On the Security of a Code-Based PIR Scheme

Svenja Lage and Hannes Bartz

Institute of Communications and Navigation,
German Aerospace Center (DLR)
{svenja.lage, hannes.bartz}@dlr.de

Abstract

Private Information Retrieval (PIR) schemes enable clients to retrieve files from a database without revealing the identity of the requested file to the server. In this paper, we prove a critical vulnerability in the code-based PIR scheme CB-cPIR. Beyond theoretical analysis and an implementation of the attack, we evaluate the CB-cPIR scheme with adjusted parameters against state-of-the-art alternatives and demonstrate that it is no longer competitive.

Key Words : Private Information Retrieval, Code-based Cryptography, Cryptanalysis

1 Introduction

Private Information Retrieval (PIR) schemes, as introduced in [CKGS98] and [KO97], enable a client to request a particular file from a database without disclosing the identity of the requested file to the server. A straightforward solution is for the user to download the entire database. Although this approach preserves the user's interest, it is clearly impractical for large databases. However, as demonstrated in [CKGS98], it represents the only possibility for information-theoretically secure PIR in the single-server scenario. As an alternative for single-server scenarios, various computationally secure PIR (cPIR) schemes based on computational hard problems have emerged, providing a trade-off between security and communication cost.

In the context of post-quantum cryptography, it is essential not only to consider problems that are presumed to be computationally hard on classical computers but also to develop cryptographic protocols that can withstand attacks from potential quantum computers. To address this challenge, one viable approach is to construct PIR schemes based on lattice problems, which are widely regarded as being resistant to quantum attacks. While

many current PIR schemes rely on lattice problems, such as the Learning With Errors (LWE) problem or its ring variant (see [HHCG$^+$23], [BV11] or [AMBFK16]), code-based cryptography presents a promising alternative to lattice-based schemes.

One approach, which defines a PIR scheme based on hard problems in coding theory, was proposed in [HHWZ20]. However, an efficient attack on this scheme was presented in [BL21]. In this paper, we study the code-based computationally secure CB-cPIR scheme described in [VH24] and [HV25a], which is an adapted version of the original construction [HHWZ20] designed to be resistant to the attack in [BL21]. In [VH24] and [HV25a], the authors present their CB-cPIR scheme, which they thoroughly analyze for its security and compare to the state-of-the-art schemes SimplePIR [HHCG$^+$23] and XPIR [AMBFK16]. Their evaluation suggests that the modified scheme offers improved communication and computational costs, leading them to conclude that it represents a significant step towards practical and scalable real-world PIR based on coding theory.

In this paper, we present a critical attack against the CB-cPIR scheme, which significantly undermines the security levels claimed in [HV25a] for the proposed parameters. To validate our analysis, we not only provide theoretical results but also implemented the attack. While adapting the parameters can mitigate this vulnerability, our analysis reveals that the resulting rates of the CB-cPIR scheme are no longer favorable against those of XPIR. In contrast, the comparison with SimplePIR yields a more nuanced result, with the choice between the two schemes depending on the specific requirements and constraints of the particular underlying use case.

2 Preliminaries

Let q be a prime or a power of a prime. Denote the finite field of order q by $\mathbb{F}_q$, and its extension field of degree s by $\mathbb{F}_{q^s}$. We denote by $\mathbb{F}_q^\times = \mathbb{F}_q \setminus \{0\}$ the set of nonzero elements in $\mathbb{F}_q$. If $\Gamma = \{\gamma_1, \ldots, \gamma_s\}$ is a basis of $\mathbb{F}_{q^s}$ over $\mathbb{F}_q$, then denote by $V = \langle \gamma_1, \ldots, \gamma_v \rangle$ for $1 \leq v \leq s$ the v-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^s}$ spanned by $\gamma_1, \ldots, \gamma_v$. For a matrix $A \in \mathbb{F}_q^{m \times n}$, denote by $A_{[i:j,:]}$ the submatrix of A consisting of rows i to j, and by $A_{[:,i:j]}$ the submatrix consisting of columns i to j. Following the usual convention, we define the Kronecker product of

$$A = (a_{i,j})_{\substack{i=1,\ldots,m, \\ j=1,\ldots,n}} \in \mathbb{F}_q^{m \times n}$$

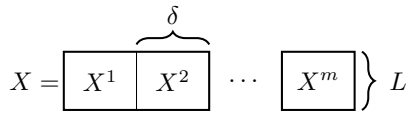
and $B \in \mathbb{F}_q^{v \times w}$ as

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{pmatrix} \in \mathbb{F}_q^{mv \times nw}.$$

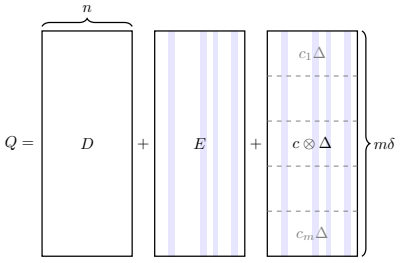
A linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of length n and dimension k is a k-dimensional linear subspace of $\mathbb{F}_q^n$. Every linear code $\mathcal{C}$ can be represented by a generator matrix $G \in \mathbb{F}_q^{k \times n}$ of rank k. Any set of k linearly independent columns of G forms an information set of the code.

3 The CB-cPIR scheme

Let v, s, n, and k be scheme parameters satisfying $v < s$ and $k < n$, and define $\delta = (s-v)(n-k)$. Suppose the database consists of m files. Each file X^i is represented as a matrix $X^i \in \mathbb{F}_q^{L \times \delta}$, where L is chosen sufficiently large to accommodate all files. The entire database is represented as $X \in \mathbb{F}_q^{L \times m\delta}$, as illustrated in Figure 1a.



(a) Database construction with m files X^i, $i = 1, \dots, m$.



(b) Construction of Q as part of the query.

Figure 1: Database layout and query construction in the CB-cPIR scheme from [VH24].

Query generation: To request file X^{i_0} for some $i_0 \in \{1, \dots, m\}$, the user performs the following steps.

- Sample uniformly at random a vector $\beta \in (\mathbb{F}_q^\times)^m$ and let $c = e_{i_0} + \beta$, where $e_{i_0} \in \mathbb{F}_q^m$ is the i_0-th unit vector.
- Choose an $[n, k]$-linear code $\mathcal{C} \subseteq \mathbb{F}_{q^s}^n$ and construct a matrix $D \in \mathbb{F}_{q^s}^{m\delta \times n}$, where each row is a codeword from $\mathcal{C}$. Randomly choose an information set $I \subset \{1, \dots, n\}$ of $\mathcal{C}$ with $|I| = k$.
- Choose a random basis $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_s\}$ of $\mathbb{F}_{q^s}$ over $\mathbb{F}_q$. Let $V = \langle \gamma_1, \dots, \gamma_v \rangle$ be the $\mathbb{F}_q$-linear, v-dimensional subspace of $\mathbb{F}_{q^s}$ spanned by the first v basis vectors, and let $W = \langle \gamma_{v+1}, \dots, \gamma_s \rangle$. Randomly choose $\tilde{E} \in V^{m\delta \times (n-k)}$ and extend it to a matrix $E \in V^{m\delta \times n}$ by inserting zero-columns at the positions of the information set I.

- Randomly choose $\tilde{\Delta} \in W^{\delta \times (n-k)}$ with full row-rank over $\mathbb{F}_q$, i.e., $rk_{\mathbb{F}_q}(\tilde{\Delta}) = \delta$, and extend $\tilde{\Delta}$ to $\Delta \in W^{\delta \times n}$ by inserting zero-columns at the positions of the information set I.
- As a first part of the query, set $Q = D + E + c \otimes \Delta$.
- Choose another linear code $\mathcal{C}_\beta$ with an information set I_β and define subspaces V_β and W_β to generate matrices D_β, E_β and Δ_β according to the same procedure.
- Set $Q_\beta = D_\beta + E_\beta + \beta \otimes \Delta_\beta$.

The user sends Q and Q_β to the server. An illustration of the query generation in the scheme is shown in Figure 1b.

Answer: The server computes the product $R = X \cdot Q \in \mathbb{F}_{q^s}^{L \times n}$. Additionally, it computes the product $R_\beta = X \cdot Q_\beta$ and transmits both matrices to the client.

File extraction: Note that each row of the response R consists of a codeword and two errors supported outside of the information set I. Using decoding techniques, the user obtains the error only. By projection onto W, he deletes the error introduced by the matrix E. Proceeding similarly with the rows of R_β, the user can cancel out the β-dependent part and can reconstruct the desired file (see [VH24, HV25a] for details).

The authors of [VH24] and [HV25a] further proposes a modification to enhance the rate of the scheme. This modification allows for f files to be requested simultaneously using the same vector β, thereby enabling the client to store R_β at the client's side in the first query. Subsequent queries no longer require the transmission of Q_β, thereby improving the overall rate. As stated in [VH24, Theorem 2], the PIR rate is given by

$$R_{\text{CB-cPIR}} = \frac{fL\delta \log_2(q)}{(f+1)(m\delta n + Ln)\log_2(q^s)}.$$

In scenarios where multiple files are requested concurrently, the weight of c can be reduced by the number of files requested. This reduction in weight has the potential to introduce a vulnerability, particularly for large values of f. However, it has been countered by the authors that the construction of c can always be performed in such a manner that it possesses a sufficiently high weight, thereby rendering this attack infeasible.

A thorough analysis of various attacks, including information set decoding, subspace attacks and the subquery attack from [BL21] as well as an advanced variant, was conducted in [HV25a]. This led to the proposal of

Table 1: Proposed parameters and resulting approximative rate in the case $L \gg m\delta$ and $f = 1$ for the CB-cPIR scheme in [HV25a].

q	s	v	n	k	δ	$R_{\text{CB-cPIR}}$
32	32	31	100	50	50	1/128
32	32	30	100	50	100	1/64
2^{16}	12	10	100	50	100	1/24
$2^{32} - 5$	6	4	120	60	120	1/12
2^{32}	5	3	100	50	100	1/10
$2^{61} - 1$	6	2	100	50	200	1/6

parameters with corresponding rates and security levels, which are summarized in Table 1 and Table 2. Note that the displayed rate is approximated for the scenario where $L \gg m\delta$ and $f = 1$, resulting in $R_{\text{CB-cPIR}} \approx \frac{\delta}{2ns}$. Furthermore, an iterative version of the scheme was introduced, which reorders the database into a square or hypercube structure, inspired by [KO97], to optimize communication costs. This approach is particularly effective when dealing with small file sizes in large databases. In comparison to the recent lattice-based protocols SimplePIR [HHCG$^+$23] and XPIR [AMBFK16], the authors claim lower communication and computational costs.

After publishing the preprint associated with this abstract [LB25], the authors released an updated version of the scheme [HV25b]. While this slight adaptation prevents the direct application of the attack described in this work, the revised scheme remains vulnerable to related attacks. Due to space constraints, a description and security analysis of the updated version is presented in the full paper.

4 An attack on the scheme

We will now demonstrate how to compromise the CB-cPIR scheme. To achieve this, we will divide the matrix Q into blocks consisting of δ rows each as

$$Q = \begin{pmatrix} Q^1 \\ \vdots \\ Q^m \end{pmatrix} = \begin{pmatrix} D^1 \\ \vdots \\ D^m \end{pmatrix} + \begin{pmatrix} E^1 \\ \vdots \\ E^m \end{pmatrix} + \begin{pmatrix} c_1\Delta \\ \vdots \\ c_m\Delta \end{pmatrix},$$

where

$$c_i = \begin{cases} \beta_i & \text{if } i \neq i_0 \\ 1 + \beta_{i_0} & \text{if } i = i_0. \end{cases} \tag{1}$$

The attack will be carried out in two consecutive steps.

Step 1: Constructing an Auxiliary Matrix
The first step is an auxiliary step that will enable us to distinguish between blocks with and without a Δ-dependent part later on. To this end, we construct a matrix $A \in \mathbb{F}_{q^s}^{(ns-\delta+1)\times n}$ with a similar structure to Q, but with low rank. Assume that the number of files m in the database fulfills $m > ns-\delta$. Then A consists of the first row of each of the submatrices $Q^1, \dots, Q^m$ as displayed in Figure 2. Observe that A shares a similar structure with Q, as each row comprises a codeword from the linear code plus an error term from V in the columns outside the information set. However, unlike Q, every row of A features the same vector from Δ, scaled by different factors from $\mathbb{F}_q$. Hence, following the same arguments as in [BL21, Corollary 3.2] and interpreting A as a matrix in $\mathbb{F}_q^{m\times ns}$, we have $rk_{\mathbb{F}_q}(A) \leq ns - \delta + 1$. The rank mirrors the dimension of the code, the error matrix E, and the first row of Δ. To reduce computational complexity, we restrict A to the first $ns-\delta+1$ rows.

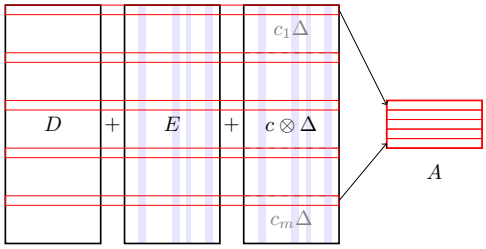


Figure 2: Visualization of the first step within the attack: Construction of the auxiliary matrix A.

Step 2: Identify the index of interest
We now proceed to analyze each block $Q^1, \dots, Q^m$ individually with the objective of reconstructing the index of interest i_0. Let $J = \{1, \dots, m\}$ be the set of indices that are candidates for the index of interest. We randomly select two indices $i, j \in J$ and apply the following procedure to determine whether one of these indices coincides with i_0.

Step 2.1: Find $\alpha \in \mathbb{F}_q$ such that $\alpha c_i + c_j = 0$.
To this end, consider the matrix

$$D^{i,j} = D^{i,j}(\alpha_2, \dots, \alpha_\delta) = \begin{pmatrix} \alpha_2 Q^i_{[2,:]} + Q^j_{[2,:]} \\ \vdots \\ \alpha_\delta Q^i_{[\delta,:]} + Q^j_{[\delta,:]} \end{pmatrix}$$

for different values $\alpha_2, \dots, \alpha_\delta \in \mathbb{F}_q$. Note that if the equation $\alpha_k c_i + c_j = 0$ holds in $\mathbb{F}_q$ for some $k \in \{2, \dots, \delta\}$, then the k-th row of $D^{i,j}$ will not have a Δ-dependent component. In this case, appending this row to A will not increase the rank of A. On the other hand, if the row has a Δ-dependent

component, then appending it to A will increase the rank by one with very high probability. We can efficiently test all values of $\alpha_2, \ldots, \alpha_\delta$ simultaneously by simply appending $D^{i,j}$ to A. If the resulting matrix has rank $rk_{\mathbb{F}_q}(A) + \delta - 1$, then none of the tested values satisfies our condition. Conversely, if the rank is less than $rk_{\mathbb{F}_q}(A) + \delta - 1$, then, given that Δ has full rank by definition, there is one $\alpha \in \{\alpha_2, \ldots, \alpha_\delta\}$ fulfilling the equation.

In the worst-case scenario, we need to calculate the rank of an $\mathbb{F}_q^{ns \times ns}$ matrix $\lceil \frac{q}{\delta - 1} \rceil$ times to identify the tuple $(\alpha_2, \ldots, \alpha_\delta)$ that contains the correct value of α. To pinpoint this value from the tuple, we can employ a binary-search-like algorithm, as introduced in [Lag25] for an attack on a different PIR scheme. Thereby, we can identify the value of α from the right tuple in at most $\log(\delta)$ rank calculations of a matrix in $\mathbb{F}_q^{ns \times ns}$.

Step 2.2: Check if $i_0 \in \{i, j\}$:
Determining the value of $\alpha \in \mathbb{F}_q$ with

$$0 = \alpha c_i + c_j = \begin{cases} \alpha\beta_i + \beta_j & \text{if } i_0 \neq i \text{ and } i_0 \neq j \\ \alpha\beta_i + 1 + \beta_{i_0} & \text{if } i_0 = j \\ \alpha(1 + \beta_{i_0}) + \beta_j & \text{if } i_0 = i \end{cases}$$

does not directly yield c_i or c_j. Nevertheless, the relation between c_i and c_j can be utilized to verify whether the index of interest is equal to i or j. To achieve this, we repeat Step 1 with Q_β instead of Q and obtain a matrix $A_\beta \in \mathbb{F}_{q^s}^{(ns-\delta+1) \times n}$ with $rk_{\mathbb{F}_q}(A_\beta) \leq ns - \delta + 1$. Subsequently, we append the vector $\alpha Q^i_{\beta_{[2,:]}} + Q^j_{\beta_{[2,:]}}$ to A_β. If the rank of the resulting matrix increases by one, then it follows that $\alpha\beta_i + \beta_j \neq 0$, which implies $i_0 \in \{i, j\}$. In this case, repeat Step 2 with the same value of i but a different value of j to check if $i_0 = i$ or $i_0 = j$. Conversely, if the rank does not increase, then $\alpha\beta_i + \beta_j = 0$, which implies that $i_0 \neq i$ and $i_0 \neq j$. We can thus exclude these indices from consideration and repeat the procedure using the smaller index set $J \setminus \{i, j\}$. It is worth noting that each block can be analyzed independently, rendering the attack highly parallelizable. To optimize computational efficiency, a precomputation phase is initiated, transforming matrix A into row echelon form. In each subsequent iteration, instead of performing a complete rank calculation, only a check is necessary to verify if individual vectors reside within the precomputed row space. Hence, the overall cost is dominated by the number of iteration steps, which is in line with similar approaches in [BL21] and [HV25a].

Theorem 1. *The above algorithms reconstructs the index of interest in $O(\lceil \frac{q}{\delta - 1} \rceil)$ operations over $\mathbb{F}_q$.*

The proposed attack significantly compromises the security guarantees claimed in [HV25a]. Specifically, for the recommended parameters of the

CB-cPIR scheme, the attack complexity according to Theorem 1 counting the number of rank calculations is presented in Table 2. Note that the attack complexity is evaluated under the assumption that only one file is requested simultaneously. If, however, β is reused multiple times, the attack complexity per file decreases further.

Table 2: Attack complexity for the parameters proposed in [HV25a] in comparison to the best-known attacks from [HV25a] as security level as well as the minimal database size m required for the attack.

q	s	v	n	k	δ	Security level	Attack complexity	Minimal value of m
32	32	31	100	50	50	2^{113}	2^{1}	3151
32	32	30	100	50	100	2^{113}	2^{1}	3101
2^{16}	12	10	100	50	100	2^{113}	2^{9}	1101
$2^{32}-5$	6	4	120	60	120	2^{128}	2^{25}	601
2^{32}	5	3	100	50	100	2^{96}	2^{25}	401
$2^{61}-1$	6	2	100	50	200	2^{113}	2^{53}	401

Remark 1. *Our attack requires a database of a certain size, specifically $m > ns - \delta$. With the given parameters, this translates to a database size of* 401 *to* 3.151 *as evident from Table 2, a range that is consistent with modern database scales. However, if the database is comprised of fewer files, the attack can be adapted to accommodate smaller sizes. To achieve this, rather than extracting solely the first row from each block $Q^1, \ldots, Q^m$ to construct A, multiple rows can be utilized. By selecting $p < \delta$ rows from each block to form A, the resulting matrix will comprise pm rows, thereby relaxing the requirement to $mp > ns - \delta$. That way, our attack is applicable to databases of size at least $(\delta - 1)m > ns - \delta$, which, for the proposed parameters, translates to a minimum of approximately* 10 *files in most cases. This threshold is remarkably low, rendering the attack viable for virtually all realistic database scenarios.*

We successfully implemented the attack utilizing SageMath [S$^+$24] on a standard laptop (Intel Core i7-1370P, 1,9 GHz, 16GB RAM). Notably, for the initial two parameter sets where $q = 32$, the attack executed in a matter of seconds. However, as the value of q increases, the time complexity of our attack grows in accordance with Theorem 1. This highlights a limitation of our attack, as larger values of q may render the attack impractical due to the increased computational requirements.

5 Comparison with Lattice-Based Schemes

One natural approach to prevent our attack is to increase the field size order q. However, to avoid providing any information, the ratio $\frac{q}{\delta}$ must align with

the desired security level. While increasing q enhances security against our attack, it also significantly raises computational costs for both the server and the client, as operations in larger fields are generally more resource-intensive. Additionally, the scheme achieves higher rates when $L \gg m\delta$. As a consequence, if q is already large, the files $X^i \in \mathbb{F}_q^{L\times\delta}$ become quite substantial for reasonable rates.

In accordance with the methodology outlined in [HV25a], we aim to conduct a comparative analysis of the communication rate of the CB-cPIR protocol against established PIR schemes such as XPIR [AMBFK16] and SimplePIR [HHCG$^+$23]. The rate for the XPIR scheme is defined by

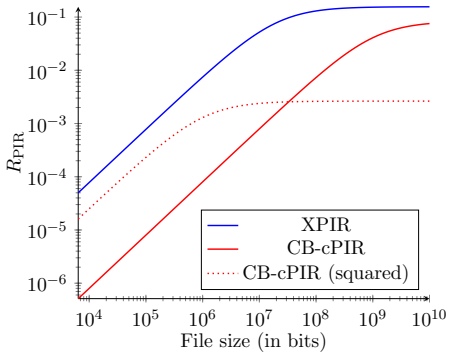
$$R_{\text{XPIR}}(F) = \frac{F}{ms_c + F\frac{s_c}{c_p}},$$

where s_p and s_c denote the sizes (in bits) of the plaintext and ciphertext, respectively, and F represents the bitsize of a single database file. For the XPIR scheme, we employ the parameters $(n, \log(q)) = (1024, 60)$, which provide a security level of 97 bits. As specified in [HV25a], the corresponding values of $s_c = 128,000$ and $s_p = 20,000$ for these parameters are utilized in the calculation. For a fair comparison, we evaluate the CB-cPIR rate using the parameters $q = 2^{104}$, $s = 6$, $v = 4$, $n = 100$, and $k = 50$, which are chosen to ensure a security level of 97, taking into account all known attacks. The communication rates for a fixed number of $m = 1,000$ files in the database, expressed as functions of the file size F (in bits), are illustrated in Figure 3a. Consistent with theoretical expectations, the CB-cPIR protocol employing a squared database demonstrates superior efficiency compared to its original counterpart for smaller file sizes. Conversely, for larger file sizes, the original CB-cPIR scheme exhibits enhanced performance. Neither variant of CB-cPIR achieves a rate comparable to that of XPIR. For small file sizes, all schemes perform worse than the trivial solution of downloading the entire database. Notably, only rates exceeding $1/m$ correspond to practical scenarios.

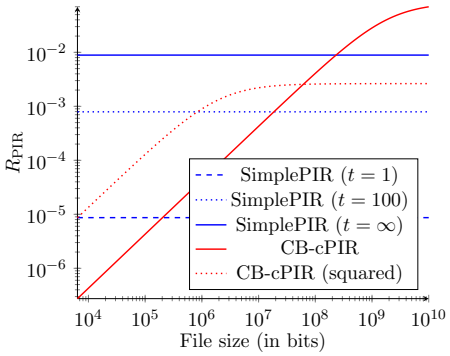
Similarly, we can compare the CB-cPIR scheme to SimplePIR. In the SimplePIR protocol, the user is required to download a one-time hint, which can be reused across multiple queries. Assuming the cost of downloading the hint is amortized over t queries, the rate of SimplePIR can be expressed as

$$R_{\text{SimplePIR}}(F) = \frac{F\log_2(p)}{(nFt^{-1}\sqrt{m} + (F + \log_2(p))\sqrt{m})\log_2(q)},$$

where F represents the file size in bits, and (q, p, n) are the scheme parameters. For our analysis, we select $(q, p, n) = (2^{32}, 495, 1024)$, which provides a



(a) XPIR rate for $(n, log(q)) = (1024, 60)$ and the rate of th CB-cPIR scheme ($q = 2^{104}$, $s = 6$, $v = 4$, $n = 100$, and $k = 50$) both for a security level of 97 bits.



(b) SimplePIR rate for $(q, p, n) = (2^{32}, 495, 1024)$ and a comparable CB-cPIR scheme rate ($q = 2^{104}$, $s = 6$, $v = 4$, $n = 120$, and $k = 60$) both for a security level of 128 bits.

Figure 3: Comparison of CB-cPIR using the original database as well as a squared version against XPIR and Simple PIR with $m = 1,000$ fixed.

security level of 128 bits. To achieve the same security level for the CB-cPIR scheme, we choose $q = 2^{135}$, $s = 6$, $v = 4$, $n = 120$, and $k = 60$.

The resulting rates for varying file sizes, with a fixed total number of files $m = 1,000$, are illustrated in Figure 3b. The case $t = \infty$ serves as a theoretical upper bound, representing the idealized scenario where the cost of downloading the hint is ignored. This limiting case provides a benchmark for the optimal performance of SimplePIR. Notably, for large file sizes, the CB-cPIR scheme may be preferred over SimplePIR. However, for smaller file sizes, the choice between the two schemes depends on how frequently the hint is reused in SimplePIR.

6 Conclusion

This paper reveals a vulnerability in the CB-cPIR scheme, demonstrating that the original parameters proposed in [HV25a] no longer align with the given security levels. While adjusting the parameters can restore the desired security level, our analysis shows that the scheme's performance is no longer competitive with state-of-the-art schemes like XPIR. Our findings highlight the need for a thorough reevaluation of the CB-cPIR scheme's design and parameters to ensure its security and performance meet the expected standards.

Acknowledgment

This work has been supported in part by funding from Agentur für Innovation in der Cybersicherheit GmbH.

References

[AMBFK16] C. Aguilar-Melchor, J. Barrier, L. Fousse, and M.-O. Killijian. XPIR: Private information retrieval for everyone. In *Proceedings on Privacy Enhancing Technologies*, pages 155–174, 2016.

[BL21] S. Bordage and J. Lavauzelle. On the privacy of a code-based single-server computational PIR scheme. *Cryptography Commun.*, 13(4):519–526, 2021.

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, 2011.

[CKGS98] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[HHCG+23] A. Henzinger, M. M. Hong, H. Corrigan-Gibbs, S. Meiklejohn, and V. Vaikuntanathan. One server for the price of two: simple and fast single-server private information retrieval. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, pages 3889–3905, USA, 2023. USENIX Association.

[HHWZ20] L. Holzbaur, C. Hollanti, and A. Wachter-Zeh. Computational code-based single-server private information retrieval. In *2020 IEEE International Symposium on Information Theory, ISIT 2020 - Proceedings*, IEEE International Symposium on Information Theory - Proceedings, pages 1065–1070. Institute of Electrical and Electronics Engineers Inc., 2020.

[HV25a] C. Hollanti and N. Verma. CB-cPIR: Code-based computational private information retrieval, 2025. https://arxiv.org/pdf/2505.03407v1.

[HV25b] C. Hollanti and N. Verma. CB-cPIR: Code-based computational private information retrieval, 2025. https://arxiv.org/pdf/2505.03407v2.

[KO97] 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*, pages 364–373, 1997.

[Lag25] S. Lage. Cryptanalysis of a lattice-based PIR scheme for arbitrary database sizes, 2025. https://arxiv.org/abs/2505.05934.

[LB25] S. Lage and H. Bartz. On the security of a code-based PIR scheme, 2025. https://arxiv.org/abs/2507.19295.

[S+24] W. A. Stein et al. *Sage Mathematics Software (Version 10.3)*. The Sage Development Team, 2024. `http://www.sagemath.org`.

[VH24] N. Verma and C. Hollanti. Code-based single-server private information retrieval: Circumventing the sub-query attack. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 2880–2885, 2024.