SPIR with Colluding and Non-Replicated Servers from a Noisy Channel

Amirhossein Shekofteh
Department of Computer Science & Engineering
The University of Texas at Arlington
Arlington, TX 76019
amirhossein.shekofteh@uta.edu

Rémi A. Chou
Department of Computer Science & Engineering
The University of Texas at Arlington
Arlington, TX 76019
remi.chou@uta.edu

2024 60th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3315-4103-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ALLERTON63246.2024.10735295

***Abstract*—We study the problem of Symmetric Private Information Retrieval (SPIR) in a scenario with L non-replicated and colluding servers, and M independent files distributed across these servers. In this setting, communication takes place through a noisy multiple-access channel and a noiseless public channel. The client must retrieve one of the M files such that (i) the client's choice must not be revealed to the servers, and (ii) the client must not learn any information about non-selected files. Our main contribution is showing that, for a specific class of channels and without requiring shared randomness among servers, positive rates are achievable even when all the servers collude. Additionally, we present an example of channel where distributing files across multiple servers yields an achievable rate that outperforms a setting where all the files are stored on a single server.**

***Index Terms*—symmetric private information retrieval, oblivious transfer, private information retrieval**

I. Introduction

Symmetric Private Information Retrieval (SPIR), first introduced in [1], is an extended form of Private Information Retrieval (PIR) [2]–[4]. PIR allows a client to retrieve a file from a server without the server knowing which file was retrieved. SPIR addresses both the privacy of the client's queries and the privacy of the server's contents against the client. Specifically, SPIR also ensures that the client cannot learn more than the selected file, preserving server privacy. Recent studies have investigated different settings of the SPIR problem [5]–[11], including settings with replicated servers, shared randomness among the servers, and noiseless communication. For large databases, replicating data can be costly in terms of storage and maintenance. Additionally, in a noiseless SPIR setting, achieving information-theoretic security is impossible if all servers collude.

To address these challenges, in this work, we consider SPIR with non-replicated and colluding servers, and without requiring shared randomness among servers. Specifically, consider M independent files, each split into L segments, with each segment stored on one of L servers. The client must retrieve one of the M files, by querying each server for its corresponding segments, without revealing the choice to the servers, and the client must not learn any information about the non-selected files. Communication occurs via a noisy multiple-access channel (MAC), where the servers control the inputs and the client observes the output, and a noiseless public channel, where the client can communicate with the servers. All entities strictly follow the protocol. The main contributions of this paper are: (i) Demonstrating that, for non-replicated servers and without requiring shared randomness among the servers, positive rates are achievable for a certain class of channels, even when all the servers can collude; (ii) Demonstrating that distributing files across multiple non-replicated servers can yield a rate gain compared to setups where all files are stored on a single server.

This work was supported in part by NSF grant CCF-2401373.

A. Related works

Most SPIR settings focus on non-colluding and replicated servers that share common randomness, e.g., [7]–[11]. Furthermore, in [5], [6], [12] if all the servers collude, then information-theoretic security cannot be achieved.

Another series of related works focuses on oblivious transfer (OT) protocols under information-theoretic security guarantees e.g., [13]–[15]. Similar to SPIR, OT allows a server to send the file selected by the client without revealing the client's choice or any information about the non-selected files. Most works on OT consider protocols over noisy channels with a single server and client, e.g., [16], [17], with the exception of [18], which considers two non-replicated servers, and [19], which considers two replicated servers.

Unlike most OT works, e.g., [13]–[17], most SPIR works consider multiple servers with replicated data and communication over a noiseless channel, e.g., [5]–[11].

B. Main differences with previous works

Our setting considers a noisy MAC, whose output is observed by the client. In contrast, [5], [6], [18], [20] rely on a noiseless channel model, and the results cannot be applied to our setting.

To ensure privacy in our achievability scheme, the servers encrypt their files with secret keys, allowing the client to decrypt only the chosen file. To this end, we utilize a secret-key generation model similar to [21], which allows all servers to generate multiple keys with the client simultaneously. In contrast, [17] employs a key generation method where a single server generates a key with the client.

We store files on multiple servers using our distribution method without data replication. This differs from approaches that replicate all files on each server, e.g., [5]–[12], [19], [22]–[24], and from methods that store all files on a single server without replication, e.g., [16], [17]. Compared to [17], our distribution method may yield a rate gain, as demonstrated in the example provided in Section III.

Our approach achieves positive rates for certain classes of channels without requiring shared randomness and when all the servers can collude. In contrast, many SPIR settings assume non-colluding servers with shared randomness, e.g., [7]–[11], [19], [20], [22], and in settings where T out of N servers can collude, e.g., [5], [6], information-theoretic security is unattainable if all servers collude, i.e., when $T = N$.

C. Paper organization

First, we formally introduce the setting in Section II. Then, we present the main results in Section III. In Section IV, we show the achievability proof for two files, and then, in Section V, we provide the achievability proof for an arbitrary number of files. Finally, we provide concluding remarks in Section VI.

II. Problem statement

For $a, b \in \mathbb{R}$, define $[\![a, b]\!] \triangleq [\lfloor a \rfloor, \lceil b \rceil] \cap \mathbb{N}$. For any $x \in [\![0, 1]\!]$, define $\bar{x} \triangleq 1 - x$. Consider a Multiple Access Channel (MAC) $(\mathcal{X}_{\mathcal{L}}, W_{Y|X_{\mathcal{L}}}, \mathcal{Y})$, where $\mathcal{X}_{\mathcal{L}} \triangleq \bigtimes_{l\in\mathcal{L}} \mathcal{X}_l$, L is the number of inputs, and $\mathcal{L} \triangleq [\![1, L]\!]$. In the following, we assume that all the participants in the protocol strictly follow the protocol.

Definition 1. *An $(n, M, (k_l)_{l\in\mathcal{L}})$ Symmetric Private Information Retrieval (SPIR) protocol consists of*

- *A set of files $D_{\mathcal{M}}$, where $\mathcal{M} \triangleq [\![1, M]\!]$, uniformly distributed over $\{0,1\}^{\sum_{i=1}^{L} k_i}$. Each file $D_{\mathcal{M}} \triangleq (D_m)_{m\in\mathcal{M}}$ is divided into L segments, with each segment stored on a separate server. For $i \in \mathcal{M}$, define $D_i \triangleq K_{\mathcal{L},i}$, were $K_{\mathcal{L},i} \triangleq (K_{l,i})_{l\in\mathcal{L}}$ and each $K_{l,i}$ is a uniformly distributed string over $\{0,1\}^{k_l}$, available at Server $l \in \mathcal{L}$.*
- *A random variable Z, uniformly distributed over $\mathcal{M}$ which represents the client's file choice, i.e., $Z = i \in \mathcal{M}$ means that the client is requesting D_i;*
- *Independent random variables R_l, $l \in [\![0, L]\!]$, which represent local randomness available at Server l, with local randomness R_0 available at the client;*
- *For $t \in [\![1, n]\!]$, r_t is the number of public communication rounds between the servers and the client between the $t-th$ and $t+1-th$ channel uses;*
- *For $l \in \mathcal{L}$, $F_{l,0} \triangleq \emptyset$;*
- *For $t \in [\![1, n]\!]$ and $i \in [\![1, 2L]\!]$, $m_{i,t,0} \triangleq \emptyset$;*

and operates as follows from $t = 1$ to $t = n$, for $l \in \mathcal{L}$ and $l' \triangleq l + L$.

The servers send $(X_{\mathcal{L}})_t \triangleq ((X_l)_t)_{l\in\mathcal{L}} \in \mathcal{X}_{\mathcal{L}}$ over the MAC $(\mathcal{X}_{\mathcal{L}}, W_{Y|X_{\mathcal{L}}}, \mathcal{Y})$ and the client observes $Y_t \in \mathcal{Y}$. $(X_l)_t$ is a function of $(K_{l,\mathcal{M}}, R_l, (F_{l,i})_{i\in[\![1,t-1]\!]})$, $F_{l,t}$ represents all the messages publicly exchanged between Server l and the client between the $t-th$ and $t+1-th$ channel use. Specifically, $F_{l,t} \triangleq (m_{l,t,j}, m_{l',t,j})_{j\in[\![1,r_t]\!]}$, where $(m_{l,t,j})_{j\in[\![1,r_t]\!]}$ and $(m_{l',t,j})_{j\in[\![1,r_t]\!]}$ are defined as follows:

From $j = 1$ to r_t:

- *Server l sends to the client the message*

$$m_{l,t,j}(K_{l,\mathcal{M}}, R_l, (F_{\mathcal{L},a})_{a\in[\![1,t-1]\!]}, (m_{\mathcal{C},t,i})_{i\in[\![1,j-1]\!]}) \quad (1)$$

- *The client sends to Server $(l' - L)$ the message*

$$m_{l',t,j}(Z, R_0, Y^t, (F_{\mathcal{L},a})_{a\in[\![1,t-1]\!]}, (m_{\mathcal{C},t,i})_{i\in[\![1,j-1]\!]}) \quad (2)$$

where $(F_{\mathcal{L},a})_{a\in[\![1,t-1]\!]} \triangleq (F_{l,a})_{l\in\mathcal{L},a\in[\![1,t-1]\!]}$, $\mathcal{C} \triangleq [\![1, 2L]\!]$ and $(m_{\mathcal{C},t,i})_{i\in[\![1,j-1]\!]} \triangleq (m_{c,t,i})_{c\in\mathcal{C},i\in[\![1,j-1]\!]}$;

After the final round, define the entire public communication by $F \triangleq (F_{i,t})_{i\in\mathcal{C},t\in[\![1,n]\!]}$. The client forms $\widehat{K}_{\mathcal{L},Z}$, an estimate of $K_{\mathcal{L},Z}$, from (Z, R_0, Y^n, F). Finally, the client forms $\widehat{D}_Z$, an estimate of D_Z from $\widehat{K}_{\mathcal{L},Z}$.

Definition 2. *A rate R is achievable if there exists a sequence of $(n, M, (k_l)_{l\in\mathcal{L}})$ SPIR protocols such that $\lim_{n\to\infty} \frac{\sum_{l\in\mathcal{L}} k_l}{n} = R$,*

$$\lim_{n\to\infty} \mathbb{P}[\widehat{D}_Z \neq D_Z] = 0, \quad (3)$$

$$\lim_{n\to\infty} I(D_{\mathcal{M}}, R_{\mathcal{L}}, X_{\mathcal{L}}^n, F; Z) = 0, \quad (4)$$

$$\lim_{n\to\infty} I(Z, R_0, Y^n, F; D_{\mathcal{M}\setminus\{Z\}}) = 0. \quad (5)$$

Equation (3) ensures that the client obtains the selected file. Equation (4) ensures the client's privacy by keeping the file selection Z private from the servers. Equation (5) ensures the servers' privacy by keeping the non-selected files private from the client.

In [17, Definition 2], the Binary Erasure Channel (BEC) is used to achieve a positive rate in an OT setting with one server, one client, and two files. [17] also considers the Generalized Erasure Channel (GEC), as a generalization of the BEC. In this paper, we define a broader class of channels, the Erasure Multiple Access Channel (EMAC), which supports an arbitrary number of inputs. Note that the GEC is a special case of the EMAC.

Definition 3. *An EMAC is a MAC $(\mathcal{X}_{\mathcal{L}}, W_{Y|X_{\mathcal{L}}}, \mathcal{Y})$, where, for some nonempty set $\mathcal{Y}_1 \subset \mathcal{Y}$, the probabilities $W_{Y|X_{\mathcal{L}}}(y|x_{\mathcal{L}})$, $y \in \mathcal{Y}_1$, do not depend on $x_{\mathcal{L}} \triangleq (x_l)_{l\in\mathcal{L}} \in \mathcal{X}_{\mathcal{L}}$. Outputs $y \in \mathcal{Y}_1$ are considered erasures as they provide no information about the inputs. The erasure probability of an EMAC is defined as $\epsilon \triangleq \sum_{y\in\mathcal{Y}_1} W(y|x_{\mathcal{L}})$.*

In Section III, we provide achievable rates for the EMAC and present an example to compare these results with related works.

III. Main results

In this section, we present achievale rates for our setting in Theorems 1 and 2. Then, we provide an example to demonstrate how our derived achievable rate can outperform single-server settings.

Theorem 1. *An achievable rate for the EMAC* $(\mathcal{X}_{\mathcal{L}}, W_{Y|X_{\mathcal{L}}}, \mathcal{Y})$ *with erasure probability* $\epsilon \in [1/2, 1]$ *is*

$$\max_{p_{x_{\mathcal{L}}}} \frac{I(X_{\mathcal{L}}; Y)}{M-1}. \tag{6}$$

Theorem 2. *An achievable rate for the EMAC* $(\mathcal{X}_{\mathcal{L}}, W_{Y|X_{\mathcal{L}}}, \mathcal{Y})$ *with erasure probability* $\epsilon \in [0, 1/2]$ *is*

$$\max_{p_{x_{\mathcal{L}}}} \frac{\epsilon}{(M-1)(1-\epsilon)} I(X_{\mathcal{L}}; Y). \tag{7}$$

To prove Theorem 1, we first consider Algorithm 1 in Section IV to address the special case $M = 2$ files. We then iterate Algorithm 1 using Algorithm 2, as detailed in Section V, to establish Theorem 1. The proof of Theorem 2 is similar to that of Theorem 1 and is omitted due to space constraints.

To show the benefit of distributing files across multiple servers, we present an example involving an EMAC with two servers and one client. By distributing the files among the servers, we aim to achieve a higher achievable rate compared to settings where all files are stored on a single server. Specifically, we show that, for a specific EMAC, our setting achieves a rate gain of $log(3)$ compared to the model in [17, Theorem 2], which uses a Binary Erasure Channel (BEC) and stores all files on a single server.

1) Distributed files setting: Consider an EMAC $(\mathcal{X}_1 \times \mathcal{X}_2, W_{Y|X_1,X_2}, \mathcal{Y})$ with inputs $X_1, X_2 \in \mathcal{X}_1 \times \mathcal{X}_2 \triangleq \{0, 1\}$, $Y \in \mathcal{Y} \triangleq \{0, 1, 2, e\}$ and $\mathcal{Y}_1 = \{e\}$. In this setup, each server stores a portion of the file, and the sum of the server inputs is passed through an erasure channel $\mathcal{E}$ that erases the input with probability $\alpha \in [0, 1/2]$ or transmits it unchanged, as illustrated in Figure 1.

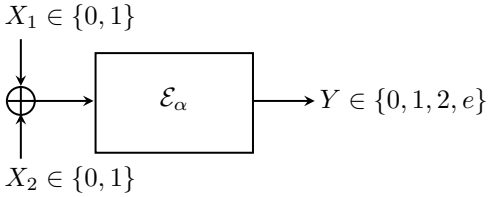

Figure 1. An EMAC with erasure probability α

We define $W_{Y|X_1,X_2}$ and p_{x_1,x_2} as shown in (8).

$$W_{Y|X_1,X_2} = \begin{bmatrix} W(0|0,0) & W(0|1,0) & W(0|0,1) & W(0|1,1) \\ W(1|0,0) & W(1|1,0) & W(1|0,1) & W(1|1,1) \\ W(2|0,0) & W(2|1,0) & W(2|0,1) & W(2|1,1) \\ W(e|0,0) & W(e|1,0) & W(e|0,1) & W(e|1,1) \end{bmatrix} = \begin{bmatrix} (1-\alpha) & 0 & 0 & 0 \\ 0 & (1-\alpha) & (1-\alpha) & 0 \\ 0 & 0 & 0 & (1-\alpha) \\ \alpha & \alpha & \alpha & \alpha \end{bmatrix},$$

$$p_{X_1,X_2} = \begin{bmatrix} p(0,0) \\ p(1,0) \\ p(0,1) \\ p(1,1) \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}. \tag{8}$$

For $\alpha \geq \frac{1}{2}$, by (6), we have

$$R = (1-\alpha)\log(3), \tag{9}$$

similarly, for $\alpha \leq \frac{1}{2}$, by (7), we have

$$R = \alpha \log(3). \tag{10}$$

2) Single-server setting: [17, Theorem 2] considers a BEC model, where all files are stored on a single server. The model in [17] aligns with our setup, where one server stores all the files and the other server does not send anything, i.e., its input is zero. Without loss of generality, assume that X_2 is always zero. By removing the terms $W_{Y|X_1,X_2=1}$ and $p_{x_1,x_2=1}$ from (8), we can represent the updated $W_{Y|X_1,X_2}$ and p_{x_1,x_2} as shown in (12) and illustrated in Figure 2. Hence, the BEC model in [17] is a special case of our EMAC setting, for which the capacity is known to be

$$C_{\text{BEC}} = \min(1-\alpha, \alpha). \tag{11}$$

$$W_{Y|X_1,X_2} = \begin{bmatrix} W(0|0,0) & W(0|1,0) \\ W(1|0,0) & W(1|1,0) \\ W(e|0,0) & W(e|1,0) \end{bmatrix} = \begin{bmatrix} (1-\alpha) & 0 \\ 0 & (1-\alpha) \\ \alpha & \alpha \end{bmatrix},$$

$$p_{X_1,X_2} = \begin{bmatrix} p(0,0) \\ p(1,0) \end{bmatrix} = \begin{bmatrix} p \\ 1-p \end{bmatrix}. \tag{12}$$

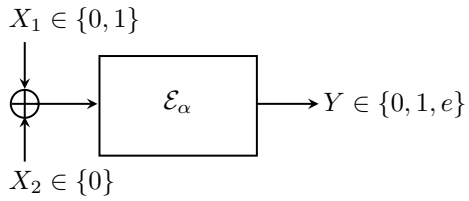

Figure 2. An EMAC with erasure probability of α and $X_2 \in \{0\}$

For the distributed files setting, we showed that the rate $R = \min(1-\alpha, \alpha)\log(3)$ is achievable. For the single-server setting, we showed that $C_{\text{BEC}} = \min(1-\alpha, \alpha)$. This demonstrates a rate gain of $\log(3)$ in our distributed model.

IV. Proof of Theorem 1 for M=2

In Section IV-A, we present Algorithm 1 to derive an achievable rate when $M = 2$. Then, in Section IV-B, we present the analysis of our coding scheme. To preserve the servers' privacy, the servers encrypt their files with secret keys,

allowing the client to decrypt only the selected file. These keys are uniformly distributed and independent, and will be generated using Lemma 1.

Lemma 1 ([21, Section 2]). *Let $(A_{\mathcal{L}}^n, B^n)$ be i.i.d. repetitions of random variables $(A_{\mathcal{L}}, B)$. For any $\delta > 0$ and $i \in \mathcal{L}$ there exist functions $\kappa_i(A_i^n)$ and $V_i(A_i^n)$, where the range of κ_i is $\{0,1\}^{k_i}$,*

$$k_{\mathcal{L}} = n(I(A_{\mathcal{L}}; B) - \delta), \tag{13}$$

$$H(\kappa_{\mathcal{L}} | V_{\mathcal{L}}, B^n) \leq \delta, \tag{14}$$

$$\lim_{n\to\infty} I(\kappa_{\mathcal{L}}; V_{\mathcal{L}}) = 0, \tag{15}$$

$$\lim_{n\to\infty} k_{\mathcal{L}} - H(\kappa_{\mathcal{L}}) = 0. \tag{16}$$

where $k_{\mathcal{L}} \triangleq \sum_{l\in\mathcal{L}} k_l$, $\kappa_{\mathcal{L}} \triangleq (\kappa_i(A_i^n))_{i\in\mathcal{L}}$, and $V_{\mathcal{L}} \triangleq (V_i(A_i^n))_{i\in\mathcal{L}}$.

A. Coding scheme

Each entity follows the protocol described in Algorithm 1.

B. Coding scheme analysis

1) Reliability: One can show that the client can recover $\kappa_{\mathcal{L},Z}$, using $(V_{\mathcal{L},Z}, Y^n[S_Z])$ by (14), and then compute $K_{\mathcal{L},Z}$ by (19).

2) Privacy of the client selection: Define $A_{\mathcal{L},\mathcal{M}} \triangleq (A_{i,j})_{i\in\mathcal{L},j\in\mathcal{M}}$, where $A_{i,j} \triangleq (V_{i,j}, \kappa_{i,j} \oplus K_{i,j})$ for $i \in \mathcal{L}$ and $j \in \mathcal{M}$, and define $S_{\mathcal{M}} \triangleq (S_j)_{j\in\mathcal{M}}$. Then, we have

$$\begin{aligned}
&I(D_{\mathcal{M}}, R_{\mathcal{L}}, X_{\mathcal{L}}^n, F; Z)\\
&= I(K_{\mathcal{L},\mathcal{M}}, R_{\mathcal{L}}, X_{\mathcal{L}}^n, F; Z)\\
&= I(K_{\mathcal{L},\mathcal{M}}, R_{\mathcal{L}}, X_{\mathcal{L}}^n, A_{\mathcal{L},\mathcal{M}}, S_{\mathcal{M}}; Z)\\
&\overset{(a)}{=} I(K_{\mathcal{L},\mathcal{M}}, R_{\mathcal{L}}, X_{\mathcal{L}}^n, S_{\mathcal{M}}; Z)\\
&\overset{(b)}{=} I(X_{\mathcal{L}}^n, S_{\mathcal{M}}; Z)\\
&= I(S_{\mathcal{M}}; Z) + I(X_{\mathcal{L}}^n; Z | S_{\mathcal{M}})\\
&\overset{(c)}{\leq} I(X_{\mathcal{L}}^n; S_{\mathcal{M}}, Z)\\
&\leq I(X_{\mathcal{L}}^n; \mathcal{G}, \mathcal{B}, S_{\mathcal{M}}, Z)\\
&\overset{(d)}{=} I(X_{\mathcal{L}}^n; \mathcal{G}, \mathcal{B}, Z)\\
&= I(X_{\mathcal{L}}^n; Z) + I(X_{\mathcal{L}}^n; \mathcal{G}, \mathcal{B} | Z)\\
&\overset{(e)}{=} I(X_{\mathcal{L}}^n; \mathcal{G}, \mathcal{B})\\
&\overset{(f)}{=} 0,
\end{aligned} \tag{20}$$

where (a) holds because for $l \in \mathcal{L}$, $A_{l,\mathcal{M}}$ is a function of $(K_{l,\mathcal{M}}, X_l^n, S_{\mathcal{M}})$, (b) holds because $(K_{\mathcal{L},\mathcal{M}}, R_{\mathcal{L}})$ are independent of $(X_{\mathcal{L}}^n, S_{\mathcal{M}}, Z)$, (c) holds by the chain rule and $I(S_{\mathcal{M}}; Z)$ is equal to zero, because Z and $S_{\mathcal{M}}$ are independent, (d) holds because $S_{\mathcal{M}}$ are functions of $(\mathcal{G}, \mathcal{B}, Z)$, (e) holds because Z is independent of $(X_{\mathcal{L}}^n, \mathcal{G}, \mathcal{B})$, (f) holds because the pair of random sets $(\mathcal{G}, \mathcal{B})$ is independent of $X_{\mathcal{L}}^n$.

3) Privacy of the non-selected strings for the servers: The proof that the client does not learn any information about the non-selected files is omitted due to space constraints.

Algorithm 1 2-source SPIR

1: Server $i \in \mathcal{L}$ sends X_i^n, i.i.d. according to $p_{x_{\mathcal{L}}}$, over the EMAC $(\mathcal{X}_{\mathcal{L}}, V_{Y|X_{\mathcal{L}}}, \mathcal{Y})$.

2: Upon observing $Y^n = (Y_1, ..., Y_n)$, the client follows the protocol as outlined below

- Define
$$\mathcal{G}_1 \triangleq \{i \in [\![1, n]\!] : Y_i \in \mathcal{Y}_0\},$$
$$\mathcal{B}_1 \triangleq \{i \in [\![1, n]\!] : Y_i \in \mathcal{Y}_1\}.$$
- Define $I \triangleq \min(|\mathcal{G}_1|, |\mathcal{B}_1|)$.
- If $|\mathcal{G}_1| > I$, construct $\mathcal{G}$ and $\mathcal{B}$ such that $|\mathcal{G}| = I$
$$\mathcal{G} \triangleq \{i \in \mathcal{G}_1 : R_0(i) = 1\},$$
$$\mathcal{B} \triangleq \mathcal{B}_1, \tag{17}$$
where R_0 is a sequence of n independent bits such that $\mathbb{P}[R_0(i) = 1] = \frac{\epsilon}{1-\epsilon}, \forall i \in [\![1, n]\!]$. Otherwise, construct construct $\mathcal{G}$ and $\mathcal{B}$ such that $|\mathcal{B}| = I$
$$\mathcal{B} \triangleq \{i \in \mathcal{B}_1 : R_0(i) = 1\},$$
$$\mathcal{G} \triangleq \mathcal{G}_1, \tag{18}$$
where R_0 is a sequence of n independent bits such that $\mathbb{P}[R_0(i) = 1] = \frac{1-\epsilon}{\epsilon}, \forall i \in [\![1, n]\!]$.
- Define
$$S_0 \triangleq \begin{cases} \mathcal{G} & \text{if } Z = 0 \\ \mathcal{B} & \text{if } Z = 1 \end{cases}, \quad S_1 \triangleq \begin{cases} \mathcal{B} & \text{if } Z = 0 \\ \mathcal{G} & \text{if } Z = 1 \end{cases}.$$
- Send S_0 and S_1 to the servers.

3: The Servers respond as follows:

- The servers check whether $|S_0| \leq rn$ or $|S_1| \leq rn$, where $r \leq \min(\epsilon, 1-\epsilon)$, if either conditions holds, the servers abort the protocol. Otherwise, for $j \in [\![0, 1]\!]$, the server selects the first rn elements of S_j to form $S_j' \triangleq S_j([\![1, rn]\!])$.
- For $j \in \{0, 1\}$, Server $i \in \mathcal{L}$ applies Lemma 1 with the substitutions $n \leftarrow rn$, $A_i^n \leftarrow X_i^n[S_j']$ and $B^n \leftarrow Y^n[S_j']$, then computes
  - $\kappa_{i,j} \triangleq \kappa_i(X_i^n[S_j'])$,
  - $V_{i,j} \triangleq V_i(X_i^n[S_j'])$.

4: Server i sends $(V_{i,j}, (K_j + \kappa_{i,j}))_{j\in\{0,1\}}$ to the client over the noiseless channel.

5: The client obtains its file selection as follows

- One can show that the client can recover $\kappa_{\mathcal{L},Z}$, using $(V_{\mathcal{L},Z}, Y^n[S_Z])$ by (14). The client then computes
$$\kappa_{i,Z} \oplus (K_{i,Z} \oplus \kappa_{i,Z}) = K_{i,Z}. \tag{19}$$
- Finally, the client recovers D_Z from $(K_{i,Z})_{i\in\mathcal{L}}$.

4) Achieved rate: The rate is

$$\lim_{n\to\infty} \frac{k_{\mathcal{L}}}{n} = I(X_{\mathcal{L}}; Y) - \delta, \tag{21}$$

where the equality holds with the substitutions $A_{\mathcal{L}} \leftarrow X_{\mathcal{L}}$, $B \leftarrow Y$ in (13).

V. Proof of Theorem 1 for $M > 2$

In this section, we present Algorithm 2 to derive an achievable rate for M-source SPIR. We first introduce the coding scheme in Section V-A, and then present its analysis in Section V-B.

A. Coding scheme

The achievability scheme is described in Algorithm 2.

Algorithm 2 M-source SPIR from $(M-1)$ 2-source SPIR

Require: $M-2$ sequences $(S_t)_{t\in[\![1,M-2]\!]}$ uniformly distributed over $\{0,1\}^{k_{\mathcal{L}}}$. For $t \in [\![1, M-2]\!]$, S_t is represented as $(S_{j,t})_{j\in\mathcal{L}}$, where $S_{j,t}$ is uniformly distributed over $\{0,1\}^{k_j}$.

1: Server $j \in \mathcal{L}$ forms $(C_{j,t})_{t\in[\![1,M-1]\!]}$ as follows:

$$\begin{aligned}
&(C_{j,1}[0], C_{j,1}[1]) \\
&\quad \triangleq (K_{j,1}, S_{j,1}), \\
&(C_{j,t}[0], C_{j,t}[1]) \\
&\quad \triangleq (K_{j,t} \oplus S_{j,t-1}, S_{j,t-1} \oplus S_{j,t}), \forall t \in [\![2, M-2]\!], \\
&(C_{j,M-1}[0], C_{j,M-1}[1]) \\
&\quad \triangleq (K_{j,M-1} \oplus S_{j,M-2}, S_{j,M-2} \oplus K_{j,M}).
\end{aligned} \tag{22}$$

For $t \in [\![1, M-1]\!]$, define $C_{j,t} \triangleq (C_{j,t}[0], C_{j,t}[1])$.

2: The client forms

$$Z_t \triangleq \mathbb{1}\{t < Z\}, \forall t \in [\![1, M-1]\!]. \tag{23}$$

3: **for** $t = 1$ to $M-1$ **do**

4: The client and servers perform Algorithm 1 using the sequences $(C_{j,t}[0], C_{j,t}[1])$ at Server $j \in \mathcal{L}$ and the selection Z_t for the client.

5: **end for**

6: By Lines 2-4, the client forms

$$K_{j,Z} = \bigoplus_{t=1}^{Z} C_{j,t}[Z_t]. \tag{24}$$

B. Coding scheme analysis

1) Reliability: The client can obtain the selected file D_z by first acquiring $K_{j,Z}$ in Line 6 of Algorithm 2, and then computing D_z using $(K_{j,Z})_{j\in\mathcal{L}}$.

2) Privacy of the client selection: The proof that the servers gain no information about the client's choice is omitted due to space constraints.

3) Privacy of the non-selected strings for the servers: The proof that the client does not learn any information about the non-selected files is omitted due to space constraints.

4) Achieved rate: To compute the achieved rate, we normalize the file length by $n(M-1)$ rather than n, because the channel is utilized $n(M-1)$ times:

$$\begin{aligned}
\frac{|D_Z|}{n(M-1)} &= \frac{|K_{\mathcal{L},Z}|}{n(M-1)} \\
&= \frac{k_{\mathcal{L}}}{n(M-1)} \\
&\xrightarrow{n\to\infty} \frac{I(X_{\mathcal{L}}; Y) - \delta}{M-1},
\end{aligned} \tag{25}$$

where the first two equalities hold by the definitions of the files, and the limit holds by Lemma 1.

VI. Conclusion

We derived an achievable rate for the SPIR problem with non-replicated and non-colluding servers, using a noisy multiple access channel. Our results showed that, for a certain class of channels, positive rates are achievable even when all servers can collude. Additionally, for the special case of two colluding servers and two files, we gave an example of an EMAC that yields a rate gain of $\log(3)$ compared to the BEC model presented in [17, Theorem 2].

References

[1] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," in *Proc. 30th Annu. ACM Symp. Theory Comput. (STOC)*, 1998, pp. 151–160.

[2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[3] I. Goldberg, "Improving the robustness of private information retrieval," in *Proc. IEEE Symp. Secur. Priv.*, 2007, pp. 131–148.

[4] A. Beimel and Y. Ishai, "Information-theoretic private information retrieval: a unified construction," in *Automata, Lang. Program.*, 2001, pp. 912–926.

[5] Q. Wang and M. Skoglund, "Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2017, pp. 71–75.

[6] ——, "On PIR and symmetric PIR from colluding databases with adversaries and eavesdroppers," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3183–3197, 2019.

[7] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, 2019.

[8] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[9] Z. Wang and S. Ulukus, "Communication cost of two-database symmetric private information retrieval: a conditional disclosure of multiple secrets perspective," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 402–407.

[10] ——, "Symmetric private information retrieval at the private information retrieval rate," *IEEE J. Sel. Areas Inf. Theory*, vol. 3, no. 2, pp. 350–361, 2022.

[11] ——, "Symmetric private information retrieval with user-side common randomness," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2021, pp. 2119–2124.

[12] K. Banawan and S. Ulukus, "The capacity of private information retrieval from Byzantine and colluding databases," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1206–1219, 2018.

[13] M. O. Rabin, "How to exchange secrets with oblivious transfer," *Cryptology ePrint Archive*, vol. Report 2005/187, 2005, available: https://eprint.iacr.org/2005/187.

[14] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.

[15] C. Crépeau, "Equivalence between two flavours of oblivious transfers," in *Advances in Cryptology—EUROCRYPT '87*, 1987, pp. 350–354.
[16] A. C. A. Nascimento and A. Winter, "On the oblivious-transfer capacity of noisy resources," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2572–2581, 2008.
[17] R. Ahlswede and I. Csiszár, "On oblivious transfer capacity," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2007, pp. 2061–2064.
[18] R. A. Chou, "Pairwise oblivious transfer," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2021, pp. 1–5.
[19] T. Pei, W. Kang, and N. Liu, "The capacity of oblivious transfer with replicated databases and binary erasure multiple access channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2023, pp. 1592–1596.
[20] R. Chou, "Dual-source symmetric PIR without data replication or shared randomness," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2024, accepted.
[21] L. Lai and L. Huie, "Simultaneously generating multiple keys in many to one networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2013, pp. 2394–2398.
[22] K. Banawan and S. Ulukus, "Noisy private information retrieval: on separability of channel coding and information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8232–8249, 2019.
[23] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.
[24] K. Banawan and S. Ulukus, "Multi-message private information retrieval: capacity results and near-optimal schemes," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6842–6862, 2018.