

The Capacity of Multi-user Private Information Retrieval for Computationally Limited Databases

William Barnhart* and Zhi Tian†

Department of Electrical and Computer Engineering, George Mason University

Fairfax, VA, USA

*wbarnha2@gmu.edu, †ztian@gmu.edu

Abstract—We present a private information retrieval (PIR) scheme that allows a user to retrieve a single message from an arbitrary number of databases by colluding with other users while hiding the desired message index. This scheme is of particular significance when there is only one accessible database—a special case that turns out to be more challenging for PIR in the multi-database case. The upper bound for privacy-preserving capacity for these scenarios is $C = (1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}})^{-1}$, where K is the number of messages and S represents the quantity of information sources such as $S = N + U - 1$ for U users and N databases. We show that the proposed information retrieval scheme attains the capacity bound even when only one database is present, which differs from most existing works that hinge on the access to multiple databases in order to hide user privacy. Unlike the multi-database case, this scheme capitalizes on the inability for a database to cross-reference queries made by multiple users due to computational complexity.

Index Terms—Private information retrieval, capacity, multiple users

I. INTRODUCTION

The area of Private Information Retrieval (PIR) focuses on obtaining information from various databases while protecting user privacy. The goal of PIR is to maintain privacy while retrieving the largest amount of information possible while downloading the minimum quantity of irrelevant information necessary in order to obfuscate the desired message index. In the classical PIR setting [1], there are multiple non-communicating and replicated databases that contain the same messages from which a user may download a particular message of interest. When multiple identical databases are present, a user can download different segments of the desired message from different databases and then patch them into a complete message, thus hiding its intent for all databases. For this purpose, the user prepares a query for each database, such that the queries do not reveal the user's intent. From the perspective of information theory, the goal of query design is to maximize retrieval efficiency. Our motivation behind studying this problem in PIR stems from the problem of privacy-preserving information retrieval with a single database. We use capacity to determine how efficient we can create a

PIR scheme in terms of the ratio of desired information bits to the total quantity of information amassed by a single user.

There has been considerable progress on topics that involve databases trying various strategies to compromise privacy, such as databases sharing information with other databases to compromise the privacy of a user, known as colluding databases [2]. When up to T of the N databases are colluding, the setting is known as "T-private PIR" (TPIR). If all databases are colluding, then capacity for a single user is equivalent to the setting when accessing a single database, which is a major issue encountered in PIR. What has not been considered by [3] is the presence of multiple users working together to obfuscate the message index while accessing information. This is known as "single-server multi-user PIR" (SSMUPIR), for which there is little prior work [4]. This solution can be expanded to solve other problems, such as systems with multiple databases. While [4] has successfully proven the feasibility for a multi-user setting, the maximum retrieval rate has yet to be found. There has been prior work for single-server PIR in [5] and becomes feasible when communication complexity satisfies a particular threshold. However, there has not yet been any systematic research on how a retrieval scheme can be designed for an individual user to interact with multiple users and an arbitrary number of databases while maintaining privacy.

This paper aims to solve the capacity of both the single database problem and the multi-database problems. Our solution considers the unifying setting with an arbitrary number of databases because many real-world applications create collusion across networked databases. When instantiated to the little-studied single database setting, our novel approach derives from the observation that a setting with one database and multiple users can be mapped as a mirrored version of [6]. This approach allows us to design a retrieval scheme and analyze maximum retrieval efficiency. We refer to the setting of multiple users accessing multiple databases as multi-user PIR (MUPIR). Our setting for introducing MUPIR stems from the basis for network coding in [7] to improve throughput via creating multiple channels. In our setting, we create more channels by increasing the number of users that can interact with a fixed set of databases. We focus on the problem created by databases examining queries made by a single

This work was partly supported by the US NSF grant # SaTC-1704274.

user and determine that potential information sources, users and databases, should be treated as separate random variables. Along this line, [8] demonstrates how three random variables could be used for characterizing capacity, which could be useful for creating an extension of the work presented by [6]. A query must be made in order to move information from a database to a user, which contains a linear combination of bits from multiple sources of information.

The ensuing paper is organized as follows. Section II defines the problem of interest. Section III presents the capacity of MUPIR, followed by a scheme that achieves this capacity. Section IV formally proves the capacity of MUPIR, and Section V demonstrates that the proposed scheme for MUPIR reaches the derived capacity. Concluding remarks are included in Section VI.

Notation: To represent the set of answers $\{A_1^{[\theta]}, \dots, A_U^{[\theta]}\}$, θ being the desired message index and U as the number of users, we use $A_{1:U}^{[\theta]}$. For $\{A_{1:U}^{[1]}, \dots, A_{1:U}^{[K]}\}$, the notation $A_{1:U}^{[1:K]}$ is applied. To represent $Q_{1:S}^{[\theta]}$, we employ the compact form $\sum Q$. For any positive integer A , we denote an index set using $[A] = \{1, 2, \dots, A\}$. We utilize the notation $Q \sim A$ to indicate Q and A are identically distributed. Let $[U] \setminus u = [U']$, so that $u \notin [U']$.

II. PROBLEM FORMULATION

Consider a network of U users who seek to access messages contained in N databases. It is assumed that all of the N databases contain the same K messages W_k , $\forall k \in [K]$. Each message W_θ is made of L bits, which can be downloaded bit by bit and then patched together to reconstruct the original message. A user seeks to retrieve message W_θ by having a user generate a total of S queries for message θ , denoted as $Q_s^{[\theta]}$, sent to all $S = N + U - 1$ sources. Each source answers the original user with $A_s^{[\theta]}$, $\forall s \in [S]$, and allows for the desired message W_θ to be obtainable from a set of D bits. The scheme utilized to retrieve information, $Q(s, \theta)$, generates $Q_s^{[\theta]}$. When query $Q_s^{[\theta]}$ is sent to user u , then the query is forwarded to a single, arbitrary database n . Answer $A_s^{[\theta]}$ is returned to user u , who returns the answer to the original user. If query $Q_s^{[\theta]}$ is sent to database n , then $A_s^{[\theta]}$ is directly returned to the original user.

Our objective is to retrieve a message W_θ from a database and other users without revealing the value of the message index, θ , while obtaining the message at the maximum possible rate. We define rate as a ratio $R = L/D$, where L is the length of the intended message W_θ , and D is total number of downloaded bits. In order for a single user to hide the message index from $U - 1$ users and N databases, all queries generated by $Q(s, \theta)$ must be identical in structure for the same sets of K distinct messages. This leaves us to readapt the privacy constraint defined by [5, eq. (8)] as

$$(Q_s^{[1]}, A_s^{[1]}, W_{1:K}) \sim (Q_s^{[k]}, A_s^{[k]}, W_{1:K}), \forall s \in [S], k \in [K]. \quad (1)$$

Additionally, to maintain privacy, each query $Q_s^{[\theta]}$, $\forall s \in [S]$ must be independent of the desired message index, θ . We

express this by evaluating the mutual information, $I(\cdot; \cdot)$, between $Q_{1:S}^{[k]}$ and $W_{1:K}$, ensuring there is no definite overlap between the queries and messages. We formally express this as defined in [2, eq. (3)]:

$$I(Q_{1:S}^{[k]}; W_{1:K}) = 0, \quad \forall k \in [K]. \quad (2)$$

To represent the size of message W_θ in terms of bits, we use the binary entropy function $H(W_\theta)$. We express the size of W_θ related to the size of other messages by using index k in the form

$$H(W_\theta) = H(W_k) = L, \quad \forall k \in [K]. \quad (3)$$

The purpose of finding capacity in PIR is to find the maximum rate of retrieval, which focuses on obtain the largest possible message by downloading the smallest amount of bits while satisfying the criterion for privacy. The total size (in bits) of all K messages can be equivalently represented as

$$\sum_{k=1}^K H(W_k) = K H(W_\theta) = H(W_{1:K}). \quad (4)$$

The total quantity of downloaded information by a central user from S sources is defined as $D = SH(A_1^{[k]})$. Alternatively, for desired message W_θ , $D = SH(A_1^{[\theta]})$ is downloaded. We are not concerned with the upload cost for sending queries, so we do not directly take into account the magnitude of $H(Q_s^{[\theta]})$. In order to decode the quantity information withheld in an answer, a user must have knowledge of all queries. Thus, we must evaluate the entropy of downloaded information in the form $H(A_s^{[\theta]} | \sum Q)$. We adapt the definition of rate in [3]:

$$R \triangleq \frac{L}{D} = \frac{H(W_\theta)}{SH(A_1^{[\theta]} | \sum Q)} \quad (5)$$

We use the general notation C_{MUPIR} to represent capacity in a multi-user and a multi-database setting. The relation between rate and capacity can be expressed as $R \leq C_{\text{MUPIR}}$, since it is not possible to achieve a more efficient rate than the capacity. In order to design an efficient retrieval scheme, [6] states the following three principles, the first two of which are imposed to ensure user privacy, while the third principle is used to improve efficiency for retrieving information.¹

- (1) *Queries to each information source obtain forced symmetry*
- (2) *Message symmetry is enforced within all answers and queries*
- (3) *Side information of undesired messages is exploited to retrieve new desired information*

Based upon these principles, we next analyze capacity and design retrieval algorithms for databases to address capacity-achieving solutions for classical PIR in order to evaluate the computational cost of thwarting our scheme. Our designs are developed for two scenarios: One is when the databases honestly respond to queries but do not analyze the queries, and the other is when databases can analyze and cross-reference

¹Each enumerated item corresponds to arrows indicating different steps of generating queries.

queries from multiple users to determine the user's desired message index. In order to determine our scheme's usefulness, we must determine the computational cost associated with analyzing the query schemes. Evaluating Algorithms 1 and 2 of Appendices A and B, it is feasible to compute the desired message index with a computational complexity of $O(n^2)$. Due to the similarities between Algorithms 1 and 2, n would have to correspond to the length of each query sent to a database. If a database does have the option to cross-reference queries, then the remaining set of $U - 1$ would be forced to replicate all contents stored in a database to guarantee privacy. Given the scope of our problem, we do not take into consideration the efficiency of replicating all K messages across all $U - 1$ users.

III. MAIN RESULTS

A. PIR Capacity for Independent Databases

We start with the first case assuming that users cannot collude and the databases are non-communicating. Under such assumptions, there has not been a great deal of research performed with multi-user PIR settings, specifically addressing multi-user and multi-database settings. We find that users can collude together against an arbitrary number of databases while satisfying the defined privacy constraints. Our main result on the user privacy-preserving capacity is given by Theorem 1.

Theorem 1: Our capacity for MUPIR applies to settings with N non-curious or computationally-limited databases and U trustworthy users. We characterize $S = N + U - 1$ sources of information to have the capacity:

$$C_{\text{MUPIR}} = \left(1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}} \right)^{-1} \quad (6)$$

Our theorem is similar in design to the theorem for classical PIR proposed in [6], however, our novel approach focuses on adding on additional users treated like databases in order to increase retrieval rate.

B. Proposed Retrieval Scheme

A functional system with multiple users and a single database can be generated by mirroring the classical PIR problem in [6] and adopting the necessary privacy constraints. Our solution focuses on the SSMUPIR setting and its extension, the MUPIR setting. The fundamental process behind our scheme is as follows:

- 1) Generate a query scheme with other users to access information from an arbitrary number of databases
- 2) Send queries to the database(s) and other users
- 3) Receive and combine answers from the database(s) and other users

A visual model of the mirrored scheme we create is shown in Fig. 1. The upper bound for capacity in a setting with a single user is achieved by a scheme utilized in [6]. Calling function $Q(s, \theta)$ results in " k -sums," generated by the equation $\sum_{k=1}^K (S-1)^{k-1} \binom{K}{k}$. There are also two other options for the $U - 1$ users

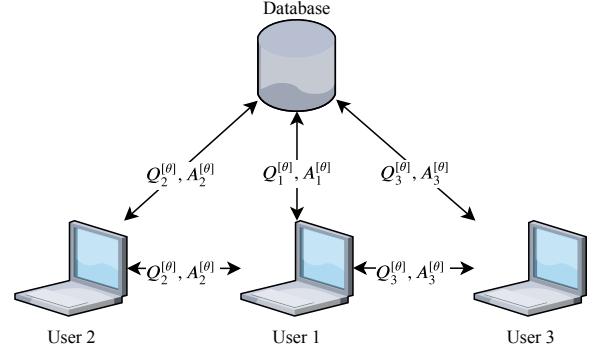


Fig. 1. A visual depiction of a user attempting PIR with two other users and a database.

that are considered: In some contexts, all $U - 1$ users must replicate all contents stored within a select database in order for a central user to preserve privacy from other users. This is necessary when a database cross-references queries made by multiple users and wants to see if there are any colluding users and reveal a potential desired message index. If a database is computationally bounded and cannot analyze queries made by all users, then there is no need for $U - 1$ users to replicate the entirety of the database with their own means of storage. Our scheme would have to be limited in application to settings with databases that cannot afford to cross-reference queries from all users. We create another example in Fig. 2 while assuming the three defined principles are followed. Consider a simple PIR setting with a single database ($N = 1$) and two users ($U = 2$). Our scheme begins by having an arbitrary user (U1) request a portion of a desired message bit of W_1 from the database DB1, which in this case is a_1 . The first principle (1) is applied to result in the other user (U2) to request bit a_2 . Applying the next principle (2) results in bits b_1 and b_2 being requested by each respective source. Principle (3) is responsible for bits b_1 and b_2 being utilized as side information and compressed in each users' queries to allow for efficiency. By randomly requesting each bit from a message to create an equal probability of all message indices being desired, privacy is preserved.

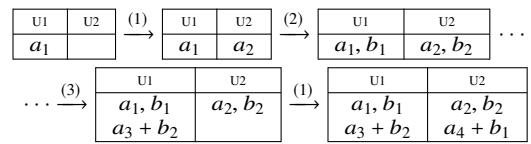


Fig. 2. Demonstration of Query Generation for $K = 2$, $N = 1$, and $U = 2$.

In this setting, a message of size $L = 4$ bits is recovered from a set of $D = 6$ downloaded bits. Thus, a rate of $R = \frac{L}{D} = \frac{4}{6} = \frac{2}{3}$ is achieved and satisfies the capacity in Theorem 1. To prove that rate increases, we move forward by formally proving the relationship with N databases and U users with capacity.

IV. PROVING CAPACITY

For the original setting of SSMUPIR, our solution stems from setting $N = 1$ and $S = U$ in eqn. 6, yielding the capacity

for SSMUPIR. We formally prove the capacities of SSMUPIR and MUPIR by finding the quantity of desired information, $H(W_\theta)$ and the cardinality of $Q(s, \theta)$. We apply this to the definition of rate in (5) in order to find capacity.

Lemma 1: $\forall S \in \mathbb{N}_{>0}$ and $\theta \in [K]$, message W_θ can be successfully retrieved while maintaining privacy from all sources and achieve $R \leq (1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}})^{-1}$.

Proof: For simplicity, let $\theta = 1$ and $K = 1$, to obtain message W_1 . Building off the previous Lemma, we characterize the capacity of MUPIR with respect to preserving privacy from other users, utilizing the function $I(A_{1:U}^{[1]}; W_1)$. Similar to the the setting for SSMUPIR, the identity for $H(A_n^{[1]} | \sum Q) \neq H(A_u^{[1]} | \sum Q)$ applies to settings where both U and N are greater than 1, which is characterized by $I(A_{1:U}^{[1]}; A_{1:N}^{[1]}; W_1)$ for solving capacity.² We evaluate the upper bound of mutual information to be:

$$I(A_{1:U}^{[1]}; A_{1:N}^{[1]}; W_1 | \sum Q) \leq I(A_{1:U}^{[1]}; A_{1:N}^{[1]} | \sum Q) \quad (7)$$

Evaluating the boundary of mutual information is critical for solving capacity. Typically, for a setting of U users interfacing with a single database, $I(A_1^{[1]}, A_{2:U}^{[1]}; W_1 | \sum Q)$ could be used to evaluate how much information is downloaded. The combination of U users and N databases requires a different approach to be taken with an allowance for more random variables. For the case of a single user and a single database, the quantity of downloaded information is determined by $I(A_1^{[1]}, A_1^{[1]}; W_1 | Q) \leq H(A_1^{[1]} | \sum Q)$. This is the result of only one source of new information, rather than two sources, due to a single user being unable to generate new information independently. This distinction causes us to substitute $I(A_{1:U}^{[1]}; A_{1:N}^{[1]} | \sum Q)$ with $I(A_{2:U}^{[1]}; A_{1:N}^{[1]} | \sum Q)$. In order to find our desired result, we must minimize conditional entropy so that $H(A_{2:U}^{[1]} | A_{1:N}^{[1]}) = H(A_{1:N}^{[1]} | A_{2:U}^{[1]}) = 0$ to achieve

$$I(A_{2:U}^{[1]}; A_{1:N}^{[1]} | \sum Q) = H(A_{2:U}^{[1]}, A_{1:N}^{[1]} | \sum Q) \quad (8)$$

We combine answers $A_{2:U}^{[1]}$ and $A_{1:N}^{[1]}$ to create $A_{1:S}^{[1]}$ to represent new information from an equivalent quantity of S sources. We conveniently characterize the new capacity starting from

$$H(W_1) = H(W_1 | \sum Q) \leq I(A_{1:S}^{[1]}; W_1 | \sum Q) \quad (9)$$

From here, we can apply the same techniques for entropy conditioning used in [9] by evaluating $I(A_1^{[1]}, A_{2:S}^{[1:K]}; W_{1:K} | \sum Q)$ with the assumption that $K = 3$. By induction, we obtain our capacity for the MUPIR with preservation of privacy from all $S = N + U - 1$ sources:

$$R \leq \left(1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}}\right)^{-1} \quad (10)$$

This concludes our proof for capacity of all PIR settings with multiple users or databases. ■

²A brief overview of how we utilize multivariate mutual information is included in Appendix C.

V. ACHIEVABILITY OF THE PROPOSED SCHEME

Lemma 2: $\forall \theta \in [K]$, and $s \in [S]$, the structure of $Q(u, \theta)$ achieves the cardinality $|Q(s, \theta)| = S^{K-1} + \frac{1}{S-1}(S^{K-1})$ in a single database setting. The deterministic function can be extended to utilize settings where $N > 1$ by generating $(N + U - 2)^{k-1}$ instances for each k -sum.

Proof: The query scheme used to address this problem is nearly identical to the k -sum equation presented in [6]. A necessary modification is that $(U - 1)^{k-1}$ instances for each k -sum must increase for each database present to generate $(N + U - 2)^{k-1}$ instances.

$$|Q(u, \theta)| = |Q(s, \theta)| = \sum_{k=1}^K \binom{K}{k} (S-1)^{k-1} \quad (11)$$

$$= S^{K-1} + \frac{1}{S-1}(S^{K-1} - 1) \quad (12)$$

We now know the relationship between downloaded information and replicated databases and can determine rate. ■

With the length of the desired message and cardinality of downloaded information known, we can determine the information-theoretic rate.

Lemma 3: The rate achieved is equal to the capacity, $C_{\text{MUPIR}} = (1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}})^{-1}$.

Proof: Assuming the cardinality of $Q(s, \theta)$ presented in Lemma 5 is correct, the result can applied to the definition of rate to find that the scheme is capacity-achieving.

$$R \triangleq \frac{L}{D} = \frac{S^K}{S|Q(s, \theta)|} = \frac{S^{K-1}}{\sum_{k=1}^K \binom{K}{k} (S-1)^{k-1}} \quad (13)$$

$$= \left(1 + \frac{\frac{1}{S}(1 - \frac{1}{S^{K-1}})}{1 - \frac{1}{S}}\right)^{-1} = \left(1 + \frac{1}{S} + \dots + \frac{1}{S^{K-1}}\right)^{-1} \quad (14)$$

The rate achieved by (14) is equal to the capacity presented in Theorem 1. ■

VI. CONCLUSION

Our adaptation of the original scheme presented in [6] shows that there is a great deal of potential for undermining systems of colluding databases by introducing other users as potential sources of information. We are not surprised that the capacity of the three possible scenarios in focus is identical. Intuitively, we would expect there to be a lower capacity in other scenarios where there are malicious users or databases having knowledge of user collusion. We have successfully contributed to the field of PIR by 1) finding the upper bound of capacity for settings with multiple users in a single database setting, 2) characterized the capacity for a user for settings with a user interacting with other users alongside multiple databases, and 3) creating a PIR scheme that can be utilized in all of these settings. However, this scheme is only guaranteed to protect privacy if the database analyzes queries made by a single user at a time due to computational costs. If the database in question cross-references all queries made by all users, then privacy is at risk of being compromised. By solving this issue, a breakthrough can be made in PIR for securing privacy against many networked databases.

APPENDIX A

ALGORITHM FOR BRUTE FORCE ANALYSIS OF A QUERY MADE BY AN INDIVIDUAL USER TO FIND THE DESIRED MESSAGE INDEX

Algorithm 1 Input: Query Set $Q(\text{User}, \theta)$. Output: θ .

Require: All queries from a distinct user to a database are complete.

- 1: Initialize: Associative array μ containing all bits for K messages as keys with their respective indices mapped as values $\forall k \in [K]$. Initialize array β of length K containing all zeroes.
- 2: **for all** $\alpha \in Q(\text{User}, \theta)$ **do** {For each bit...}
- 3: **if** $\alpha \notin \mu$ **then** {...if a bit is unidentifiable...}
- 4: **for all** $\alpha' \in Q(\text{User}, \theta)$ **do** {...then compare to all other bits...}
- 5: **if** $(\alpha \oplus \alpha') \in \mu$ **then** {...if an identifiable bit is generated...}
- 6: $\beta[\mu[\alpha \oplus \alpha']] += 1$ {...increment the respective index in the list for storing the numbers of identified bits.}
- 7: **end if**
- 8: **end for**
- 9: **else** {If the bit is already distinguishable...}
- 10: $\beta[\mu[\alpha]] += 1$ {...increment the respective index...}
- 11: **end if**
- 12: **end for**
- 13: **return** $\max(\beta)$ {...and end by returning θ .}

APPENDIX B

ALGORITHM FOR BRUTE FORCE ANALYSIS OF QUERIES MADE BY MULTIPLE USERS TO FIND THE DESIRED MESSAGE INDEX

Algorithm 2 Input: Query Sets $\sum Q$. Output: θ .

Require: All queries from a distinct set of users to a database are complete.

- 1: Initialize: Associative array μ containing all bits for K messages as keys with their respective indices mapped as values $\forall k \in [K]$. Initialize array β of length K containing all zeroes.
- 2: **for all** $u \in [U]$ **do** {For each user...}
- 3: **for all** $\alpha \in Q(u, \theta)$ **do** {...analyze each bit...}
- 4: **for all** $u' \in [U']$ **do** {...by comparing to other users...}
- 5: **if** $\alpha \notin \mu$ **then** {...if a bit is unidentifiable...}
- 6: **for all** $\alpha' \in Q(u', \theta)$ **do** {...compared to all other bits...}
- 7: **if** $(\alpha \oplus \alpha') \in \mu$ **then** {...if an identifiable bit is generated...}
- 8: $\beta[\mu[\alpha \oplus \alpha']] += 1$ {...update the respective index count in the list for storing the numbers of identified bits.}

- 9: **end if**
- 10: **end for**
- 11: **else** {If the bit is already distinguishable...}
- 12: $\beta[\mu[\alpha]] += 1$ {...increment the respective index...}
- 13: **end if**
- 14: **end for**
- 15: **end for**
- 16: **end for**
- 17: **return** $\max(\beta)$ {...and return θ .}

APPENDIX C

CHARACTERIZATION OF MULTIVARIATE MUTUAL INFORMATION

To characterize $I(A_{1:U}^{[1]}; A_{1:N}^{[1]}; W_1)$, we apply the following: From Theorem 2 of [8], the upper bound for mutual information utilizing three random variables (X , Y , and Z) is

$$I(X; Y; Z) \leq \min\{I(X; Y), I(X; Z), I(Y; Z)\}$$

This identity stems from using mutual information in the form $I(X; Y)$. The upper bound of interaction information can be expressed as

$$H(W_1) \leq \min\{I(A_{1:U}^{[1]}; A_{1:N}^{[1]}), I(A_{1:U}^{[1]}; W_1), I(A_{1:N}^{[1]}; W_1)\}$$

By definition, $I(A_{1:U}^{[1]}; W_1)$ and $I(A_{1:N}^{[1]}; W_1)$ must be maximized in order to achieve capacity, thus obtaining:

$$H(W_1) \leq I(A_{1:U}^{[1]}; A_{1:N}^{[1]})$$

REFERENCES

- [1] B. Chor, O. Goldreich, and E. Kushilevitz, "Private Information Retrieval," *Journal of the ACM*, vol. 45, no. 6, p. 18, Nov. 1998.
- [2] H. Sun and S. A. Jafar, "The Capacity of Robust Private Information Retrieval With Colluding Databases," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2361–2370, Apr. 2018.
- [3] —, "The Capacity of Private Information Retrieval with Colluding Databases," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec. 2016, pp. 941–946.
- [4] S. Li and M. Gastpar, "Single-server Multi-user Private Information Retrieval with Side Information," in *2018 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 1954–1958.
- [5] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*. Miami Beach, FL, USA: IEEE Comput. Soc, 1997, pp. 364–373. [Online]. Available: <http://ieeexplore.ieee.org/document/646125/>
- [6] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.
- [7] R. Ahlswede, Ning Cai, S.-R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
- [8] R. Yeung, "A New Outlook on Shannon's Information Measures," *IEEE Transactions on Information Theory*, vol. 37, no. 3, pp. 466–474, May 1991.
- [9] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.