1 \leq k \leq \ell_1))$ at the database $\mathcal{DB}$, where b_i is U_i's reference biometric template with bit-length ℓ_1, $\alpha_i^{(k)} = g^{b_i^{(k)}} h^{\beta_{ik}}\ (1 \leq k \leq \ell_1)$, and $\beta_{ik}\ (1 \leq k \leq \ell_1)$ are randomly chosen from $\mathbb{Z}_n$.

If U_i wants to authenticate himself to the server $\mathcal{S}$ through the authentication client $\mathcal{C}$, then the procedure is as follows.

1. $\mathcal{C}$ extracts U_i's biometric template b_i' and sends c and c_k $(1 \le k \le \ell_1)$ to $\mathcal{DB}$, where $c = g^{ID_i} h^r$, $c_k = g^{b_i'^{(k)}} h^{s_k}$ $(1 \le k \le \ell_1)$, r and s_k $(1 \le k \le \ell_1)$ are randomly chosen from $\mathbb{Z}_n$. Simultaneously, $\mathcal{C}$ sends ID_i to $\mathcal{S}$.
2. After receiving c and c_k $(1 \le k \le \ell_1)$, $\mathcal{DB}$ performs in a similar way as in the EPIR protocol for computing Hamming distance except that it computes C_j for every $1 \le j \le N$ as follows.
   (a) For every $1 \le k \le \ell_1$, compute $m_{j,k}$ where

$$\begin{aligned}
m_{j,k} &= \frac{\hat{e}(c_k \alpha_j^{(k)}, g)}{\hat{e}(c_k, \alpha_j^{(k)})^2} \\
&= \frac{\hat{e}(c_k g^{b_j^{(k)}} h^{\beta_{jk}}, g)}{\hat{e}(c_k, g^{b_j^{(k)}} h^{\beta_{jk}})^2} \\
&= \frac{\hat{e}(g^{b_i'^{(k)}} h^{s_k+\beta_{jk}} g^{b_j^{(k)}}, g)}{\hat{e}(g^{b_i'^{(k)}} h^{s_k}, g^{b_j^{(k)}} h^{\beta_{jk}})^2} \\
&= \frac{\hat{e}(g^{b_i'^{(k)}} g^{b_j^{(k)}}, g)\hat{e}(h^{s_k+\beta_{jk}}, g)}{\hat{e}(g^{b_i'^{(k)}}, g^{b_j^{(k)}})^2 \hat{e}(h,g)^{2(s_k b_j^{(k)} + b_i'^{(k)}\beta_{jk} + s_k \beta_{jk} \log_g h)}} \\
&= \hat{e}(g,g)^{b_i'^{(k)} + b_j^{(k)} - 2b_i'^{(k)} b_j^{(k)}} \hat{e}(h,g)^{s_k(1-2\beta_{jk}\log_g h - 2b_j^{(k)}) + \beta_{jk}(1-2b_i'^{(k)})} \\
&= \hat{e}(g,g)^{b_i'^{(k)} \oplus R_j^{(k)}} \hat{e}(h,g)^{s_k(1-2\beta_{jk}\log_g h - 2b_j^{(k)}) + \beta_{jk}(1-2b_i'^{(k)})}
\end{aligned}$$

   (b) Let $x_{jk} = s_k(1 - 2\beta_{jk} \log_g h - 2b_j^{(k)}) + \beta_{jk}(1 - 2b_i'^{(k)})$ $(1 \le k \le \ell_1)$, compute C_j, where r_j, r_j' are randomly chosen from $\mathbb{Z}_n$ and

$$\begin{aligned}
C_j &= \hat{e}(c g^{-ID_j} h^{r_j'}, g)^{r_j} \prod_{k=1}^{\ell_1} (m_{j,k})^{w_k} \\
&= \hat{e}(g^{ID_i - ID_j} h^{r + r_j'}, g)^{r_j} \prod_{k=1}^{\ell_1} \hat{e}(g,g)^{w_k (b_i'^{(k)} \oplus b_j^{(k)})} \hat{e}(h,g)^{w_k x_{jk}} \\
&= \hat{e}(g,g)^{r_j(ID_i - ID_j) + \sum_{k=1}^{\ell_1} w_k (b_i'^{(k)} \oplus b_j^{(k)})} \hat{e}(h,g)^{r_j(r + r_j') + \sum_{k=1}^{\ell_1} w_k x_{jk}}
\end{aligned}$$

3. $\mathcal{S}$ runs a PIR to retrieve C_i, and computes d satisfying $C_i^{q_1} = \hat{e}(g^{q_1}, g)^d$. $\mathcal{S}$ accepts the request if d is smaller than a threshold value; otherwise rejects it.

We first emphasize that the GOS NIZK proofs are omitted in this authentication scheme because c and c_k $(1 \le k \le \ell_1)$ are sent by $\mathcal{C}$ which is trusted by all parties.

It is easy to verify that impersonation attacks are prevented based on the biometric distribution assumption. Every authentication is indeed an execution of the EPIR protocol for computing Hamming distance between $\mathcal{S}$ and $\mathcal{DB}$, though we have made some small modifications. As a result, this scheme achieves the same security properties as those of the previous scheme.

Compared with the previous scheme, this scheme is more convenient for human users and the the client $\mathcal{C}$, where a human user does not need to store any information and secure sketch is not needed to be implemented in $\mathcal{C}$. Another advantage of this protocol is that it works even when secure sketches are not practical (i.e. when noise is high).

6 Conclusion

In this paper we formulated the concept of EPIR and proposed two protocols: one for testing equality and the other for computing Hamming distance. The randomizations in both protocols are performed to avoid using a SPIR protocol in order to achieve the privacy for the database. In addition, the randomizations also guarantee that the privacy for the database is unconditionally achieved (without any computational assumption). It is a challenging task to design more efficient EPIR protocols, especially to reduce the computational complexity. In this paper, we also showed how to construct strong privacy-preserving biometric-based authentication schemes by employing these EPIR protocols. Some further work is required to evaluate the performance of these schemes in practice.

References

1. M. J. Atallah, K. B. Frikken, M. T. Goodrich, and R. Tamassia. Secure biometric authentication for weak computational devices. In *Financial Cryptography*, pages 357–371, 2005.
2. R. M. Bolle, J. H. Connell, and N. K. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.
3. D. Boneh, E. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In J. Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.
4. X. Boyen. Reusable cryptographic fuzzy extractors. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 82–91. ACM Press, 2004.
5. X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In R. Cramer, editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 147–163. Springer, 2005.
6. J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer. An application of the Goldwasser-Micali cryptosystem to biometric authentication. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007 Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer, 2007.
7. C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer, 1999.
8. R. Canetti, Y. Ishai, R. Kumar, M. K. Reiter, R. Rubinfeld, and R. N. Wright. Selective private function evaluation with applications to private statistics. In *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 293–304. ACM Press, 2001.
9. B. Chor and N. Gilboa. Computationally private information retrieval (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 304–313, 1997.
10. B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.
11. C. Crepeau and L. Salvail. Oblivious verification of common string. *CWI Quarterly , special issue for Crypto Course 10th Anniversary*, 8(2):97–109, 1995.
12. G. D. Crescenzo, R. Graveman, R. Ge, and G. Arce. Approximate message authentication and biometric entity authentication. In A. S. Patrick and M. Yung, editors, *Financial Cryptography and Data Security, 9th International Conference*, volume 3570 of *Lecture Notes in Computer Science*, pages 240–254. Springer, 2005.
13. G. D. Crescenzo, T. Malkin, and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. In B. Prenell, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 122–138. Springer, 2000.
14. Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In C. Dwork, editor, *Advances in Cryptology — CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer, 2006.
15. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.
16. W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In *ACSAC '01: Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–110. IEEE Computer Society, 2001.
17. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *NSPW '01: Proceedings of the 2001 workshop on New security paradigms*, pages 13–22. ACM Press, 2001.
18. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and D. Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1985.

19. Ronald Fagin, Moni Naor, and Peter Winkler. Comparing information without leaking it. *Communications of the ACM*, 39(5):77–85, 1996.
20. M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In J. kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 303–324. Springer, 2005.
21. M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.
22. W. Gasarch. A survey on private information retrieval. http://www.cs.umd.edu/ gasarch/pir/pir.html.
23. C. Gentry and Z. Ramzan. Single-database private information retrieval with constant communication rate. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815. Springer, 2005.
24. Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 151–160, 1998.
25. B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen. On private scalar product computation for privacy-preserving data mining. In *Information Security and Cryptology — ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 104–120. Springer, 2004.
26. O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.
27. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *Advances in Cryptology — EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer, 2006.
28. F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.
29. J. D. Woodward Jr., N. M. Orlans, and P. T. Higgins. *Biometrics (Paperback)*. McGraw-Hill/OsborneMedia, 2002.
30. A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.
31. A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.
32. E. Kiltz, G. Leander, and J. Malone-Lee. Secure computation of the mean and related statistics. In J. Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 283–302. Springer, 2005.
33. E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: Single database, computationally-private information retrieval. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97*, pages 364–373, 1997.
34. J. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In J. Kittler and M. S. Nixon, editors, *Audio-and Video-Based Biometrie Person Authentication, 4th International Conference*, volume 2688 of *Lecture Notes in Computer Science*, pages 393–402. Springer, 2003.
35. H. Lipmaa. An oblivious transfer protocol with log-squared communication. In J. Zhou, J. Lopez, R. H. Deng, and F. Bao, editors, *Information Security, 8th International Conference, ISC 2005*, volume 3650 of *Lecture Notes in Computer Science*, pages 314–328. Springer, 2005.
36. D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer, 2003.
37. S. K. Mishra and P. Sarkar. Symmetrically private information retrieval. In B. K. Roy and E. Okamoto, editors, *Progress in Cryptology — INDOCRYPT 2000*, volume 1977 of *Lecture Notes in Computer Science*, pages 225–236. Springer, 2000.
38. M. Naor and B. Pinkas. Oblivious polynomial evaluation. *SIAM J. Comput.*, 35(5):1254–1281, 2006.
39. R. Ostrovsky and W. E. Skeith III. A survey of single database PIR: Techniques and applications. Cryptology ePrint Archive: Report 2007/059, 2007.
40. N. Ratha, J. Connell, R. M. Bolle, and S. Chikkerur. Cancelable biometrics: A case study in fingerprints. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 370–373. IEEE Computer Society, 2006.
41. N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.
42. R. Safavi-Naini and D. Tonien. Fuzzy universal hashing and approximate authentication. Cryptology ePrint Archive: Report 2005/256, 2005.
43. B. Schoenmakers and P. Tuyls. Efficient binary conversion for Paillier encrypted values. In S. Vaudenay, editor, *Advances in Cryptology — EUROCRYPT '06*, volume 4004 of *Lecture Notes in Computer Science*, pages 522–537. Springer, 2006.
44. P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G. Jan Schrijen, A. M. Bazen, and R. N. J. Veldhuis. Practical biometric authentication with template protection. In T. Kanade, A. K. Jain, and N. K. Ratha, editors, *Audio- and Video-Based Biometric Person Authentication, 5th International Conference*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer, 2005.
45. P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In *ECCV Workshop BioAW*, pages 158–170, 2004.
46. P. Tuyls, E. Verbitskiy, J. Goseling, and D. Denteneer. Privacy protecting biometric authentication systems: an overview. In *EUSIPCO 2004*, 2004.

47. U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.
48. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 639–644, 2002.
49. E. Verbitskiy, P. Tuyls, D. Denteneer, and J. P. Linnartz. Reliable biometric authentication with privacy protection. In *SPIE Biometric Technology for Human Identification Conf.*, 2004.
50. M. J. Atallah W. Du. Protocols for secure remote database access with approximate matching. Technical report, CERIAS, Purdue University, 2000. CERIAS TR 2000-15.
51. A. Yao. Protocols for secure computations. In *Proceedings of the twenty-third annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

Appendix A: Introduction to the ElGamal encryption scheme

The algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of the ElGamal public key encryption scheme [18] are defined as follows:

1. The key generation algorithm Gen takes a security parameter 1^k as input and generates two primes p, q satisfying $q|p-1$. Let $\mathbb{G}$ be the subgroup of order q in $\mathbb{Z}_p^*$, g be a generator of $\mathbb{G}$. The private key x which is randomly chosen from $\mathbb{Z}_q$, and the public key is $y = g^x$. Let Ω be a bijective map from $\mathbb{Z}_q$ to $\mathbb{G}$.
2. The encryption algorithm Enc takes a message m and the public key y as input, and outputs the ciphertext $c = (c_1, c_2) = (g^r, y^r \Omega(m))$ where r is randomly chosen from $\mathbb{Z}_q^*$.
3. The decryption algorithm Dec takes a ciphertext $c = (c_1, c_2)$ and the private key x as input, and outputs the message $m = \Omega^{-1}((c_1^{-x} c_2)$.

It is well-known that the ElGamal scheme is semantically secure based on the DDH assumption. In other words, an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has only a negligible advantage in the following game.

$$
\begin{array}{l}
\mathbf{Exp}^{\text{IND-CPA}}_{\mathcal{E},\mathcal{A}} \\
\left|\begin{array}{rl}
(sk, pk) & \leftarrow \mathsf{Gen}(1^k) \\
(m_0, m_1) & \leftarrow \mathcal{A}_1(pk) \\
b & \stackrel{R}{\leftarrow} \{0,1\} \\
c & \leftarrow \mathsf{Enc}(m_b, pk) \\
b' & \leftarrow \mathcal{A}_2(m_0, m_1, c, pk)
\end{array}\right. \\
\text{return } b'
\end{array}
$$

At the end of this game, the attacker's advantage is defined to be $|\Pr[b' = b] - \frac{1}{2}|$.

Appendix B: Introduction to the BGN Scheme

The algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ of the BGN encryption scheme [3] are defined as follows:

1. The key generation algorithm Gen takes a security parameter 1^k as input and generates a tuple $(n, q_1, q_2, \mathbb{G}, \mathbb{G}_1, \hat{e}, g, u, h)$, where q_1 and q_2 are two primes, $n = q_1 q_2$, $\mathbb{G}$ and $\mathbb{G}_1$ are two cyclic groups of order n, g and u are generators of $\mathbb{G}$, and $h = u^{q_2}$. The private key $sk = q_1$, and the public key is $pk = (n, \mathbb{G}, \mathbb{G}_1, \hat{e}, g, h)$.
2. The encryption algorithm Enc takes a message $m \in \mathbb{Z}_{q_2}$ and the public key pk as input, and outputs the ciphertext $c = g^m h^r$ where r is randomly chosen from $\mathbb{Z}_n$.
3. The decryption algorithm Dec takes a ciphertext c and the private key sk as input, and outputs the message $c^{q_1} = (g^{q_1})^m$. Then compute the discrete log of c^{q_1} base g^{q_1}.

It is proved by Boneh, Goh, and Nissim that this scheme is semantically secure given the subgroup decision problem is hard for $(n, \mathbb{G}, \mathbb{G}_1, \hat{e})$.

Appendix C: Introduction to Secure Sketches

Roughly speaking, a secure sketch scheme $(\mathsf{SS}, \mathsf{Rec})$ allows recovery of a hidden value from any value close to this hidden value. Informally, the algorithm SS take a value x as input and outputs some public value y, and the algorithm Rec takes a value x' and y as input and outputs a value x''. If x' and x are close enough, then $x'' = x$.

We take the Code-Offset Construction given in [15] as an example. let C be a $[n, k, 2t+1]$ error-correction code over a field $\mathbb{F}$. With input $x \in \mathbb{F}^n$, y is computed as $\mathsf{SS}(x) = x - c$, where c is a random codeword. With input (x', y), Rec computes x'' in the following way: compute $c' = x' - y$, decode c' to obtain c'', and set $x'' = c'' + y$.

Appendix D: Introduction to the GOS NIZK protocol

The following is the Non-Interactive Zero Knowledge (NIZK) protocol of Groth, Ostrovsky, and Sahai [27]. It is shown to possess perfect completeness, perfect soundness, computational zero-knowledge, and honest prover state reconstruction.

- Common reference string:
  1. $(n, q_1, q_2, \mathbb{G}, \mathbb{G}_1, \hat{e}, g, h) \leftarrow \mathsf{Gen}$;
  2. Return $\sigma = (n, \mathbb{G}, \mathbb{G}_1, \hat{e}, g, h)$
- Statement: The statement is an element $c \in \mathbb{G}$. The claim is that there exists a pair $(m, w) \in \mathbb{Z}^2$ so $m \in \{0, 1\}$ and $c = g^m h^w$.
- Proof: Input $(\sigma, c, (m, w))$.
  1. Check $m \in \{0, 1\}$ and $c = g^m h^w$;
  2. $r \in_R \mathbb{Z}_n^*$;
  3. $\pi_1 = h^r, \pi_2 = (g^{2m-1} h^w)^{wr^{-1}}, \pi_3 = g^r$;
  4. Return $\pi = (\pi_1, \pi_2, \pi_3)$.
- Verification: Input $(\sigma, c, \pi = (\pi_1, \pi_2, \pi_3))$.
  1. Check $c \in \mathbb{G}$ and $\pi \in \mathbb{G}^3$;
  2. Check $\hat{e}(c, cg^{-1}) = \hat{e}(\pi_1, \pi_2)$ and $\hat{e}(h, \pi_3) = \hat{e}(\pi_1, g)$;
  3. Return 1 if both checks succeed, 0 otherwise.

Note that this protocol achieves perfect soundness even if the prover is given q_1 since it is an NIZK proof system.