8 BK-ASPIR

8.1 Introduction

A Private Information Retrieval (PIR) involves two players: a Sender and a Receiver. The Sender manages a database DB, and answers queries on DB submitted by the Receiver. The main goal of PIR schemes is to allow the Receiver to retrieve a DB record of his choice without the Sender learning the content of his query. The property of hiding the content of the Receiver’s query from the Sender is called Privacy for the Receiver. PIR schemes are mainly concerned with providing Privacy for the Receiver. There are settings however, where the Sender too is interested in controlling access to his database. For example, the Sender could be a multimedia provider with a business model based on charging a fee for every piece of content accessed in his database. A solution to this type of settings can be obtained by using Symmetrically Private Information Retrieval (SPIR) schemes. A SPIR scheme allows a Receiver to efficiently retrieve records from the Sender’s database such that the following two properties are assured:

(1) Privacy for the Receiver : the Sender does not learn any information about the index of the target record.

(2) Privacy for the Sender : the Receiver does not learn any information on the database content, other than the target record.

Some of the records in the Database may be owned by a group of people.
Record owners can authorize the eligible persons to access the data. Consider this scenario, if a person wants to access a particular record only with proper authorizations from the set of record owners and he does not want to disclose the record index to the Database owner, but at the same time he is able to convince the Database owner that he is an authorized person to retrieve record. This could be addressed by ASPIR protocol. ASPIR[47], assumes a setting where sensitive information belonging to users is stored on a remote database managed by a party called a Sender. The setting includes an additional party called a Receiver who retrieves data owned by the user, from a database managed by the Sender, such that the following three requirements are satisfied:

1. **Privacy for the data-subject**: the Receiver can retrieve a data record only if he has a valid authorization to do so from the record owner.

2. **Privacy for the Receiver**: the sender is convinced that the Receiver’s query is authorized by the owner of the target record, without learning any information about the content of the query, or the identity of the record owner, and

3. **Privacy for the Sender**: the Receiver cannot retrieve information about more than one record per query.

In this Chapter, we construct a BK-ASPIR protocol that allows a Receiver to retrieve a record if he has authorization from the owners of the same record who do not have equal ownership rights.
8.1.1 Our Contribution

We propose a BK-ASPIR with unequal ownership rights based on BK-CP-ABE scheme. In this construction, a database record owners agree on a generalized access policy $P$. An access matrix $M$ can be formed using the three rules in LISS [21], to represent the owners (authorizers) present in the access policy $P$. Owners of the record jointly select a master key $s \in \mathbb{Z}_p$ and split it into shares using the access matrix $M$. Shares will be distributed according to the access policy $P$. The Authorizer uses their share to encrypt the secret keys and sends it to the eligible Receivers. Accessing the record is possible only if the Receiver obtains the corresponding secret keys from the set of owners.

8.1.2 Related Work

ASPIR construction in [47] cover a setting where each record in the database is owned by a single user. Moreover, the first construction relies on privacy-preserving digital credentials, homomorphic encryption and SPIR systems. The digital credential primitive has been used in addition to hide the index of the retrieved record, and to guarantee the unforgeability of the issued authorizations. They gave three ASPIR constructions, first is based on a modified version of Discrete Logarithm based credentials, second one based on RSA version of Brands credentials and the third one based on Elgamal cryptosystems.

The ASPIR protocol has been extended to a context where each database
record can have multiple owners. A multi-authorizer accredited SPIR scheme where data records stored on a Sender’s database can be retrieved by a Receiver only if (1) the Receiver has authorizations to do so from the target record owners, and (2) without the Sender learning information about the index of the retrieved record or the identity of any of the record owners. In addition, the proposed scheme allows record owners to encode, in the issued authorizations, any privacy policy they want to enforce on their data, including the Receiver's identity, an expiry date etc. Layouni et al [50] proposed a t-out-of n threshold multi-authorizer ASPIR variant, where records can be privately retrieved by a Receiver, as long as he has authorizations from t out of the n owners of the target record. They also provide a construction where owners of the same record do not have equal ownership rights.

Yinan et al [67] proposed two ASPIR protocols using Water’s [8] Ciphertext-Policy Attribute-Based Encryption for threshold multi-authorizer. In the first scheme, authorizers could authorize the data receiver separately by signing a signature. The sensitive data is preserved in a database DB and controlled by a third trusted party named sender. Sender will encrypt the required data with the access structure and send the ciphertext to data receiver. If the receiver gets enough authorizations which satisfy the access structure he can decrypts the ciphertext. In the second construction each authorizer will generate an authorization on a requirement which includes the receiver identity, index of the data and some other policy. The sender will also generate a ciphertext based on the requirement and the
access structure. The Receiver could retrieve the data from the ciphertext obtained through ASPIR scheme if the access structure is satisfied.

8.2 Definitions and Security Model

**Definition 17** Access structure Let \( \{1, 2, \ldots, n\} \) be a set of parties. A collection \( \Gamma \subseteq 2^{\{1, 2, \ldots, n\}} \) is monotone if \( \forall B, C : B \in \Gamma \) and \( B \subseteq C \) then \( C \in \Gamma \). An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) \( \Gamma \) of non-empty subsets of \( \{1, 2, \ldots, n\} \) i.e \( \Gamma \subseteq 2^{\{1, 2, \ldots, n\}} \setminus \emptyset \). The sets in \( \Gamma \) are called the authorized sets, and the sets not in \( \Gamma \) are called the unauthorized sets.

8.2.1 BK-ASPIR Protocol

It consists of four algorithms

**Setup(1^k)** It takes as input a system security parameter \( k \) and outputs, a public parameter \( PK \) and a master key \( MK \).

**KeyGen(PK, AA)** It takes as input the Public key \( PK \) of the system and the set of authorizers \( AA \). It outputs the secret key of the authorizer \( SK_{A_i} \) to the corresponding authorizer and the public key of the authorizer \( PK_{A_i} \) to the Sender.

**Encrypt(PK, PK_{A_i}, SK_{A_i}, m, r, \mathcal{P})** It takes as input the public parameters \( PK \) of the system, public key \( PK_{A_i} \) of the authorizer, secret key of the authorizer \( SK_{A_i} \),
data m, record index r and an access structure \( \mathcal{P} \). It outputs the ciphertext CT as well as the secret keys of the Receiver \( SKR_{d_i} \).

\textbf{Decrypt}(CT, SKR_{d_i}) \quad \text{It takes as the input the ciphertext CT which contains an access structure } \mathcal{P}, \text{ and the secret keys } SKR_{d_i} \text{ from the Receivers. It outputs the data } m \text{ or the special symbol } \perp \text{ which indicates failure.}

\subsection{Security Model for BK-ASPIR Protocol}

BK-ASPIR protocol is said to secure if

- The protocol satisfies the "Privacy of the Receiver" and "Privacy of the Sender" and

- A Receiver cannot retrieve a given data with non-negligible probability unless he has authorizations from a subset of authorizers of the data satisfying the access structure of authorizers.

"Privacy of the Receiver" and "Privacy of the Sender" are usually provided by the conventional SPIR scheme, so it is enough to satisfy the second requirement. To prove the same the following game has to be performed between the Adversary and the challenger.

\textbf{Init}

The adversary chooses the challenge access policy \( \mathcal{P}^* \) and gives it to the challenger.
Setup

The challenger runs the Setup algorithm and gives the public key parameters of the authorizers, \( PK_{A_i} \) to the adversary.

Phase 1

The adversary requests the secret key of the authorizer \( SKR_{d_i} \) to the Encrypt oracle for any authorizers set \( \omega = \{A_j/A_j \in AA\} \) with the restriction that \( A_j \notin \mathcal{P}^* \). The Challenger returns \( SKR_{d_i} \) to the adversary.

Challenge

The adversary submits two equal length messages \( M_0 \) and \( M_1 \). The Challenger flips a random coin \( d \), and encrypts \( M_d \) under \( \mathcal{P}^* \). The ciphertext \( CT^* \) is given to the adversary.

Phase 2

The adversary can continue querying Encrypt oracle with the same restriction as during Phase 1.

Guess

The adversary outputs a guess \( d' \) of \( d \).

Definition 18 An BK-ASPIR scheme is said to be secure against an adaptive chosen-plaintext attack (CPA) if any polynomial time adversaries have only a negligible
advantage in the IND-CPA game, where the advantage is defined to be \( \epsilon = |Pr[d' = d] - \frac{1}{2}| \).

8.3 BK-ASPIR Construction

In this scheme, we assume that each receiver R gets a RecID, which is unique and we use \( r = H(RecID||i) \), where \( i \) is the record index and \( H \) is the hash function \( H(x) : \{0,1\} \rightarrow G_0 \). The data receiver generates a ASPIR requirement on \( r \) and sends it to the Sender without \( i \). According to the ASPIR scheme, receiver could obtain a ciphertext on the requirement of \( r \) without letting the Sender know the index \( i \). In this construction, record owners agree on a generalized access structure \( P \). An access matrix \( M \) can be formed using the three rules in LISS [21], to represent the owners(authorizers) present in the access policy \( P \). Record owners jointly select a master key \( x \in \mathbb{Z}_p \) and split it into shares using the access matrix \( M \). Shares will be distributed according to the access policy \( P \). Record owners(authorizers) use their shares to generate the Secret Keys of the Receivers. Accessing the record is possible only if the Receiver obtains the corresponding secret keys from the set of owners.

**Setup** \((1^k)\)

The setup algorithm chooses a group \( G_0 \) of prime order \( p \) and a generator \( g \). Let \( AA = \{A_1, A_2, ..., A_n\} \) be the set of authorizers. Let \( y = e(g, g) \).

Public Key \( PK = (g, y) \).
KeyGen \((PK, AA)\)

This algorithm takes as input the public key PK of the system and the set of authorizers. With that it performs the following:

For each authorizer \(A_i\), it chooses a random element \(t_i \in \mathbb{Z}_p\) and calculates \(T_i = g^{t_i}\).

Master Secret key of the authorizer is \(MSK_{A_i} = t_i\) and the Public key is \(PK_{A_i} = T_i\).

\textbf{Encrypt}(PK, PK_{A_i}, SK_{A_i}, \mathcal{P}, m, r)

The encryption algorithm takes as input the public key PK of the system, public key and secret of the authorizer, a data \(m \in G_1\) to encrypt, index \(r\) and the access policy \(\mathcal{P}\).

\textbf{Step 1:} Select a random element \(s \in [-2^\ell, 2^\ell]\).

\(M\) is the distribution matrix constructed for the access policy \(\mathcal{P}\). Choose \(\rho = (s, \rho_2, ..., \rho_e)^T\), where \(\rho_i\)'s are uniformly random chosen integers in \([-2^{\ell_0+k}, 2^{\ell_0+k}]\).

\textbf{Step 2:} Computes the following and publishes the ciphertext CT:

\begin{enumerate}
  \item \(M \cdot \rho = (s_1, ..., s_d)^T\)
  \item \(C_1 = m \cdot y^{rs} = m \cdot e(g, g)^{rs}\)
  \item For each authorizer \(A_i \in \mathcal{P}\) it calculates \(C_i^* = T_i^r\).
\end{enumerate}

Ciphertext \(CT = (C_1, C_i^*)\).
**Step 3:** In this step the authorizer generates the Secret Key for the eligible Receivers:

Authorizer $A_i$ computes the Secret key $SKR_{d_i} = g^{s_i t_i^{-1}}$ and sends it to the eligible Receivers.

**Decrypt($CT, SKR_{d_i}$)**

The decryption algorithm takes as input a ciphertext $CT$ and a set of secret keys from the eligible receivers that can be collected in set $A$. Suppose $A$ satisfies the access policy $\mathcal{P}$ then by Lemma 1, $\lambda_i$ must exist. With this, it is possible to reconstruct the secret using $\sum_{i \in A} \lambda_i s_i = s$.

The decryption algorithm computes

$$C_1 \left( \prod_{i \in A} e(C_i^* (SKR_{d_i})^{\lambda_i}) \right) = m.$$

### 8.4 Security Analysis

**Theorem 8.1** In BK-ASPIR protocol, Privacy for the Sender, Receiver and authorizer is fully achieved under DBDH assumption and through secure SPIR protocol.

**Proof:** "Privacy for the Sender" and "Privacy for the Receiver" properties are provided by the underlying SPIR primitive. In this theorem we provide the security of the "Privacy for the Authorizer". Suppose we have an adversary $A$ with non-
negligible advantage $\epsilon$ in the selective security game against our construction. We show how to use the adversary $A$ to build a simulator $B$ that is able to solve the DBDH assumption. The Challenger gives the simulator $B$ the DBDH challenge: $(g, A, B, C, D) = (g, g^a, g^b, g^s, D)$.

**Init.** The adversary chooses the challenge access policy $(M', \mathcal{P}^*)$ and gives it to the simulator.

**Setup** For all $a_j \in \mathcal{P}^*$ it chooses a random $q_j \in \mathbb{Z}_p$ and set $T_j = g^{q_j}$. The simulator $B$ sends the public parameters to $A$.

**Phase 1** $A$ makes secret key requests for any set of authorizers $\omega = \{a_j/a_j \in U\}$ with the restriction that $a_j \not\in \mathcal{P}^*$. On each request $B$, will choose uniformly random integers $z_2, ..., z_h \in [-2^{\epsilon_0+k}, 2^{\epsilon_0+k}]$ and share the secret $s$, using the vector $\Phi = (s, z_2, ..., z_h)$.

Create the distribution matrix $M$, for the access policy $\mathcal{P}^*$. Compute $M \cdot \Phi$ and use the shares to encrypt the access policy with corresponding shares of the authorizers present in the access policy $\mathcal{P}^*$,

$$SKR_{a_j} = g^{s, q_j}, \forall a_j \in \omega$$

**Challenge** $A$ submits two messages $m_0, m_1 \in G_1$. The simulator flips a fair binary coin $d$, and returns the encryption of $m_d$. The encryption of $m_d$ can be done as follows:
Table 6: Comparison of ASPIR Schemes

<table>
<thead>
<tr>
<th>Method</th>
<th>Encryption Time</th>
<th>Decryption Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yinan et al. [67]</td>
<td>$n + 1$ pairings, $2n$ exponentiations</td>
<td>$2n$ pairings</td>
</tr>
<tr>
<td>BK-ASPIR</td>
<td>$1$ pairing, $n + 1$ exponentiations</td>
<td>$n$ pairings</td>
</tr>
</tbody>
</table>

$C_1 = m_d e(g, g)^{bs}$

$C_j^* = g^{(q_j)}$.

**Phase 2**  
Same as Phase 1.

**Guess**  
A outputs a guess $d'$ of $d$. The simulator then outputs 0 to the guesses that $D = e(g, g)^{abs}$ if $d' = d$; otherwise, it outputs 1 to indicate that it believes $D$ is random group element in $G_1$.

When $D$ is a tuple the simulator $B$ gives a perfect simulation, so we have that $Pr [B(\rho, D = e(g, g)^{abs}) = 0] = \frac{1}{2} + \epsilon$.

When $D$ is a random group element the message $m_d$ is completely hidden from the adversary, and we have $Pr [B(\rho, D = R) = 0] = \frac{1}{2}$.

**8.4.1 Efficiency**

In Table 6, we compare BK-ASPIR method and Yinan et al. [67] method in terms of number of pairings and exponentiations taken by the Encryption and Decryption algorithm. It becomes obvious from the table that BK-ASPIR requires less number of pairings and exponentiations than the Yinan et al. method.
Summary

We proposed a BK-ASPIR protocol that allows a Receiver to retrieve a record if he has authorizations from the owners of the same record do not have equal ownership rights. It also supports the receiver to retrieve the record if he has \( t \) authorizations out of \( n \) authorizers. We constructed this BK-ASPIR protocol using BK-CP-ABE, and the scheme is provably secure under the Decisional Bilinear Diffie-Hellman assumption.