Private Information Retrieval with Accountability at Erasure-Optimal Server Redundancy

Myeongjun Kim, Hyunseok Jeong, Mincheol Son, and Young-Sik Kim*

DGIST, Daegu, Republic of Korea
{sanmaru98u,hsj,encrypted.def,ysk}@dgist.ac.kr

Abstract. Private information retrieval (PIR) hides which database item a client requests. We study PIR that also identifies servers that respond incorrectly. Against a coalition of at most t corrupted servers, the client must retrieve the item and identify every deviating server when at most d deviate. Above that threshold, it must abort and return an empty identification set, with a specified statistical error.

We present SynTrace, a single-round compiler for linear PIR supporting audit queries whose honest answers belong to a public code. The client combines the retrieval query with m audit queries using secret linear mixing and independent blinding. Audit syndromes identify deviations before reconstruction. We prove perfect query privacy, zero false-identification probability within the deviation threshold, and statistical realization of an accountable PIR functionality.

For Shamir-based PIR, SynTrace uses $k = t + d + 1$ servers. Under the stated privacy and download assumptions, tolerating d missing responses already requires this many servers, so identification adds no server overhead. With this number of servers, $m = d$ audit queries suffice with a suitable field, and are necessary within the specified syndrome-span decoder class. Additional audit queries can reduce communication by allowing a smaller field. For one 32 KiB record from a 1 GiB database, with five servers, privacy and robustness against two corrupted servers, and error at most 2^{-40}, SynTrace uses 2.25 MiB of communication, compared with 8.96 MiB for the homomorphic-MAC construction of Eriguchi, Kurosawa and Nuida (TCC 2022) under the same workload and thresholds. Batched retrievals can share the audit queries.

Keywords: Private information retrieval · Information-theoretic security · Accountability · Reed–Solomon codes

1 Introduction

Private information retrieval (PIR), introduced by Chor et al. [6], allows a client to retrieve an entry x_i of a database x replicated among k servers without revealing the index i. The requested entry may reveal a user's interests or intentions, even when the database itself is public. An investor, for example, may wish to

* Corresponding author.

retrieve a stock price without disclosing which stock they are considering. Similarly, a user consulting a medical information database may wish to keep the condition of interest private.

1.1 Additional Requirements for PIR

Beyond query privacy, PIR applications may benefit from additional guarantees. In this paper, we focus on *Byzantine robustness*, *accountability*, and *guaranteed abort*.

Robustness allows the client to retrieve the requested item despite a bounded number of missing or incorrect server responses. This guarantee has been studied in robust and error-correcting PIR [4,7,9,11,12,13].

In addition to recovering the requested item, a client may wish to identify the servers that returned incorrect answers, so that faulty servers can be repaired or misbehaving servers penalized or excluded from subsequent sessions. Following Cao et al. [5], we refer to this property as *accountability*. Zhang and Safavi-Naini [15] combine unconditional query privacy with computational verification and support identification of cheating servers. Identification of incorrect retrieval answers also appears in the constructions of Devet, Goldberg and Heninger [7] and Eriguchi, Kurosawa and Nuida [9].

When the number of deviations exceeds the deviation threshold, robustness may no longer be guaranteed. The property of *guaranteed abort* ensures that, even in this case, the client terminates the protocol safely by aborting without returning an incorrect item. This guarantee holds within the assumed corruption bound, except with a specified statistical error.

1.2 Existing PIR Constructions

To explain how robustness and accountability can be achieved, we first recall a multi-server PIR construction based on secret sharing. To retrieve the i-th entry, the client forms a vector e_i whose i-th entry is 1 and all other entries are 0, and sends a share $q_j = (q_{j,1}, \ldots, q_{j,n})$ of this vector to each server j. For a database $x = (x_1, \ldots, x_n)$, server j returns the linear combination

$$a_j = \langle q_j, x \rangle = \sum_{\ell=1}^{n} q_{j,\ell} x_\ell.$$

The client combines the answers to obtain the requested item, while any coalition of at most t servers learns nothing about the index. When instantiated with Shamir secret sharing, the polynomial structure of the shares also allows missing or incorrect answers to be handled through interpolation and error-correcting techniques [11].

The scheme of Devet, Goldberg and Heninger [7] provides robustness by using responses from several retrievals together. A single retrieval may not provide enough information to locate incorrect answers. Across several retrievals,

however, the same honest servers consistently supply correct answers. Their responses therefore satisfy the polynomial relations in every retrieval, giving the client additional constraints for locating errors. The decoder uses these constraints to find the response polynomials jointly, requiring them to agree with the received answers on a common set of servers. When decoding succeeds, the client obtains the requested data and identifies the erroneous responses.

Compared with decoding each polynomial separately, joint decoding can reduce the number of servers required for the same privacy and deviation bounds. With k servers, the linear joint decoder requires $k \geq t + d + 2$, where t is the maximum number of corrupted servers and d is the maximum number of deviating servers. If the initial retrieval cannot be decoded, the client sends additional queries and combines their responses with those already collected. This increases communication and can require more than one round. The client also blinds its queries with fresh secret random scalars to randomize the errors used in joint decoding.

An alternative is to verify individual answers before reconstruction, as in the homomorphic message authentication code construction of Eriguchi, Kurosawa and Nuida [9]. For each ordered pair of servers (i, j), the client adds a linear term with a random coefficient to each coordinate of the query q_i and sends the resulting polynomials to server i. The server performs the base PIR's answer computation on these polynomial inputs to obtain an authentication polynomial G_{ij}. When computed correctly, its constant term is the response to the original query q_i. Server i submits a single answer a_i together with an authentication polynomial G_{ij} for each checking server j.

Server j instead receives the values of the query polynomials at a common point α_{ij} kept secret by the client. These values form a query vector, on which server j performs the same answer computation used by server i, using its own copy of the database. It returns the resulting value σ_{ij}. Correct computation commutes with polynomial evaluation, so this value must equal $G_{ij}(\alpha_{ij})$ when both servers respond correctly. The check for the pair (i, j) passes only if both equalities hold,

$$G_{ij}(0) = a_i \qquad \text{and} \qquad G_{ij}(\alpha_{ij}) = \sigma_{ij}.$$

If server j is honest, an incorrect answer can pass both checks only if the modified authentication polynomial agrees with the correct one at this hidden point, an event whose probability is bounded by the polynomial degree and the field size.

Under an honest-majority assumption, these pairwise checks allow the client to identify servers that returned incorrect answers with high probability. The client discards their answers and reconstructs the requested item from sufficiently many remaining responses using the base PIR's reconstruction algorithm, all within one communication round. Reconstruction in its Shamir-based instantiation requires $k \geq t + d + 1$ servers. The construction uses a quadratic number of pairwise authentication components in k.

1.3 Security Definitions

PIR security is commonly analyzed by proving the desired properties separately. The query-privacy proof shows that the queries hide the requested index, while correctness ensures that honest responses yield the requested item. For error-detecting PIR, an additional guarantee limits the probability of accepting an incorrect value while allowing the client to abort. However, Alon and Beimel [1] show that these guarantees alone can still permit *selective-abort* attacks. Malicious servers can make the probability of abort depend on the requested index even though the queries reveal nothing about it. The success or failure of the retrieval can therefore leak information.

To capture this interaction, Alon and Beimel define malicious PIR using the real/ideal paradigm. In the ideal execution, a trusted party performs the retrieval. A security proof must show that every real adversary can be simulated in this ideal execution, jointly reproducing the adversary's view and the client's output. The simulator must reproduce abort behavior without knowing the requested index.

To capture accountability as well, we include the identification set L in the client's output (out, L) and compare it jointly with the adversary's view in the real and ideal executions. Here out is the retrieved value or an abort symbol. The actual deviation set $\mathcal{M}$ is determined by the queries and responses in each execution. Our definition requires L and $\mathcal{M}$ to match exactly whenever the number of deviations is within the deviation threshold, except with a specified statistical error.

1.4 Our Contributions

We present SynTrace, a compiler that adds robustness, accountability and guaranteed abort to a class of linear PIR schemes while preserving information-theoretic privacy and using one round of communication. Starting from a base PIR, the client uses its query-generation algorithm to produce the retrieval query and adds m additional queries, called *audit queries*. The client blinds each query by multiplying it by an independently sampled secret nonzero scalar, then mixes the queries for each server using a secret invertible matrix U. Each server applies the base PIR's answer algorithm to the mixed query components. The client undoes the mixing and blinding, checks the audit answers to identify deviating servers, and uses the base PIR's reconstruction algorithm on the remaining retrieval answers. This avoids the $O(k^2)$ pairwise authentication components used in the homomorphic-MAC construction of [9].

The compiler applies to linear PIR schemes that support audit queries indistinguishable from retrieval queries to any coalition of at most t servers, while allowing the client to check known relations among honest audit answers without knowing the database. Linearity allows the client to mix the queries and separate the resulting answers. Section 4.2 formalizes the required properties of the base PIR and its audit queries.

Table 1. Comparison with related PIR schemes.

Scheme	Accountability	Robustness	Guaranteed abort	Servers	Rounds
EKN22-TCC [9]*	◯	Statistical	◯	$t+d+1$	1
EKN22-ITC [8]	◯	Perfect	×	$t+2d+1$	1
Cao23 [5]	◯	Perfect	×	$t+d+1$	1
EKN24 [10]	×	Perfect	◯	$2t+15$	1
AB25 [1]	×	Statistical	×	$t+d+1$	1
DGH12 [7]	◯	Statistical	×	$t+d+2$	>1
Zhang14 [15]	◯	Statistical	×	$t+1$	>1
Yao19 [14]	◯	Perfect	×	$2d+1$	>1
SynTrace (this work)	◯	Statistical	◯	$t+d+1$	1

* EKN22-TCC also supports perfect retrieval, in which case $k = t + 2d + 1$.

Table 2. Communication for retrieving one 32 KiB record from a 1 GiB database of 32,768 records, with $(k, t, d) = (5, 2, 2)$. Both constructions use one communication round and target failure probability at most 2^{-40}. SynTrace uses $m = 4$ audit queries per server. Section 6.3 gives the cost formulas and parameter selection.

Scheme	q	Upload (MiB)	Download (MiB)	Total (MiB)
EKN22-TCC [9, Thm. 7]	2^{43}	7.56	1.41	8.96
SynTrace	2^{15}	**1.46**	**0.78**	**2.25**

Against a static coalition of at most t corrupted servers, the compiler tolerates up to $d \leq t$ deviations. Under the stated reconstruction and parameter conditions, the client recovers the requested item and identifies exactly the deviating servers within this threshold, except with a chosen statistical error. Within this bound it never identifies a correctly responding server. When more than d servers deviate, SynTrace provides *guaranteed abort* with an empty identification set, except with the stated statistical error.

Applying SynTrace to a Shamir-based PIR yields an accountable PIR with $k = t + d + 1$ servers. Even reconstructing the item from all but d responses requires at least $t + d + 1$ servers under our privacy and download assumptions. The number of servers is therefore optimal under these assumptions, with no additional servers needed for accountability. We compare SynTrace with existing PIR schemes in Table 1.

For a concrete comparison of communication costs, we use the construction of Eriguchi, Kurosawa and Nuida [9], which also provides robustness and accountability in one communication round with the same number of servers and information-theoretic query privacy. Table 2 compares the two schemes for retrieving one 32 KiB record from a 1 GiB database with $k = 5$ and $t = d = 2$, targeting failure probability at most 2^{-40}. Each scheme uses its own field size and packs the same records accordingly. The MAC construction uses 8.96 MiB of total communication, whereas SynTrace uses 2.25 MiB, a reduction of 74.95%.

We further extend SynTrace to batched retrieval with shared audits, amortizing the communication cost of accountability across the requested items. A batch of B retrievals shares a single set of m audit queries per server, requiring $B+m$ query and answer components rather than the $B(m+1)$ components of independent SynTrace executions. The per-item communication factor is therefore $1+m/B$ relative to the same base PIR over the same field. We prove robustness, exact identification and guaranteed abort for the entire batch in one communication round, with the same (k, t, d) and the same upper bound on statistical security error, without a multiplicative factor B. Thus, larger batches reduce the communication overhead per item without requiring more servers or additional audit queries. Section 6.4 presents the batched construction and its guarantees.

2 Preliminaries

We introduce the PIR interface, explain how a compiler builds on it, and recall the linear-code properties used to check audit responses.

2.1 Notation

For an integer $k \geq 1$ write $[k] = \{1, \ldots, k\}$. We write $x \leftarrow \mathcal{X}$ for uniform sampling, $\mathbb{F}_q$ for the field with q elements, and $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$. For $v \in \mathbb{F}_q^k$, $\mathrm{supp}(v) = \{j : v_j \neq 0\}$, and e_ℓ is the ℓ-th standard basis vector. The statistical distance is $\Delta(X, Y) = \frac{1}{2} \sum_\omega |\Pr[X = \omega] - \Pr[Y = \omega]|$. We use $\perp$ for failure. We omit internal randomness from the inputs of randomized algorithms. Probabilities are taken over all randomness in the stated experiment unless specified otherwise.

The PIR parameters are summarized in Table 3. We write k for the number of servers, t for the maximum number of corrupted servers, and $d \leq t$ for the maximum number of deviating servers tolerated for robustness and identification. The query and answer lengths per server are denoted by ℓ_Q and ℓ_A, respectively, measured in field elements.

The *reconstruction threshold* u ensures that reconstruction succeeds from any set of at least u correctly computed responses. All concrete base PIR constructions used in this paper have $u = t + 1$.

In SynTrace, *retrieval queries* are used to retrieve the requested item x_i, while *audit queries* are auxiliary queries used to identify deviating servers.

2.2 Multi-Server PIR

In a standard multi-server PIR setting, a database $x \in \mathbb{F}_q^n$ is replicated across k servers. The client wishes to retrieve x_i without revealing i to a coalition of servers. The client has no database-dependent advice or preprocessing, and its query generation uses only the index, public parameters and private randomness.

Figure 1 provides a graphical illustration of a single-round PIR protocol $\Pi = (\mathsf{Query}, \{\mathsf{Ans}_j\}_{j\in[k]}, \mathsf{Rec})$. The client runs the randomized algorithm $\mathsf{Query}(i)$ to

Table 3. Main notation used in this paper.

Notation	Meaning
k	Number of servers.
t	Maximum number of corrupted servers.
d	Maximum number of deviating servers tolerated for robustness and identification.
u	Reconstruction threshold of the base PIR.
m	Number of audit queries per server for one retrieval.
q, $\mathbb{F}_q$	Field size and the finite field with q elements.
n, $x \in \mathbb{F}_q^n$	Database length in field elements and the database.
ℓ_Q, ℓ_A	Base-PIR query and answer lengths per server, in field elements.
$\perp$	Failure symbol, also used for missing or malformed responses.

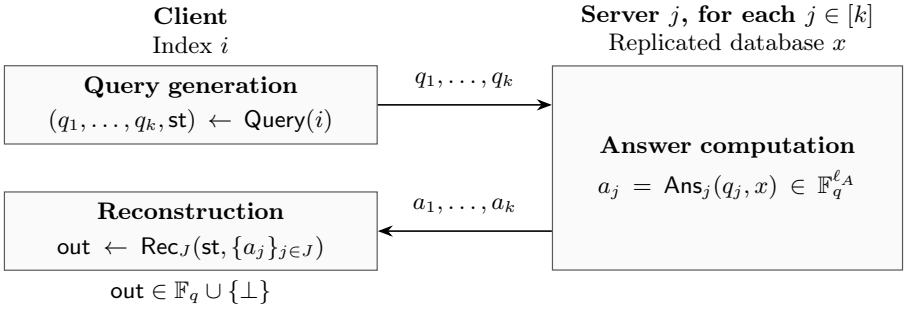


Fig. 1. A single-round multi-server PIR protocol. The client sends a single query to each of the k servers in parallel and reconstructs from the responses of servers in $J \subseteq [k]$.

obtain $(q_1, \ldots, q_k, \mathsf{st})$, sends q_j to server j, and retains st for reconstruction. Upon receiving q_j, server j computes $a_j = \mathsf{Ans}_j(q_j, x) \in \mathbb{F}_q^{\ell_A}$ using a public deterministic algorithm. To recover the requested item, the client applies $\mathsf{Rec}_J(\mathsf{st}, \{a_j\}_{j \in J})$ to the answers from a set J of servers and outputs an element of $\mathbb{F}_q \cup \{\perp\}$. The retained state includes any information needed for reconstruction, such as coefficients determined by the query randomness.

For $t \in [k-1]$, the protocol is *perfectly t-private* if, for every $T \subseteq [k]$ with $|T| \leq t$ and every pair of indices i, i', the query tuples $(q_j)_{j \in T}$ generated for i and i' have identical distributions. The definition in Section 3 additionally considers the client's output under malicious responses.

2.3 PIR Compilers

A *PIR compiler* transforms a base PIR protocol into another PIR protocol with additional guarantees. The base protocol supplies the query-generation, answer and reconstruction algorithms. The compiler uses these algorithms as components of the new protocol and adds the operations needed for its guarantees. For

example, existing constructions transform ordinary PIR into error-correcting PIR [9], or combine linear PIR with commitments to verify retrieved values [5]. Each compiler specifies the properties required of its base protocol.

Figure 2 illustrates this relationship for $\mathsf{SynTrace}$. The superscript base marks the algorithms being reused, and $\mathsf{SynTrace}$ marks the algorithms of the resulting protocol. Both protocols follow the client–server structure in Figure 1. In $\mathsf{SynTrace}$, the client calls $\mathsf{Query}^{\mathsf{base}}$ locally and generates additional queries using an associated audit sampler $\mathsf{Audit}^{\mathsf{base}}$. The client blinds and mixes these queries before sending them. Each server applies $\mathsf{Ans}_j^{\mathsf{base}}$ to the received components, using its copy of the database.

After receiving the responses, the client undoes the mixing and blinding, identifies deviating servers, and verifies the audit answers. It then calls $\mathsf{Rec}_J^{\mathsf{base}}$ on the remaining retrieval answers, using the state retained from base query generation. The resulting protocol returns the retrieved value together with an identification set, or aborts. The required properties of the base PIR and the full construction are given in Section 4.

2.4 Linear Codes

To express the relations among honest audit answers, we use linear codes. A linear $[N, K, D]$ code $\mathcal{C} \subseteq \mathbb{F}_q^N$ is a K-dimensional subspace whose minimum distance D is the smallest weight of a nonzero codeword. A parity-check matrix H describes the code by $\mathcal{C} = \{c : Hc = 0\}$. We write $H_{*,j}$ for its j-th column.

A parity-check matrix allows the client to verify linear relations among responses without knowing their individual correct values. If a correct response vector c belongs to $\mathcal{C}$ and the received vector is $y = c + e$, then its *syndrome* satisfies

$$Hy = H(c + e) = He.$$

The syndrome therefore depends only on the error vector. To identify its support, we use the following relation between code distance and parity-check columns.

Lemma 1 (Code distance). *A linear code with parity-check matrix H has minimum distance at least δ if and only if every $\delta - 1$ columns of H are linearly independent.*

Proof. A nonzero codeword of weight w is a nonzero linear dependence among w columns of H, so the minimum weight is the least w for which some w columns are dependent. □

A code is *maximum distance separable* (MDS) if $D = N - K + 1$. Generalized Reed–Solomon codes, used by the Shamir-based instantiation, have this property.

To handle missing responses, we restrict the code to the responding servers. For $P \subseteq [N]$, the resulting *punctured code* $\mathcal{C}|_P$ consists of codewords restricted to the positions in P. If $e = N - |P| < D$, it has length $N - e$, dimension K and minimum distance at least $D - e$.

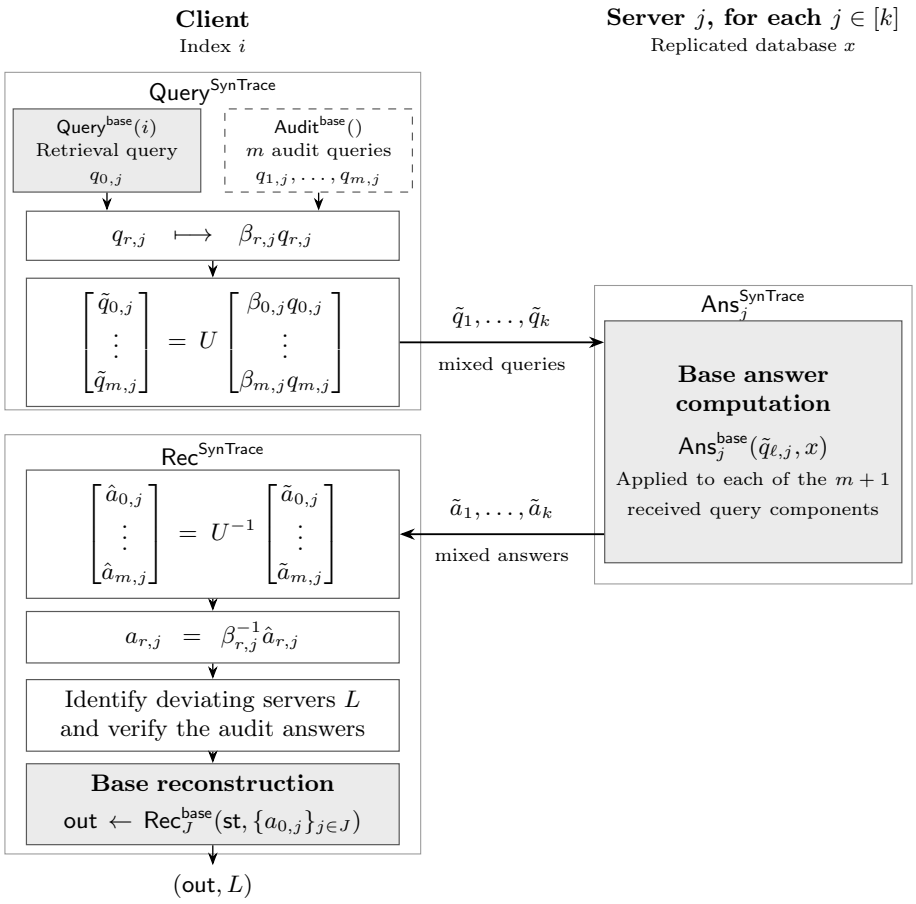


Fig. 2. How SynTrace uses a base PIR. Grey boxes indicate the algorithms reused from the base PIR. Here $\tilde{q}_j = (\tilde{q}_{0,j}, \ldots, \tilde{q}_{m,j})$ and $\tilde{a}_j = (\tilde{a}_{0,j}, \ldots, \tilde{a}_{m,j})$. The client samples an invertible matrix U and nonzero blinding scalars $\beta_{r,j}$. The set J contains the remaining servers used for reconstruction after excluding L.

3 Security Model and Definitions

In this section, we formalize robustness, accountability and guaranteed abort jointly with privacy in the real/ideal paradigm.

3.1 Adversary and Required Outputs

We consider an honest client in the single-round setting of Section 2.2 and a malicious adversary that statically corrupts a set $T \subseteq [k]$ with $|T| \leq t$, where $1 \leq d \leq t < k$. After observing the queries sent to these servers, it chooses their

responses jointly using the database, auxiliary input and its private randomness. It does not observe honest-server queries or the client's private state.

Let $\mathcal{M}$ denote the set of servers that deviate in the current execution. For queries q_j and responses a_j, this set is

$$\mathcal{M} = \{j \in [k] : a_j = \perp \text{ or } a_j \neq \mathsf{Ans}_j(q_j, x)\}.$$

Here $a_j = \perp$ represents a malformed response or failure to respond by the response deadline. Since uncorrupted servers follow the protocol, $\mathcal{M} \subseteq T$. A corrupted server that responds correctly does not belong to $\mathcal{M}$.

The client outputs (out, L), where $\mathsf{out} \in \mathbb{F}_q \cup \{\perp\}$ is the retrieved value or an abort symbol and $L \subseteq [k]$ is its *identification set*. Identification is *exact* when $L = \mathcal{M}$; a *false identification* occurs when $L \not\subseteq \mathcal{M}$. For every corrupted coalition of size at most t, the output must satisfy the following rule, except with a specified statistical error.

$$(\mathsf{out}, L) = \begin{cases} (x_i, \mathcal{M}), & |\mathcal{M}| \leq d, \\ (\perp, \emptyset), & |\mathcal{M}| > d. \end{cases}$$

3.2 Ideal Functionality

Following the real/ideal approach to malicious PIR [1], we describe the desired computation through a trusted party. Our ideal functionality $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}(k, t, d)$ holds the database x, receives the requested index i from the client and the deviation set $\mathcal{M}^\star$ from a simulator, and returns the client's output $(\mathsf{out}^\star, L^\star)$. This output consists of the requested item and the identification set when $|\mathcal{M}^\star| \leq d$, or an abort symbol and the empty set otherwise.

The simulator $\mathcal{S}$ represents the corrupted servers in the ideal execution. Using (T, x, z) without knowing i, it produces their simulated view $\mathsf{view}^\star$ and computes $\mathcal{M}^\star$ from the responses in that view. The ideal execution proceeds as follows.

Ideal execution with $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}(k, t, d)$.
Setup. Fix $x \in \mathbb{F}_q^n$, $T \subseteq [k]$ with $|T| \leq t$, and auxiliary input z. The functionality holds x, and $\mathcal{S}$ receives (T, x, z).
Client input. The client sends $i \in [n]$ to $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}$.
Simulation. Without receiving i, $\mathcal{S}$ generates $\mathsf{view}^\star$, containing the auxiliary input z and the corrupted servers' simulated messages, randomness and state. It computes $\mathcal{M}^\star \subseteq T$ from x and the query-response pairs in $\mathsf{view}^\star$ using the rule in Section 3.1, and sends $\mathcal{M}^\star$ to $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}$.
Client output. The functionality returns

$$(\mathsf{out}^\star, L^\star) = \begin{cases} (x_i, \mathcal{M}^\star), & |\mathcal{M}^\star| \leq d, \\ (\perp, \emptyset), & |\mathcal{M}^\star| > d. \end{cases}$$

Simulator output. The simulator outputs $\mathsf{view}^\star$.

3.3 Real/Ideal Security

To require a real protocol to provide the guarantees of $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}$, we compare the joint distribution of the adversary's view and the client's output in the real and ideal executions. Definition 1 formalizes PIR with accountability by requiring these distributions to be statistically close, using a simulator that does not know the requested index.

Definition 1 (PIR with accountability). *For a protocol Π, let $\mathcal{A}$ be an adversary with auxiliary input z that statically corrupts a set T with $|T| \le t$. For fixed (x, T, z, i), define*

$$\mathsf{Real}_{\mathcal{A}}^{\Pi} = \big(\mathsf{view},\ \mathsf{out},\ L\big), \qquad \mathsf{Ideal}_{\mathcal{S}}^{\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}} = \big(\mathsf{view}^{\star},\ \mathsf{out}^{\star},\ L^{\star}\big),$$

where $\mathsf{view} = (z, \mathsf{view}_{\mathcal{A}})$ contains the coalition's messages, randomness and state, together with z, and (out, L) is the client's retrieved value or abort symbol and identification set. In the ideal execution, $\mathsf{view}^{\star}$ is the simulator's output and $(\mathsf{out}^{\star}, L^{\star})$ is the client's output from $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}$.

We say that Π statistically realizes *$\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}(k, t, d)$ with error ε if for every $\mathcal{A}$ there is such a simulator $\mathcal{S}$ satisfying*

$$\Delta\big(\mathsf{Real}_{\mathcal{A}}^{\Pi}, \mathsf{Ideal}_{\mathcal{S}}^{\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}}\big) \le \varepsilon \qquad \text{for every } (x, T, z, i).$$

The following proposition shows that Definition 1 implies robustness, accountability, guaranteed abort and privacy.

Proposition 1 (Consequences of the security definition). *If Π realizes $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}(k, t, d)$ with error ε, then for every adversary and every (x, T, z, i) the following bounds hold.*

(i) (Accountability) $\Pr[L \neq \mathcal{M} \wedge |\mathcal{M}| \le d] \le \varepsilon$.

(ii) (Robustness) $\Pr[\mathsf{out} \neq x_i \wedge |\mathcal{M}| \le d] \le \varepsilon$.

(iii) (Guaranteed abort) $\Pr[(\mathsf{out}, L) \neq (\perp, \emptyset) \wedge |\mathcal{M}| > d] \le \varepsilon$.

(iv) (No false identification) $\Pr[L \not\subseteq \mathcal{M}] \le \varepsilon$.

(v) (Correctness of accepted outputs) $\Pr[\mathsf{out} \notin \{x_i, \perp\}] \le \varepsilon$.

(vi) (Privacy) *For every $i, i' \in [n]$ with the same (x, T, z) and auxiliary input z chosen independently of the requested index,*

$$\Delta\big((\mathsf{view}, \mathsf{abort}, L, \mathcal{M})_i, (\mathsf{view}, \mathsf{abort}, L, \mathcal{M})_{i'}\big) \le 2\varepsilon.$$

In (vi), abort *records whether the client aborts.*

These bounds follow directly from the statistical-distance guarantee in Definition 1. The detailed proof is given in Appendix A.1.

4 The SynTrace Compiler

The SynTrace compiler takes a base PIR Π^{base} and an audit-query algorithm Audit and produces an accountable PIR protocol Π_{SynTrace} for the security model of Section 3. Throughout this section, Query, Ans_j and Rec_J denote the algorithms of the base PIR, which the resulting protocol uses as internal components.

Figure 2 illustrates how these components are used inside SynTrace. We now specify the audit queries and the operations that combine them with the base PIR.

4.1 Construction Overview

Audit queries allow the client to identify deviating servers without knowing the database. They must remain indistinguishable from retrieval queries to the corrupted coalition. For example, Shamir-based PIR generates retrieval queries by sharing a selection vector e_i and audit queries by sharing the zero vector. Any t servers see uniform shares in both cases. Correctly computed audit answers evaluate a polynomial with constant term zero, so they belong to a public linear code even though the client does not know their values.

Simply sending the retrieval query alongside m audit queries is insufficient. Even if their order is randomly permuted, a corrupted server can guess which query is the retrieval query, alter only its answer, and answer the other m queries correctly. If the guess is correct, all audit answers remain valid, while the altered retrieval answer can cause an incorrect result, violating robustness.

To make such selective manipulation unlikely, the client first blinds each retrieval and audit query $q_{r,j}$ by multiplying it by an independently sampled nonzero scalar $\beta_{r,j} \in \mathbb{F}_q^*$. It then mixes the $m+1$ blinded queries for each server using a secret invertible matrix $U \in \mathbb{F}_q^{(m+1)\times(m+1)}$, as illustrated in Figure 2. Linearity lets the client undo both operations on the responses. The matrix hides which combinations represent audit queries; the blinding randomizes residual errors across servers, including errors chosen jointly by a coalition. Neither operation changes a correctly computed response after it is undone.

The client next computes the audit syndromes. Because correctly computed audit answers have zero syndrome, the observed syndrome span is contained in the span of the parity-check columns of deviating servers. Within the deviation threshold, sufficient residual rank makes this space contain exactly the columns corresponding to those servers. The code distance prevents a correctly responding server from being identified within the deviation threshold. The client verifies the resulting space, excludes the identified responses, and reconstructs from the remaining ones. Responses that are missing or malformed are removed before this calculation by puncturing the audit code.

4.2 Requirements on the Base PIR

Definition 2 collects the usual reconstruction and query-privacy requirements of PIR, together with linearity of the server's answer computation. These conditions

are already satisfied by existing linear PIR constructions, including Shamir-based PIR. Linearity ensures that a server's answer to a mixed query is the same linear combination of the original answers, allowing the client to undo mixing and blinding. We then specify the additional requirements on retrieval and audit queries needed by $\mathsf{SynTrace}$.

Definition 2 (Linear PIR). *Let Π^{base} be a k-server PIR over $\mathbb{F}_q$ with reconstruction threshold u and answer length $\ell_A \geq 1$, and fix $1 \leq t < k$. Write $\Pi^{\mathsf{base}} = (\mathsf{Query}, \{\mathsf{Ans}_j\}_{j\in[k]}, \mathsf{Rec})$. Let $\mathcal{Q}$ be the common query space of all servers. We require $\mathcal{Q}$ to be closed under linear combinations over $\mathbb{F}_q$ so that it also contains the mixed queries. Using the notation of Figure 1, a base-PIR execution on database x and requested index i is given by*

$$\begin{aligned}(q_1, \ldots, q_k, \mathsf{st}) &\leftarrow \mathsf{Query}(i),\\ a_j &= \mathsf{Ans}_j(q_j, x), \qquad j \in [k],\\ \mathsf{out} &= \mathsf{Rec}_J(\mathsf{st}, \{a_j\}_{j\in J}).\end{aligned}$$

We call Π^{base} a perfectly t-private linear PIR *with reconstruction threshold u if it satisfies the following conditions.*

Reconstruction. *For every $x \in \mathbb{F}_q^n$, $i \in [n]$ and $J \subseteq [k]$ with $|J| \geq u$,*

$$\Pr[\mathsf{out} = x_i] = 1.$$

Query privacy. *For every $i, i' \in [n]$, $T \subseteq [k]$ with $|T| \leq t$ and $v \in \mathcal{Q}^{|T|}$,*

$$\Pr[(q_j)_{j\in T} = v] = \Pr[(q'_j)_{j\in T} = v],$$

where $(q'_1, \ldots, q'_k, \mathsf{st}') \leftarrow \mathsf{Query}(i')$.

Linearity. *For every server j, database x, queries $q, q' \in \mathcal{Q}$ and scalars $\alpha, \beta \in \mathbb{F}_q$,*

$$\mathsf{Ans}_j(\alpha q + \beta q', x) = \alpha \mathsf{Ans}_j(q, x) + \beta \mathsf{Ans}_j(q', x).$$

To use this base PIR in $\mathsf{SynTrace}$, we require both retrieval queries and audit queries to be jointly uniform across any coalition of at most t servers. This is stronger than query privacy, which requires only index-independence. With independent sampling of the retrieval and audit queries, this uniformity preserves their distribution under nonzero blinding and invertible mixing, hiding β and U from the coalition's queries.

The client must also identify deviations without knowing the database or the correct audit answers. For each answer coordinate, the vector a of correct audit answers satisfies $Ha = 0$ for a public matrix H. If the received vector is $a + e$, then $H(a + e) = He$, removing the unknown correct answers and retaining only information about the errors.

Each column of H corresponds to a server. Independence of every $d + 1$ columns ensures that the span of the columns of at most d deviating servers contains no column of a correctly responding server, preventing false identification. Identifying every deviating server additionally requires the vectors He

from the audits to span all of their columns. The security analysis shows how blinding and mixing make this happen with high probability.

Definition 3 combines these requirements on query distributions and audit answers into the notion of a *d-audit-compatible linear PIR*, specifying which base PIR schemes can be used in SynTrace.

Definition 3 (Audit-Compatible Linear PIR). *Let Π^{base} be a linear PIR with perfect t-privacy, and fix $1 \le d \le t$. We say that Π^{base} is* d-audit-compatible *if there exists a randomized audit-query algorithm* Audit *with the following properties. The retrieval queries are generated by $(q_{0,1}, \ldots, q_{0,k}, \mathsf{st}) \leftarrow \mathsf{Query}(i)$. An audit call produces $(q_{1,1}, \ldots, q_{1,k}) \leftarrow \mathsf{Audit}$ in $\mathcal{Q}^k$.*

Query uniformity. *For every $i \in [n]$, $T \subseteq [k]$ with $|T| \le t$ and $v \in \mathcal{Q}^{|T|}$,*

$$\Pr[(q_{0,j})_{j\in T} = v] = \Pr[(q_{1,j})_{j\in T} = v] = |\mathcal{Q}|^{-|T|},$$

where the probabilities are over the randomness of Query *and* Audit*, respectively.*

Audit answer relations. *There is a public matrix $H \in \mathbb{F}_q^{h\times k}$ of full row rank with the following property. For every database x, the correct responses $a_j = \mathsf{Ans}_j(q_{1,j}, x)$ to queries generated by* Audit *satisfy*

$$H \begin{bmatrix} a_1^\top \\ \vdots \\ a_k^\top \end{bmatrix} = 0.$$

We define the audit answer code *by $\mathcal{C} := \ker H = \{a \in \mathbb{F}_q^k : Ha = 0\}$.*

Column independence. *Every matrix formed by selecting any $d+1$ columns of H has full column rank $d+1$.*

4.3 Shamir-Based Instantiation

For our concrete construction, we choose Shamir-based PIR as the base protocol Π^{base}. Its polynomial representation gives an explicit construction of the retrieval and audit queries and makes their parameters easy to determine. Shamir sharing is based on Reed–Solomon codes, and the construction extends to PIR obtained from more general MDS-code-based threshold sharing. Appendix C presents this extension and an example beyond the generalized Reed–Solomon class.

In Shamir-based PIR, retrieval queries share the selection vector e_i, so the correct responses are evaluations of a polynomial of degree at most t whose constant term is the unknown item x_i. To generate audit queries, the client instead shares the zero vector. The correct audit responses are then evaluations of a polynomial g of degree at most t with the known constant term $g(0) = 0$. This gives the client an additional linear relation among the responses, even though their individual values depend on the database.

We denote this base PIR by Π_{Shamir} and define it together with its audit-query algorithm in Figure 3.

Protocol Π_{Shamir}

Public parameters. Distinct nonzero points $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$, where $t < k \leq q-1$. Each server stores $x \in \mathbb{F}_q^n$; the client requests index $i \in [n]$.

$\mathsf{Query}(i)$. Sample $a_1, \ldots, a_t$ independently and uniformly from $\mathbb{F}_q^n$ and generate the retrieval queries

$$q_j = e_i + \sum_{s=1}^{t} a_s \alpha_j^s, \qquad j \in [k].$$

Return $(q_1, \ldots, q_k, \mathsf{st})$, where $\mathsf{st} = (\alpha_1, \ldots, \alpha_k)$.

$\mathsf{Audit}()$. Independently sample uniform vectors $b_1, \ldots, b_t \in \mathbb{F}_q^n$ and generate the audit queries

$$q_j = \sum_{s=1}^{t} b_s \alpha_j^s, \qquad j \in [k].$$

$\mathsf{Ans}_j(q_j, x)$. Return $\langle q_j, x \rangle$. For either query type, correct answers are evaluations of a polynomial g of degree at most t, with

$$g(0) = \begin{cases} x_i & \text{for retrieval queries,} \\ 0 & \text{for audit queries.} \end{cases}$$

$\mathsf{Rec}_J(\mathsf{st}, \{a_j\}_{j \in J})$. For $|J| \geq u = t+1$, interpolate g from the correctly computed retrieval answers of servers in J and output $g(0) = x_i$.

Fig. 3. Shamir-based PIR and its associated audit queries. Retrieval queries share e_i, whereas audit queries share the zero vector.

The following proposition records standard properties of Shamir-based PIR. For completeness, we include a proof in Appendix A.2.

Proposition 2 (Linear PIR Properties of Π_{Shamir}). *The protocol Π_{Shamir} is a perfectly t-private linear PIR with reconstruction threshold $u = t+1$.*

It remains to verify the additional requirements for audit compatibility. The following proposition shows that the audit queries in Figure 3 satisfy these requirements when $k \geq t + d + 1$.

Proposition 3 (Audit Compatibility of Π_{Shamir}). *The protocol Π_{Shamir} is d-audit-compatible if $k \geq t + d + 1$.*

The intuition is that Shamir shares look jointly uniform to any t servers, whether they encode a retrieval query or zero for an audit. Sharing zero also fixes the audit answers' constant term, giving the client known linear relations among them. The resulting code has distance $k - t + 1$, which is at least $d + 2$ when $k \geq t + d + 1$, ensuring the required column independence. The detailed proof is given in Appendix A.3.

The same construction also supports a matrix layout of the database, which we use for the communication analysis. Arrange the database as $X \in \mathbb{F}_q^{\ell_Q \times \ell_A}$

with $n = \ell_Q \ell_A$. To retrieve an entry, the client shares the selection vector $e_{\mathrm{row}} \in \mathbb{F}_q^{\ell_Q}$ for its row using the query algorithm of Figure 3. Each server receives a share $q_j \in \mathbb{F}_q^{\ell_Q}$ and returns the vector

$$a_j = q_j^\top X \in \mathbb{F}_q^{\ell_A}.$$

The client reconstructs the selected row from any $t+1$ correct responses and outputs the desired entry, retaining its column index in the client state.

For this layout, audit queries share the zero vector in $\mathbb{F}_q^{\ell_Q}$. Query uniformity follows from the same Shamir sharing argument, and the answer map remains linear. For each answer coordinate, the correct audit responses are evaluations of a degree-at-most-t polynomial with constant term zero. They therefore belong to the same audit code $\mathcal{C}$ and are checked by the same matrix H. Thus the matrix layout remains d-audit-compatible under the conditions of Proposition 3, with reconstruction threshold $u = t+1$. Section 6.1 chooses the matrix dimensions to balance the query and answer lengths.

4.4 The SynTrace Protocol

We now construct Π_{SynTrace} from a base PIR Π^{base} satisfying Definition 3. The client uses Query to generate the retrieval queries and Audit to generate m additional audit queries per server. It blinds and mixes these queries before sending them. Each server evaluates the base answer algorithm Ans_j on the received components. After unmixing the responses, the client uses the audit answers to identify deviating servers and verify the result, then applies Rec_J to the remaining retrieval answers. Figure 4 defines the three algorithms of Π_{SynTrace}, marked by the superscript $\mathsf{SynTrace}$. The superscript base marks calls to the base PIR's algorithms and its audit-query sampler. The client sends the queries returned by $\mathsf{Query}^{\mathsf{SynTrace}}$ and retains $\widetilde{\mathsf{st}}$ for reconstruction.

We assume $m \geq 1$ and $q \geq 3$, and require $k - d \geq u$ so that reconstruction remains possible after excluding up to d servers. The audit code $\mathcal{C}$ and its public parity-check matrix H are those of Definition 3. In the figure, $\mathrm{GL}_{m+1}(\mathbb{F}_q)$ denotes the set of invertible $(m+1) \times (m+1)$ matrices over $\mathbb{F}_q$.

The deviation set in Section 3.1 is evaluated on the complete messages $\tilde{q}_j$ and $\tilde{a}_j$, with correct responses computed componentwise by $\mathsf{Ans}_j^{\mathsf{SynTrace}}$.

Independent blinding prevents correlated errors from remaining indistinguishable after unmixing. For example, without blinding, identical errors at two servers remain identical after applying the shared matrix U^{-1}. Their audit errors are then linearly dependent, so the syndrome information need not identify the servers individually. The independent scalars $\beta_{r,j}$ randomize the errors across servers and audits, making such dependencies unlikely.

Missing or malformed responses identify their servers as deviating. If their number e exceeds d, the client aborts. Otherwise it excludes those servers and lets P denote the remaining servers. Since H checks answers from all k servers, the client replaces it with a matrix H_P expressing the audit relations that involve only servers in P. Formally, H_P is a parity-check matrix of the audit code

Protocol Π_{SynTrace}

Public: The base PIR Π^{base}, the audit code $\mathcal{C}$, and a parity-check matrix H.
Client input: The index $i \in [n]$ of the requested item.
Servers store: A replicated database $x \in \mathbb{F}_q^n$.

$\mathsf{Query}^{\mathsf{SynTrace}}(i)$.

1. Sample $U \leftarrow \mathrm{GL}_{m+1}(\mathbb{F}_q)$ and $\beta_{r,j} \leftarrow \mathbb{F}_q^*$ for $r \in \{0\} \cup [m]$, $j \in [k]$.
2. Generate one retrieval query tuple and m independent audit query tuples:

$$(q_{0,1}, \ldots, q_{0,k}, \mathsf{st}) \leftarrow \mathsf{Query}^{\mathsf{base}}(i),$$
$$(q_{r,1}, \ldots, q_{r,k}) \leftarrow \mathsf{Audit}^{\mathsf{base}}(), \qquad r \in [m].$$

3. For each server j, multiply each query $q_{r,j}$ by $\beta_{r,j}$, then mix the blinded queries using U:

$$\tilde{q}_j = \begin{bmatrix} \tilde{q}_{0,j} \\ \vdots \\ \tilde{q}_{m,j} \end{bmatrix} = U \begin{bmatrix} \beta_{0,j} q_{0,j} \\ \vdots \\ \beta_{m,j} q_{m,j} \end{bmatrix}.$$

Return $(\tilde{q}_1, \ldots, \tilde{q}_k, \widetilde{\mathsf{st}})$, with $\widetilde{\mathsf{st}} = (\mathsf{st}, U, (\beta_{r,j})_{r,j})$.

$\mathsf{Ans}_j^{\mathsf{SynTrace}}(\tilde{q}_j, x)$.

1. Compute $\tilde{a}_{r,j} = \mathsf{Ans}_j^{\mathsf{base}}(\tilde{q}_{r,j}, x)$ for $r \in \{0\} \cup [m]$ and return

$$\tilde{a}_j = (\tilde{a}_{0,j}, \ldots, \tilde{a}_{m,j}) \in \mathbb{F}_q^{(m+1)\times \ell_A}.$$

$\mathsf{Rec}^{\mathsf{SynTrace}}(\widetilde{\mathsf{st}}, \{\tilde{a}_j\}_{j\in[k]})$.

1. Parse missing or malformed responses as $\perp$ and set $L_\perp = \{j : \tilde{a}_j = \perp\}$. If $|L_\perp| > d$, return $(\perp, \emptyset)$. Otherwise let $P = [k] \setminus L_\perp$ be the set of servers with well-formed responses and compute

$$\hat{a}_j = (\hat{a}_{0,j}, \ldots, \hat{a}_{m,j}) = U^{-1}\tilde{a}_j,$$
$$a_{r,j} = \beta_{r,j}^{-1}\hat{a}_{r,j}, \qquad r \in \{0\} \cup [m], \quad j \in P.$$

2. Use a parity-check matrix H_P of $\mathcal{C}|_P$, the audit code restricted to the responding servers in P. For each audit r, stack the responses $a_{r,j}^\top$ as rows in server order and compute the syndrome matrix

$$S_r = H_P(a_{r,j}^\top)_{j\in P}, \qquad r \in [m].$$

3. Let V be the linear span of all columns of $S_1, \ldots, S_m$. Start with $L = L_\perp$. For each $j \in P$, add server j to L if its column in H_P belongs to V.
4. If $|L| > d$, return $(\perp, \emptyset)$.
5. Set $J = P \setminus L$. If $\dim V \neq |L \cap P|$ or $|J| < u$, return $(\perp, \emptyset)$.
6. Set $\mathsf{out} = \mathsf{Rec}_J^{\mathsf{base}}(\mathsf{st}, \{a_{0,j}\}_{j\in J})$. If $\mathsf{out} = \perp$, return $(\perp, \emptyset)$; otherwise return (out, L).

Fig. 4. Construction of Π_{SynTrace} from a base PIR Π^{base} and its audit-query sampler.

restricted to P. The erasure lemma in Appendix A.8 shows that the remaining code supports up to $d - e$ further deviations.

Before reconstruction, the client checks both the number of identified servers and whether their columns explain the entire syndrome space V. It first aborts if $|L| > d$. Otherwise, the columns corresponding to the identified responding servers $L \cap P$ are independent and belong to V. The audit verification check requires

$$\dim V = |L \cap P|,$$

which ensures that these columns span all of V. If this equality fails, the audit syndromes contain error information not explained by the identified servers, and the client aborts.

This verification is needed for guaranteed abort when answers are scalar and $m = d$. When all servers respond, $\dim V \leq d$, so column independence implies $|L| \leq d$ even if more than d servers deviate. The check $|L| > d$ therefore cannot detect that the deviation threshold has been exceeded (Proposition 5). The audit verification check can still reject because the identified servers' columns may not span V. Theorem 4 bounds the probability that an execution above the threshold passes both checks.

4.5 The Number of Servers and Audit Queries

We now explain why the Shamir-based construction uses $k = t + d + 1$ servers and why the identification rule requires at least d audit queries per server. The server bound follows from privacy and robustness under a download restriction, while the audit bound concerns the identification method used by SynTrace.

Consider a single-round PIR with perfectly private queries and no database-dependent client preprocessing. If d servers do not respond, only $k - d$ answers remain for reconstruction. If $k - d \leq t$, perfect query privacy prevents the remaining servers from learning the requested index. In particular, any queries they receive could have been generated for any index. For fixed queries, their answers must therefore contain enough information to recover every database entry, using the corresponding client state for each index. Reconstruction without error would then require these answers to contain at least as much information as the entire database.

If the total download from any t servers is smaller than the database, this is impossible. Hence $k - d > t$, or equivalently $k \geq t + d + 1$. The same bound holds for sufficiently large databases when download is sublinear in the database size and the reconstruction error is sufficiently small. Theorem 8 in Appendix E.1 formally establishes this lower bound under an explicit download condition. The argument uses only robustness against missing responses and does not depend on linearity or an identification rule.

The balanced SynTrace instantiation in Section 6.1 has an answer length that is sublinear in the database size. For fixed t, m and small error, it therefore satisfies the download condition for sufficiently large databases. Under these

conditions, SynTrace with $k = t + d + 1$ adds accountability without requiring servers beyond those already needed for robustness.

The zero-secret audit queries make identification possible at this number of servers. Their correct responses are evaluations of a degree-at-most-t polynomial whose constant term is known to be zero. This known value gives the client one more linear relation among the responses than ordinary retrieval queries, whose constant term is the unknown item. Proposition 3 shows that these audits satisfy the identification requirements with $k = t + d + 1$.

Turning to the audit queries, exact identification in SynTrace of up to d deviating servers requires $m \geq d$ audit queries per server. Indeed, identifying d servers requires the syndrome span V to contain their d linearly independent parity-check columns, so $\dim V \geq d$. For scalar answers, each audit contributes one syndrome vector and hence $\dim V \leq m$. For vector answers, an adversary can change only one answer coordinate at every deviating server. Unmixing and unblinding preserve this restriction, so each audit still contributes at most one syndrome direction. Thus $m < d$ makes exact identification impossible for this adversary, regardless of the answer length.

For the Shamir-based construction with $k = t + d + 1$, the audit code has $h = d + 1$ independent parity checks. Theorem 5 then guarantees that $m = d$ audits suffice for any target statistical error when the field is sufficiently large. Thus the minimum number of servers and the minimum number of audit queries are attained together. The lower bound on the number of audit queries concerns the specified mixing, blinding and identification method, rather than arbitrary PIR protocols.

Additional audit queries can reduce the field size needed for the same statistical error, so minimizing their number need not minimize communication. Section 6.2 analyzes this tradeoff.

5 Security Proof of SynTrace

In this section, we establish privacy, accountability, robustness and guaranteed abort for SynTrace. We first show that the received queries hide both the requested index and the randomness used for blinding and mixing. We then use this remaining randomness to analyze the client's identification set and retrieved value, and combine the results under the real/ideal definition.

Throughout, Π^{base} is a k-server, perfectly t-private, d-audit-compatible linear PIR with reconstruction threshold u and answer length ℓ_A. We use the parameters of Figure 4, including $m \geq 1$, $q \geq 3$ and $k - d \geq u$. Write $h = \operatorname{rank}(H) = k - \dim \mathcal{C}$ for the number of independent audit relations and $w = |\mathcal{M}|$ for the number of deviating servers.

5.1 Privacy

For a corrupted coalition $T \subseteq [k]$ with $|T| \leq t$, write

$$\tilde{q}_T = (\tilde{q}_j)_{j \in T}$$

for the queries it receives in Figure 4. The following proposition shows that observing these messages reveals neither i nor (U, β).

Proposition 4 (Privacy of mixing and blinding). *For every fixed i, the received queries $\tilde{q}_T$ are uniform on $\mathcal{Q}^{(m+1)|T|}$. Moreover, $\tilde{q}_T$ is independent of (U, β).*

Proof. By the query-uniformity condition of audit-compatible linear PIR, the queries $(q_{r,j})_{j \in T}$ are jointly uniform on $\mathcal{Q}^{|T|}$ for each $r \in \{0\} \cup [m]$. Multiplying each query by a fixed nonzero scalar preserves this distribution. The retrieval and audit queries are sampled independently, so the blinded queries $\beta_{r,j} q_{r,j}$, taken over all $r \in \{0\} \cup [m]$ and $j \in T$, remain jointly uniform even after fixing i and all blinding scalars.

For every fixed invertible U, mixing these queries at each server is a bijection and therefore also preserves their joint uniformity. Hence, for every $v \in \mathcal{Q}^{(m+1)|T|}$ and every fixed choice of U, β and i,

$$\Pr[\tilde{q}_T = v \mid U, \beta, i] = |\mathcal{Q}|^{-(m+1)|T|}.$$

The resulting distribution is the same for every i, U and β. Thus conditioning on the received queries leaves the distribution of (U, β) unchanged. □

Theorem 1 (Perfect query privacy). *For any $i, i' \in [n]$ and coalition T with $|T| \leq t$, the received queries $\tilde{q}_T$ have identical distributions under i and i'. An abort decision made by the coalition from these queries and its internal randomness also has a distribution independent of the requested index.*

Proof. Proposition 4 gives the same uniform query distribution for every index. Applying the same decision rule to these queries and the coalition's independent randomness preserves equality of distributions. □

5.2 Accountability and Robustness

Within the deviation threshold, column independence prevents false identification. Blinding and mixing are needed for the separate task of finding every deviating server. Once all such servers are excluded, the base PIR reconstructs the requested item from the remaining responses.

The client immediately identifies and excludes servers whose responses are missing or malformed. If their number e exceeds d, the client aborts. Otherwise, the remaining instance has $k - e$ servers, deviation threshold $d - e$, and $w - e$ deviating servers. Column independence implies that each nonzero audit codeword has at least $d + 2$ nonzero coordinates. Deleting e coordinates leaves at least $d - e + 2$, so any $d - e + 1$ columns of the remaining parity-check matrix are independent. Restriction preserves the audit-code dimension because no nonzero codeword can be supported on the excluded positions, leaving $h - e$ independent audit relations. The number of correct responses needed for reconstruction is unchanged.

In the proofs of accountability, robustness and guaranteed abort, the notation refers to the remaining instance. The theorem statements retain the original parameters and outputs. On successful completion, the client includes the servers identified during preprocessing in its final identification set. On abort, it returns the empty set. This lets us analyze the remaining responses without separate notation for the excluded servers.

Theorem 2 (No false identification below the threshold). *If $|\mathcal{M}| \leq d$, then $L \subseteq \mathcal{M}$ with probability* 1*.*

Proof. The client identifies a server when its column of H lies in V, the span of all columns of the audit syndrome matrices $S_1, \ldots, S_m$. By linearity and the audit-answer relations, these syndromes depend only on errors at deviating servers. Thus V is contained in the span of their columns of H.

There are at most d such columns. Independence of any $d+1$ columns prevents a correctly responding server's column from belonging to their span. Preprocessing also identifies only deviating servers, and an abort returns the empty set. Hence $L \subseteq \mathcal{M}$ for the original execution. □

Theorem 3 bounds the probability of missing any deviating server and the probability of failing to both identify all deviating servers and recover the requested item when at most d servers deviate.

Theorem 3 (Accountability and robustness below the threshold). *Condition on $|\mathcal{M}| = w \leq d$ and assume $m \geq w$. Then*

$$\Pr[L \neq \mathcal{M}] \;\leq\; \min\Big\{1,\; \frac{q^w - 1}{q-1} \cdot 2^m q^{-m}\Big\} \;<\; \frac{q}{q-1}\, 2^m q^{\,w-1-m}.$$

The same upper bound applies to $\Pr[(\mathsf{out}, L) \neq (x_i, \mathcal{M})]$.

We first explain the intuition behind the bound in Theorem 3. Identification requires the deviating servers' audit-error vectors to be linearly independent. For example, if one server's error vector is a scalar multiple of another's, their errors contribute only one independent direction to the syndromes. In general, failure occurs when a nonzero linear combination of these vectors vanishes across all audits.

There are $q^w - 1$ nonzero coefficient vectors for such a relation. Multiplying all coefficients by the same nonzero scalar gives the same relation, so there are $(q^w - 1)/(q - 1)$ candidates. Servers identified through missing or malformed responses leave fewer candidates to consider.

For a fixed relation, choose a server whose coefficient is nonzero. Applying U^{-1} to its response error cannot make the entire error zero, but the part corresponding to a particular audit can be zero. In that audit, multiplying by the inverse blinding scalar still leaves zero, so this server's blinding gives no restriction on whether the relation holds. We bound the probability of the relation by 1 in this case. If its audit error is nonzero, fixing the other scalars leaves at

most one value of its inverse blinding scalar that makes the relation hold. The probability is then at most $1/(q-1)$.

Over the random choice of U, the chosen server's error in any particular audit is zero with probability at most $1/q$. Combining the two cases bounds the probability that the relation holds in one audit by

$$\frac{1}{q} \cdot 1 + \left(1 - \frac{1}{q}\right) \frac{1}{q-1} = \frac{2}{q}. \tag{1}$$

The audits share U, so their errors need not be independent. The analysis instead uses the fact that the chosen server's errors in any specified s audits are all zero with probability at most q^{-s}. Together with independent blinding across audits, this gives $(2/q)^m$ for the relation to hold in all m audits. A union bound over the candidate relations gives the product in the theorem, capped at 1.

The detailed proof is given in Appendix A.4.

Theorem 3 shows that, for fixed w and q, each additional audit contributes a factor of $2/q$ to the bounds.

5.3 Guaranteed Abort

When $w > d$, the required output is $(\perp, \emptyset)$. Figure 4 checks both the size of L and whether the columns belonging to identified responding servers span V. The next theorem bounds the chance that an execution above the threshold passes these checks.

The proof of Theorem 3 shows that, for any fixed nonzero coefficient vector, the corresponding linear dependence among the servers' audit-error vectors holds with probability at most $2^m q^{-m}$. This calculation also applies when $w > d$. We write $F = 2^m q^{-m}$ and use this bound to analyze failure to abort.

Theorem 4 (Guaranteed abort). *Condition on $|\mathcal{M}| = w > d$. The following bounds hold.*

(i) *The client outputs $(\perp, \emptyset)$ at or before the threshold check in reconstruction step 4, except with probability at most*

$$\min\Big\{1, \frac{q^{\,h}-1}{q-1} F\Big\} \le \frac{q}{q-1} 2^m q^{\,h-1-m},$$

(ii) *Accounting also for the verification check, the client outputs $(\perp, \emptyset)$ except with probability at most*

$$\varepsilon_{\text{abort}} := \min\Big\{1,\ \Big(k \frac{q^{\,h-1}-1}{q-1} + 1\Big) F\Big\}.$$

The detailed proof is given in Appendix A.5.

Theorem 4 shows that the bound in part (i) tends to zero as q grows when $m \ge h$, whereas part (ii) already does so when $m \ge h-1$. For the Shamir-based construction with $k = t + d + 1$, we have $h = d + 1$. Thus, $m = d$ audits suffice to make the guaranteed-abort failure probability negligible in the security parameter by choosing a sufficiently large field.

5.4 Overall Security Guarantee

The preceding results describe what the coalition sees and what the client returns. To obtain the real/ideal guarantee, we must compare their joint distribution. The following lemma shows how perfect query privacy and the output bounds provide this comparison for protocols with deterministic server answers.

Lemma 2 (From output bounds to accountability). *Let Π be a single-round, perfectly t-private protocol with deterministic server answers. Suppose that, for every adversary corrupting at most t servers and every (x, T, z, i),*

$$\Pr[(\mathsf{out}, L) \neq (x_i, \mathcal{M}),\ |\mathcal{M}| \leq d] \leq \varepsilon_1,$$
$$\Pr[(\mathsf{out}, L) \neq (\perp, \emptyset),\ |\mathcal{M}| > d] \leq \varepsilon_2.$$

Then Π satisfies Definition 1 with error at most $\varepsilon_1 + \varepsilon_2$. If the corresponding failure bound holds conditionally on each deviation count that occurs with positive probability, the error is at most $\max\{\varepsilon_1, \varepsilon_2\}$. Conversely, security with error ε implies both displayed bounds with $\varepsilon_1 = \varepsilon_2 = \varepsilon$.

The detailed proof is given in Appendix A.6.

For SynTrace, the preceding output bounds hold conditionally on each deviation count. We can therefore use the larger bound as the overall security error, without adding the errors from the two threshold cases.

Theorem 5 (Security of SynTrace). *Assume $m \geq d$. Against any static corrupted coalition of size at most t, SynTrace has perfectly private queries and satisfies Definition 1 with error at most*

$$\varepsilon_{\mathrm{abort}} = \min\left\{1, \left(k\frac{q^{h-1}-1}{q-1} + 1\right) 2^m q^{-m}\right\}.$$

For each deviation count $w \leq d$ occurring with positive probability, its identification set always satisfies $L \subseteq \mathcal{M}$, and

$$\Pr[(\mathsf{out}, L) \neq (x_i, \mathcal{M}) \mid |\mathcal{M}| = w] \leq \min\left\{1, \frac{q^w - 1}{q-1} 2^m q^{-m}\right\}.$$

For each $w > d$ occurring with positive probability, the probability of an output other than $(\perp, \emptyset)$ is at most $\varepsilon_{\mathrm{abort}}$, conditionally on that value of w.

Proof. Perfect query privacy follows from Theorem 1. For $w \leq d$, Theorem 3 gives the joint accountability and robustness bound, while Theorem 2 gives $L \subseteq \mathcal{M}$ with probability 1. For $w > d$, Theorem 4 bounds failure of guaranteed abort by $\varepsilon_{\mathrm{abort}}$.

Column independence requires $h \geq d + 1$, so the bound for $w \leq d$ is also at most $\varepsilon_{\mathrm{abort}}$. Lemma 2 now gives the claimed real/ideal error. □

The error bound is $O(k2^m q^{h-2-m})$. Thus $m \geq h - 1$ allows any target statistical error by increasing the field size; Lemma 3 gives an explicit choice. For the Shamir-based construction at $k = t + d + 1$, we have $h = d + 1$, so $m = d$ suffices. When at most d deviations can occur, including the case $d = t$, only the smaller bound for $w \leq d$ is needed.

6 Instantiation and Parameters

Having established security and the server and audit bounds, we now choose parameters for the Shamir-based construction and evaluate its communication and computation costs.

6.1 Balanced Shamir

We use the matrix layout introduced in Section 4.3. For a database $X \in \mathbb{F}_q^{\ell_Q \times \ell_A}$ with $n = \ell_Q \ell_A$, each base query contains ℓ_Q field elements and each base answer contains ℓ_A field elements. One SynTrace retrieval therefore sends $k(m+1)\ell_Q$ field elements and receives $k(m+1)\ell_A$. For a fixed database size, their sum is minimized by choosing the row and column counts as close as possible. Taking $\ell_Q, \ell_A = \Theta(\sqrt{n})$, with padding when necessary, gives $O(k(m+1)\sqrt{n})$ total communication in field elements.

6.2 Analytical Cost Estimates

For a single retrieval, each server receives one message and sends one response containing $m+1$ base-PIR components. For base query and answer lengths ℓ_Q, ℓ_A, the total communication is $k(m+1)(\ell_Q + \ell_A)$ field elements. To express this cost in bits, we choose a field large enough to meet the target error $2^{-\lambda}$. The following lemma gives sufficient field sizes from the security bounds.

Lemma 3 (Field Size for a Target Statistical Error). *Let $\lambda > 0$ and $m \geq h-1$. The statistical error of SynTrace in Theorem 5 is at most $2^{-\lambda}$ if*

$$\log_2 q \geq \frac{\lambda + m + 1 + \log_2 k}{m - h + 2}.$$

For the Shamir-based construction with $k = t + d + 1$, we have $h = d + 1$. Thus, for $m \geq d$, it suffices to choose

$$\log_2 q \geq \frac{\lambda + m + 1 + \log_2 k}{m - d + 1}.$$

Proof. We first simplify the bound on statistical error from Theorem 5. Since $q \geq 3$, $k \geq 2$ and $h \geq 2$, it gives

$$\varepsilon \leq \left(k \frac{q^{h-1} - 1}{q - 1} + 1 \right) 2^m q^{-m} \leq \frac{2^{m+1} k}{q^{m-h+2}}.$$

To make this upper bound at most $2^{-\lambda}$, it is enough to require

$$q^{m-h+2} \geq 2^{\lambda+m+1} k.$$

Taking base-two logarithms and dividing by $m - h + 2 > 0$ gives the first field-size condition. Substituting $h = d + 1$ gives the condition for the Shamir-based construction. □

The probability of accepting an incorrect value when at most d servers deviate can also be bounded separately. Theorem 6 bounds this probability by dq^{-m}, so $\log_2 q \geq (\lambda + \log_2 d)/m$ suffices to make it at most $2^{-\lambda}$.

For the numerical estimates, choose $q = 2^\ell$. Let N_{db} be the database payload in bits. Put

$$n_0 = \lceil N_{\mathrm{db}}/\ell \rceil, \qquad \ell_Q = \lceil \sqrt{n_0} \rceil, \qquad \ell_A = \lceil n_0/\ell_Q \rceil,$$

and pad the database with zeros to $\ell_Q \ell_A \ell$ bits. To include batches of B retrievals sharing m audit queries, let $D = B + m$, with $B = 1$ for a single retrieval. The upload, download and total communication in bits are

$$\mathsf{upload} = kD\ell_Q\ell, \qquad \mathsf{download} = kD\ell_A\ell, \qquad \mathsf{total} = kD(\ell_Q + \ell_A)\ell. \tag{2}$$

At fixed N_{db}, the balanced cost is approximately $2kD\sqrt{N_{\mathrm{db}}\ell}$; at fixed n field elements it is instead approximately $2kD\sqrt{n}\,\ell$. These estimates fix the database bit length, but the size of a reconstructed row depends on the field. The comparison in Section 6.3 also fixes the requested record size.

Increasing m sends more queries but allows a smaller field q for the same statistical error, creating a tradeoff between the number of queries and the number of bits per field element. Lemma 5 in Appendix D.1 analyzes the choice of m that minimizes the resulting communication estimate.

Table 4. Communication for one retrieval from a 1 GiB database with $(k, t, d) = (7, 4, 2)$ and statistical error at most 2^{-128}, where $q = 2^\ell$.

m	$\log_2 q$	per server	total ($k = 7$)
2	134	786 KiB	5.37 MiB
3	68	**746 KiB**	**5.10 MiB**
4	46	767 KiB	5.25 MiB
5	35	803 KiB	5.49 MiB
6	28	838 KiB	5.73 MiB

Table 4 compares $m = 2, 3, 4, 5, 6$. Communication decreases from $m = 2$ to $m = 3$, then increases as additional queries outweigh the savings from a smaller field.

The same choice of parameters also determines the local computation and storage costs. For the balanced database layout, each server performs D vector–matrix products, requiring $O(Dn)$ field operations in total. At the $m = 3$ configuration in Table 4, the multiplication and addition counts per server are both $\approx 5 \times 10^8$. The server can compute all responses in one pass over the database using $O(D\sqrt{n})$ field elements to accumulate the answers.

Using dense arithmetic, the client's total computation is

$$O\big(D^3 + kD(D + k)\sqrt{n} + k^3\big)$$

field operations, including query generation, mixing, identification and reconstruction. For fixed k and D, this is $O(\sqrt{n})$, compared with $O(n)$ work per server. Larger batches reduce communication per retrieval but increase the client's mixing cost. The client stores the mixing matrix and blinding scalars in $O((D^2 + kD)\ell)$ bits, in addition to the queries and responses.

6.3 Comparison with EKN22-TCC

We compare SynTrace with EKN22-TCC [9] because both provide robustness and accountability in one communication round, with perfect query privacy and the same optimal number of servers. Table 2 compares communication for a fixed record size. We use the homomorphic-MAC construction of [9, Theorem 7, Section 8.1 and Appendix F]. Both schemes use $(k, t, d) = (5, 2, 2)$ and reconstruct from any $u = 3$ correct answers. Since $d = t$ in this comparison, the above-threshold condition for guaranteed abort is vacuous. Its cost when $d < t$ must instead be evaluated using Lemma 3.

We retrieve one 32 KiB record from a 1 GiB database of 32,768 records. Both schemes encode each record as one row over their chosen field and reconstruct the entire row. For SynTrace, the identification bound covers all answer coordinates jointly and thus the whole record.

For EKN22-TCC, choosing $q = 2^{43}$ gives error below 2^{-40} and total communication of 8.96 MiB. Appendix D.2 derives the field size and communication costs.

For SynTrace, choose $q = 2^{15}$ and $m = 4$. Each record has 17,477 symbols and 11 padding bits. Since at most $d = t = 2$ servers can deviate, Theorem 3 and correct reconstruction after identification give

$$\varepsilon_{\mathsf{SynTrace}} \le (q+1)2^4 q^{-4} < 4.55 \cdot 10^{-13} < 2^{-40}.$$

The upload is 1.46 MiB and the download is 0.78 MiB. Thus the total communication is 2.25 MiB, compared with 8.96 MiB for EKN22-TCC, a reduction of 74.95% computed from the unrounded counts.

6.4 Batched Retrieval

SynTrace supports batching by sharing m audit queries across B independent retrieval queries. For each server, the client blinds all $B + m$ queries and mixes them using a single $(B+m)\times(B+m)$ matrix U. The shared audits identify deviating servers, allowing reconstruction of all requested records from the remaining servers in one communication round.

Executing SynTrace separately for each record uses $B(m + 1)$ query and answer components per server, whereas a batch uses only $B + m$. For the same field and record layout, the total batch communication is

$$k(B + m)(\ell_Q + \ell_A)\log_2 q \quad \text{bits.}$$

The per-record cost is $1 + m/B$ times that of the base PIR, and the total is $(B + m)/(B(m + 1))$ times that of separate SynTrace executions.

Table 5 uses the record layout and parameters of Table 2 for both batched and separate executions. At $B = 16$, sharing the audits reduces total communication by 75% and the cost per record from 2.25 MiB to 0.56 MiB. As B increases, the cost per record approaches 0.45 MiB, the communication of the base PIR over this field.

Table 5. Communication for batches of 32 KiB records from a 1 GiB database, with $(k, t, d) = (5, 2, 2)$, $q = 2^{15}$ and $m = 4$.

B	Separate executions (MiB)	Batch total (MiB)	Batch per record (MiB)
1	2.25	2.25	2.25
2	4.49	2.70	1.35
4	8.98	3.59	0.90
8	17.97	5.39	0.67
16	35.94	8.98	0.56
32	71.88	16.17	0.51
64	143.75	30.55	0.48

The whole-batch failure bound is independent of B (Theorem 7, Appendix B). Exact identification allows every requested record to be reconstructed, so one identification-failure bound covers the entire batch. For Table 5, at most $d = t = 2$ servers deviate, and the probability of failing to identify them exactly and reconstruct all records is at most

$$(q + 1)2^4 q^{-4} < 2^{-40}$$

for every B, without increasing the field size or the number of audit queries.

7 Conclusion and Open Problems

SynTrace provides robustness, accountability and guaranteed abort for audit-compatible linear PIR through hidden audit queries and code-based identification. With Shamir-based PIR, it uses $k = t + d + 1$ servers, the minimum needed to tolerate d missing responses under the stated privacy and download assumptions. At this number of servers, $m = d$ audits are necessary and sufficient for the specified syndrome-span decoder class. Additional audits can reduce the required field size and hence communication in bits.

The record comparison shows a 74.95% communication reduction over the homomorphic-MAC construction of [9] at the same (k, t, d) and target error. Sharing m audits across B retrievals in one round reduces the per-record cost to

$1 + m/B$ times that of the base PIR over the same field. The bound on failure of exact identification and reconstruction of the entire batch is independent of B.

Open questions include proving the server lower bound without the download restriction and reducing audit or communication costs through other identification methods.

References

1. Alon, B., Beimel, A.: On the definition of malicious private information retrieval. In: 6th Conference on Information-Theoretic Cryptography (ITC). Leibniz International Proceedings in Informatics (LIPIcs), vol. 343, pp. 8:1–8:23 (2025). https://doi.org/10.4230/LIPIcs.ITC.2025.8, full version: IACR ePrint 2025/683
2. Beelen, P., Bossert, M., Puchinger, S., Rosenkilde, J.: Structural properties of twisted Reed–Solomon codes with applications to cryptography. In: 2018 IEEE International Symposium on Information Theory (ISIT). pp. 946–950. IEEE (2018). https://doi.org/10.1109/ISIT.2018.8437923
3. Beelen, P., Puchinger, S., Rosenkilde, J.: Twisted Reed–Solomon codes. IEEE Transactions on Information Theory **68**(5), 3047–3061 (2022). https://doi.org/10.1109/TIT.2022.3146254
4. Beimel, A., Stahl, Y.: Robust information-theoretic private information retrieval. Journal of Cryptology **20**(3), 295–321 (2007)
5. Cao, Q., Tran, H.Y., Dau, S.H., Yi, X., Viterbo, E., Feng, C., Huang, Y.C., Zhu, J., Kruglik, S., Kiah, H.M.: Committed private information retrieval. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds.) Computer Security – ESORICS 2023. vol. 14344, pp. 393–413. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-50594-2_20
6. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. Journal of the ACM **45**(6), 965–981 (Nov 1998). https://doi.org/10.1145/293347.293350
7. Devet, C., Goldberg, I., Heninger, N.: Optimally robust private information retrieval. In: USENIX Security Symposium. pp. 269–283 (Aug 2012), https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/devet
8. Eriguchi, R., Kurosawa, K., Nuida, K.: Multi-server PIR with full error detection and limited error correction. In: 3rd Conference on Information-Theoretic Cryptography (ITC 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 230, pp. 1:1–1:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.ITC.2022.1, full version: IACR ePrint 2022/500
9. Eriguchi, R., Kurosawa, K., Nuida, K.: On the optimal communication complexity of error-correcting multi-server PIR. In: Theory of Cryptography. vol. 13749, pp. 60–88. Springer Nature Switzerland (2022). https://doi.org/10.1007/978-3-031-22368-6_3, full version: IACR ePrint 2022/1206
10. Eriguchi, R., Kurosawa, K., Nuida, K.: Efficient and generic methods to achieve active security in private information retrieval and more advanced database search. In: Advances in Cryptology – EUROCRYPT 2024. vol. 14655, pp. 92–121. Springer Nature Switzerland (2024). https://doi.org/10.1007/978-3-031-58740-5_4, full version: IACR ePrint 2024/375

11. Goldberg, I.: Improving the robustness of private information retrieval. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P). pp. 131–148 (2007). https://doi.org/10.1109/SP.2007.23
12. Kurosawa, K.: How to correct errors in multi-server PIR. In: Advances in Cryptology – ASIACRYPT 2019. Lecture Notes in Computer Science, vol. 11922, pp. 564–574. Springer (2019). https://doi.org/10.1007/978-3-030-34621-8_20
13. Sun, H., Jafar, S.A.: The capacity of robust private information retrieval with colluding databases. IEEE Transactions on Information Theory **64**(4), 2361–2370 (2018). https://doi.org/10.1109/TIT.2017.2777490
14. Yao, X., Liu, N., Kang, W.: The capacity of multi-round private information retrieval from Byzantine databases. In: IEEE International Symposium on Information Theory (ISIT). pp. 2124–2128 (2019). https://doi.org/10.1109/ISIT.2019.8849275
15. Zhang, L.F., Safavi-Naini, R.: Verifiable multi-server private information retrieval. In: Applied Cryptography and Network Security (ACNS). Lecture Notes in Computer Science, vol. 8479, pp. 62–79 (2014). https://doi.org/10.1007/978-3-319-07536-5_5

Supplementary Material

A Security Proofs

This appendix first derives the consequences of the security definition and verifies the linear PIR properties and audit compatibility of the Shamir-based construction. It then proves the bounds for accountability and guaranteed abort in Section 5 and the lemma connecting output bounds to real/ideal security. It also bounds the probability of accepting an incorrect value and explains how puncturing and blinding enter the analysis.

A.1 Proof of Proposition 1

Proof. For fixed (x, T), the deviation set is a deterministic function of the coalition's view in both executions. Appending it to the compared tuples therefore preserves the statistical-distance bound. In the ideal world, none of the failure events in (i)–(v) can occur. By the statistical-distance bound, each occurs with probability at most ε in the real world.

For the last claim, the simulator produces the simulated view and $\mathcal{M}^\star$ from (T, x, z) without access to i. The ideal identification set and abort decision are functions of $\mathcal{M}^\star$, so their joint distribution with the simulated view and $\mathcal{M}^\star$ does not depend on the requested index. The corresponding real distributions for i and i' are each within ε of this same ideal distribution. The triangle inequality gives the bound 2ε. □

A.2 Linear PIR Properties of the Shamir-Based Construction

Proof (Proposition 2). Reconstruction. Correct retrieval answers are evaluations of a polynomial g of degree at most t with constant term x_i. Any $t+1$ answers at distinct evaluation points determine g, so Rec_J outputs $g(0) = x_i$ whenever $|J| \geq t+1$.

Query privacy. Fix a coalition T of size t. For each coordinate of a shared vector, its shares are obtained by adding the fixed secret coordinate to the image of uniform random coefficients under the matrix

$$V_T = (\alpha_j^s)_{j\in T,\, s\in[t]} = \mathrm{diag}(\alpha_j)_{j\in T}(\alpha_j^{s-1})_{j\in T,\, s\in[t]}.$$

The diagonal matrix is invertible because the evaluation points are nonzero. The second matrix is Vandermonde and is invertible because the points are distinct. Thus V_T maps uniform coefficients to uniform shares, and adding the secret coordinate preserves uniformity. The sharing randomness is independent across coordinates, so the coalition's query tuple is uniform on $(\mathbb{F}_q^n)^t$ for every shared vector. In particular, sharing e_i gives a distribution independent of i. For $|T| < t$, taking the corresponding marginal gives the uniform distribution on $(\mathbb{F}_q^n)^{|T|}$.

Linearity. The query space $\mathcal{Q} = \mathbb{F}_q^n$ is closed under linear combinations. For each fixed database x, the scalar answer map $\mathsf{Ans}_j(q, x) = \langle q, x \rangle$ is linear in q. □

A.3 Audit Compatibility of the Shamir-Based Construction

Proof (Proposition 3). We use the audit queries defined in Figure 3. By Proposition 2, it remains to verify the three audit-compatibility conditions.

Query uniformity. Any coalition T of at most t servers receives jointly uniform shares of a fixed vector, as shown in the proof of Proposition 2. This applies to both the retrieval queries sharing e_i and the audit queries sharing zero, giving the required uniform distributions on $\mathcal{Q}^{|T|}$.

Audit answer relations. For an audit query, the answer at server j is $g(\alpha_j)$, where

$$g(z) = \sum_{s=1}^{t} \langle b_s, x \rangle z^s.$$

Since $\deg g \le t$ and $g(0) = 0$, the answer vector belongs to the audit answer code

$$\mathcal{C} = \{(g(\alpha_1), \ldots, g(\alpha_k)) : \deg g \le t,\ g(0) = 0\}.$$

Choose H to be a full-row-rank parity-check matrix of $\mathcal{C}$. This matrix depends only on the public parameters and annihilates every correct audit answer vector.

Column independence. Writing $g(z) = zh(z)$ with $\deg h \le t - 1$ shows that $\mathcal{C}$ is a generalized Reed–Solomon code of dimension t with multipliers α_j. Hence

$$\operatorname{dist}(\mathcal{C}) = k - t + 1 \ge d + 2,$$

where the inequality follows from $k \ge t + d + 1$. Thus every $d + 1$ columns of H are linearly independent, establishing the final condition. □

A.4 Proof of Theorem 3

Proof. The matrix E collects the audit-answer errors from these remaining deviating servers. Each row corresponds to one server and contains its errors in all m audits. Writing $a^\star_{r,j}$ for the correct base-PIR answer to $q_{r,j}$, the row for server j is

$$\left(\begin{bmatrix} a_{1,j} \\ \vdots \\ a_{m,j} \end{bmatrix} - \begin{bmatrix} a^\star_{1,j} \\ \vdots \\ a^\star_{m,j} \end{bmatrix} \right)^\top, \qquad j \in \mathcal{M}.$$

Each audit contributes a block of ℓ_A columns, so $E \in \mathbb{F}_q^{w \times m\ell_A}$. Let $H_{\mathcal{M}}$ consist of the columns of H belonging to these servers. The audit-answer relations remove the correct answers from the syndromes, giving

$$[S_1 \mid \cdots \mid S_m] = H_{\mathcal{M}} E. \tag{3}$$

Since $w \le d$ and any $d + 1$ columns of H are independent, the columns of $H_{\mathcal{M}}$ are independent. The syndrome columns therefore span all of them exactly when

E has full row rank. Theorem 2 guarantees that no correctly responding server is falsely identified as deviating. Therefore,

$$L = \mathcal{M} \quad \Longleftrightarrow \quad \operatorname{rank}(E) = w.$$

When $L = \mathcal{M}$, excluding these servers leaves at least $k - w \geq u$ correct responses, so the verification checks pass and the client recovers x_i.

Condition on the coalition's queries and internal randomness, fixing its response errors while U and the blinding scalars remain independent and uniform by Proposition 4. Let Δ_j denote the received response minus the correct response at server j. By linearity, audit r's error is row r of $U^{-1}\Delta_j$ multiplied by $\beta_{r,j}^{-1}$.

To bound $\Pr[c^\top E = 0]$, fix a nonzero $c \in \mathbb{F}_q^w$ and choose j_0 with $c_{j_0} \neq 0$. For fixed U, if row r of $U^{-1}\Delta_{j_0}$ is nonzero, cancellation determines at most one value of β_{r,j_0}^{-1} after fixing the other scalars in that audit. The relation holds with probability at most $1/(q-1)$ for a nonzero row and at most 1 for a zero row. These are the two cases in (1).

For fixed U, independent blinding across audits lets us multiply these bounds. A zero row increases its factor from $1/(q-1)$ to 1, adding $(q-2)/(q-1)$. Expanding the product, each term selects a set $R \subseteq [m]$ of rows that must all be zero. To bound this probability, choose any nonzero column v of Δ_{j_0}. A uniform invertible U makes $U^{-1}v$ uniform among the $q^{m+1} - 1$ nonzero vectors. If the rows indexed by R are all zero, then the same coordinates of $U^{-1}v$ must be zero. There are $q^{m+1-|R|} - 1$ such nonzero vectors, so the probability is at most

$$\frac{q^{m+1-|R|} - 1}{q^{m+1} - 1} \leq q^{-|R|}. \tag{4}$$

Averaging the expanded product over U therefore gives

$$\begin{aligned}\Pr[c^\top E = 0] &\leq \frac{1}{(q-1)^m} \sum_{R \subseteq [m]} (q-2)^{|R|} q^{-|R|} \\ &= \frac{1}{(q-1)^m} \left(1 + \frac{q-2}{q}\right)^m = \left(\frac{2}{q}\right)^m.\end{aligned}$$

A union bound over the $(q^w - 1)/(q-1)$ coefficient vectors up to scaling gives the rank-deficiency bound. Since this bound increases with w, it also holds with the original deviation count before preprocessing. Averaging over the conditioned queries and randomness proves the theorem. □

A.5 Proof of Theorem 4

Proof. Condition on the coalition's queries and internal randomness, fixing its response errors while U and the blinding scalars remain independent and uniform by Proposition 4. Write H_j for the column of H corresponding to server j, and let $H_{\mathcal{M}}$ collect these columns for the deviating servers. As in the proof of Theorem 3,

$$[S_1 \mid \cdots \mid S_m] = H_{\mathcal{M}} E.$$

Thus the syndrome space V is contained in the span of the columns of $H_{\mathcal{M}}$, even when $w > d$.

For part (i), suppose the threshold check passes. Since $|L| \le d < w$, some deviating server j is not identified, so H_j lies outside V. We can choose $\nu \in \mathbb{F}_q^h$ whose inner product is zero on every vector in V but nonzero on H_j. Consequently,

$$c = H_{\mathcal{M}}^{\top}\nu \neq 0, \qquad c^{\top}E = 0.$$

The proof of Theorem 3 bounds the probability of this relation by F for each fixed ν with $c \neq 0$. Nonzero scalar multiples of ν give the same relation, leaving $(q^h - 1)/(q - 1)$ possibilities. A union bound proves part (i),

$$\Pr[\text{the threshold check passes}] \le \min\left\{1, \frac{q^h - 1}{q - 1}F\right\}.$$

For part (ii), the client must also pass verification to return a value. If $L \neq \emptyset$, choose an identified server $j \in L$. Since $H_j \in V$, the vector ν above also satisfies $\nu^{\top}H_j = 0$. Every column of H is nonzero by column independence. This extra equation therefore restricts ν to a space of dimension $h-1$, containing $(q^{h-1}-1)/(q-1)$ nonzero vectors up to scaling. A union bound over these vectors and all k servers gives

$$\Pr[\text{the client returns a value with } L \neq \emptyset] \le k\frac{q^{h-1} - 1}{q - 1}F.$$

If the client returns a value with $L = \emptyset$, verification requires $V = \{0\}$. Fix one ν with $H_{\mathcal{M}}^{\top}\nu \neq 0$, which exists because $\mathcal{M}$ is nonempty and H has no zero columns. The event $V = \{0\}$ implies $\nu^{\top}H_{\mathcal{M}}E = 0$, whose probability is at most F by the same single-relation bound. Summing the two cases yields

$$\Pr[\text{the client returns a value}] \le \min\left\{1, \left(k\frac{q^{h-1} - 1}{q - 1} + 1\right)F\right\} \le \varepsilon_{\text{abort}}.$$

Preprocessing can only decrease k and h, so both bounds also hold with the original parameters. Averaging over the conditioned queries and randomness at the original deviation count proves the theorem. □

A.6 Proof of Lemma 2

Proof. Given (T, x, z), the simulator runs $\mathsf{Query}(1)$ and gives the coalition's queries to $\mathcal{A}$. It records the resulting view and compares each response with the correct answer $\mathsf{Ans}_j(q_j, x)$ to determine $\mathcal{M}^{\star}$. It then submits this set to $\mathcal{F}_{\mathsf{PIR}}^{\mathsf{acc}}$. This is the ideal execution described in Section 3.2, and it does not require the requested index.

Perfect query privacy gives the same coalition-view distribution for index 1 as for the actual index i. We may therefore couple the real and ideal views to be

equal. Since the database and the query-response pairs determine the deviation set, the coupled executions also have the same $\mathcal{M}$. The ideal tuple is then

$$\begin{cases} (\mathsf{view}, x_i, \mathcal{M}), & |\mathcal{M}| \le d, \\ (\mathsf{view}, \perp, \emptyset), & |\mathcal{M}| > d. \end{cases}$$

The tuples differ only when one of the stated output guarantees fails. Their statistical distance is therefore at most $\varepsilon_1 + \varepsilon_2$. If the bounds hold conditionally on each deviation count, the conditional failure probability is always at most $\max\{\varepsilon_1, \varepsilon_2\}$. Averaging over the count preserves this bound.

Conversely, the deviation set is determined by the view and x in both executions, so adding it to the compared tuples does not increase their statistical distance. Neither failure event can occur in the ideal execution. The statistical-distance bound limits each real-world probability to ε. □

A.7 Incorrect-Output Probability below the Threshold

Failure to identify all deviations need not lead to an incorrect retrieved value. We bound the probability of such an output separately from identification failure.

Theorem 6 (Incorrect-output probability). *For every adversary and every $w \le d$ occurring with positive probability,*

$$\Pr[\mathsf{out} \notin \{x_i, \perp\} \mid |\mathcal{M}| = w] \le w q^{-m},$$

irrespective of whether identification is exact. For scalar answers the bound improves to $w(q-1)/(q^{m+1}-1)$.

Proof. An incorrect output requires the client to accept a response from a deviating server that it did not identify. We show that passing verification in this case forces all audit rows of that server's unmixed deviation to vanish.

Condition on the coalition's received queries and the adversary's random tape. This fixes the erased set and all deviation matrices, while U and the blinding scalars remain independently uniform by Proposition 4. Apply Lemma 4 to remove the erased positions. Write $\mathcal{M}' = \mathcal{M} \setminus L_\perp$ for the deviating servers among the remaining positions P, and form the residual matrix E on these servers.

Suppose the client delivers a value. Let $L' = L \cap P$ be the identified servers among the responding positions. Every column indexed by L' lies in V by the identification rule. These columns are independent because $L' \subseteq \mathcal{M}'$ by Theorem 2 and the punctured code retains the required column independence. The verification check gives $\dim V = |L'|$, so

$$V = \operatorname{span}\{(H_P)_{*,j} : j \in L'\}.$$

On the other hand, the residual factorization (3) holds with H_P and $\mathcal{M}'$ in place of H and $\mathcal{M}$. The columns of H_P indexed by $\mathcal{M}'$ are independent. Every vector

in V therefore has a unique representation in those columns, and its coefficients outside L' must be zero. Applied to each syndrome column, this implies

$$E_{j,(r,\ell)} = 0 \quad \text{for all } j \in \mathcal{M}' \setminus L',\ r \in [m],\ \ell \in [\ell_A].$$

Since every blinding scalar is nonzero, the same condition can be written as

$$(U^{-1}\Delta_j)_{[m],*} = 0 \qquad \text{for every } j \in \mathcal{M}' \setminus L'.$$

If no deviating server remains unidentified, all responses used in reconstruction are correct and at least u remain, so the output is x_i. An incorrect output thus implies that the last displayed event occurs for at least one server in the fixed set $\mathcal{M}'$. For each such server, choose a nonzero column v of Δ_j. The vector $U^{-1}v$ is uniform among the $q^{m+1}-1$ nonzero vectors. Only $q-1$ of them have all m audit coordinates equal to zero, so the event has probability at most $(q-1)/(q^{m+1}-1) \le q^{-m}$. A union bound gives

$$\begin{aligned}\Pr[\mathsf{out} \notin \{x_i, \perp\}] &\le \sum_{j \in \mathcal{M}'} \Pr[(U^{-1}\Delta_j)_{[m],*} = 0] \\ &\le |\mathcal{M}'| q^{-m} \le w q^{-m},\end{aligned}$$

where all probabilities are still conditional on the fixed queries and random tape. For scalar answers, this count gives the exact single-server probability $(q-1)/(q^{m+1}-1)$ and hence the sharper bound. Both bounds hold for every conditioned execution, so averaging proves the theorem. □

A.8 Puncturing and Verification

The security analysis removes missing or malformed responses before examining the remaining audit syndromes. We first justify this reduction, then explain why the additional verification check is needed for guaranteed abort.

Lemma 4 (Erasures). *Let $e = |L_\perp|$ and $h = k - \dim \mathcal{C}$. If $e > d$, the client returns $(\perp, \emptyset)$ at reconstruction step 1. Otherwise $L_\perp \subseteq \mathcal{M}$, and the audit code restricted to the responding servers $P = [k] \setminus L_\perp$ satisfies*

$$\dim(\mathcal{C}|_P) = k - h, \qquad \operatorname{dist}(\mathcal{C}|_P) \ge d + 2 - e.$$

Its length is $k - e$, so its parity-check matrix H_P has $h_e = h - e \ge 0$ rows. The identification and probability arguments in Section 5 therefore apply to the remaining servers with deviation set $\mathcal{M} \setminus L_\perp$ and threshold $d - e$. This remaining threshold may be zero.

Proof. A missing or malformed response is a deviation by the definition in Section 3.1. Thus $L_\perp \subseteq \mathcal{M}$, and the protocol aborts immediately if $e > d$.

Now suppose $e \le d$. The audit code has minimum distance at least $d+2 > e$. No nonzero codeword can be supported entirely on the e erased positions, so

restriction to P is injective and preserves the dimension. Deleting e coordinates reduces the weight of any codeword by at most e, giving

$$\operatorname{dist}(\mathcal{C}|_P) \geq \operatorname{dist}(\mathcal{C}) - e \geq (d-e) + 2.$$

The restricted code consequently has the distance required for threshold $d-e$. Its parity-check rank is its length minus its dimension, $(k-e)-(k-h) = h-e$, which is nonnegative. □

Remark 1 (Zero-filling is not a substitute). Replacing each erased response by zero and keeping the original matrix H can cause an unnecessary abort. To see this, consider a single erasure $L_\perp = \{j\}$ with every other server responding correctly. Zero-filling introduces the residual $-a^\star_{r,j}$ in each audit, giving

$$S_r = -H_{*,j}(a^\star_{r,j})^\top.$$

Unless all these audit answers vanish, V is the one-dimensional span of $H_{*,j}$. The distance bound $\operatorname{dist}(\mathcal{C}) \geq 3$ makes every other column independent of $H_{*,j}$, so no responding server is identified. The verification check then rejects because $\dim V = 1$ but $|L \cap P| = 0$.

For the Shamir instantiation with scalar answers and a nonzero database, the erased server's audit answers are independent uniform field elements. The resulting unnecessary abort therefore has probability $1 - q^{-m}$.

Puncturing preserves the audit constraints, but checking the number of identified servers alone need not detect an execution above the threshold. The following result explains the role of the additional verification check.

Proposition 5 (When the threshold check cannot reject). *Let $e = |L_\perp| \leq d$ and suppose answers are scalar. The audit syndrome space satisfies $\dim V \leq m$. If $m \leq d-e$, then $|L| \leq d$, so the threshold check at reconstruction step 4 cannot cause an abort.*

In particular, with no erasures and $m = d$, guaranteed abort depends on the verification check. Theorem 4 bounds its failure probability by $O(k2^d q^{h-2-d})$, which is $O(k2^d/q)$ when $h = d+1$, as in the Shamir instantiation with $k = t+d+1$.

Proof. With scalar answers, each audit contributes one syndrome vector. These m vectors span V, so $\dim V \leq m$.

Suppose $m \leq d-e$. Each identified responding server contributes a column of H_P lying in V. If $d-e+1$ such servers were identified, their columns would lie in a space of dimension at most $d-e$ and hence be dependent. This contradicts the distance bound in Lemma 4, which makes every $d-e+1$ columns independent. Therefore

$$|L \cap P| \leq d-e, \qquad |L| = e + |L \cap P| \leq d$$

at the threshold check. The final probability bound follows by substituting $m = d$ into Theorem 4. □

B Batched Retrieval

The communication saving in Section 6.4 relies on sharing the same audit queries across all requested items. The following theorem bounds failure for the entire batch.

Theorem 7 (Batched SynTrace). *Let Π^{base} satisfy Definition 3 for (t, d), with reconstruction threshold u and answer length $\ell_A \geq 1$, let $k - d \geq u$, $m \geq d$ and $B \geq 1$, and fix a coalition of size at most t. Generate retrieval queries for indices $i_1, \ldots, i_B$ through B independent calls to* Query*, and audit queries through m independent calls to* Audit*. Apply* SynTrace *to the $D = B + m$ queries for each server, using $U \leftarrow \mathrm{GL}_D(\mathbb{F}_q)$ and an independent nonzero blinding scalar for each query. The identification and verification steps use the answers to the m audit queries; reconstruction uses the same remaining server set for all B requested items. Put*

$$F_{B,m,q} := \frac{2^m q^B - 1}{q^{B+m} - 1} \leq 2^m q^{-m},$$
$$G_{B,m,q} := \frac{q^B - 1}{q^{B+m} - 1} < q^{-m}.$$

For every fixed nonzero dependence direction c of the residual matrix, $\Pr[c^\top E = 0] \leq F_{B,m,q}$, for every $\ell_A \geq 1$. Condition on any deviation count w occurring with positive probability. If $w \leq d$, $L \subseteq \mathcal{M}$ with probability 1 and

$$\Pr[L \neq \mathcal{M} \mid |\mathcal{M}| = w] \leq \min\left\{1, \frac{q^w - 1}{q - 1} F_{B,m,q}\right\}.$$

Exact identification allows the client to recover the entire requested batch. Independently of whether identification succeeds, the probability that the client delivers a batch with any *incorrect component is at most $wG_{B,m,q}$. Above the deviation threshold, the client outputs $(\perp, \emptyset)$ except with probability at most*

$$\varepsilon_B := \min\left\{1, \left(k \frac{q^{h-1} - 1}{q - 1} + 1\right) F_{B,m,q}\right\}, \qquad h = k - \dim \mathcal{C}.$$

The scheme realizes the batched version of $\mathcal{F}^{\mathsf{acc}}_{\mathsf{PIR}}(k, t, d)$, which takes an index tuple and returns $((x_{i_1}, \ldots, x_{i_B}), \mathcal{M}^\star)$ within the deviation threshold and $(\perp, \emptyset)$ above it, with error at most ε_B. The real/ideal comparison and simulator consistency requirement are those of Definition 1, with a batch-valued out*. Its communication is $B + m$ base-PIR query and answer components, giving a per-item factor $1 + m/B$ relative to the same base PIR over the same field.*

Proof. By the query-distribution condition of Definition 3, independent calls to Query give a product-uniform distribution of the coalition's queries for every requested index tuple. Proposition 4 therefore applies with D queries per server. Condition on the received queries and adversarial randomness, which fixes the deviations and erasures while preserving uniform independent mixing and blinding. Remove erasures as in Lemma 4; it suffices to analyze the remaining nonzero deviation matrices.

Fix $c \neq 0$ and $j_0 \in \mathrm{supp}(c)$. Write A for the set of m audit coordinates and let Z count the rows of $(U^{-1}\Delta_{j_0})_{A,*}$ that vanish. The fixed-direction argument in Appendix A gives the conditional bound $(q-1)^{-(m-Z)}$. Choose a nonzero column δ of Δ_{j_0}. For every $R \subseteq A$,

$$\Pr[(U^{-1}\Delta_{j_0})_{R,*} = 0] \leq \Pr[(U^{-1}\delta)_R = 0] = \frac{q^{D-|R|}-1}{q^D-1}.$$

Since $q \geq 3$, the coefficients in the expansion of $(q-1)^Z$ are nonnegative, and hence

$$\begin{aligned}\mathbb{E}[(q-1)^Z] &\leq \sum_{R\subseteq A} (q-2)^{|R|} \frac{q^{D-|R|}-1}{q^D-1} \\ &= \frac{q^D(2(q-1)/q)^m - (q-1)^m}{q^D-1} = (q-1)^m F_{B,m,q}.\end{aligned}$$

Multiplication by $(q-1)^{-m}$ proves the fixed-direction bound for every answer length. The projective union bound gives identification, and the deterministic support argument still excludes false identification within the deviation threshold.

If the verification check permits delivery while a deviating server j remains unexcluded, the argument of Theorem 6 forces all audit rows of $U^{-1}\Delta_j$ to vanish. For any fixed nonzero column of Δ_j, this implies that its image under U^{-1} is supported on the B retrieval coordinates, an event of probability $G_{B,m,q}$. A union bound over at most w deviating servers bounds the probability of any such hidden deviation. On the complementary event every remaining response is correct, so base-PIR reconstruction recovers all B requested items. This single event controls the whole batch and introduces no factor of B. Exact identification also makes both conditions of the verification check hold, since $k - d \geq u$.

For outputs above the threshold, repeat the witness count of Theorem 4 using $F_{B,m,q}$. More than d erasures cause immediate output $(\perp, \emptyset)$. As $h \geq d+1$, ε_B dominates the identification bound within the deviation threshold. The simulation argument of Lemma 2 uses a fixed dummy index tuple and computes $\mathcal{M}^\star$ from the complete simulated responses, satisfying the same consistency requirement. Averaging the conditional output bounds over w then gives realization error at most ε_B. Finally, $F_{B,m,q} \leq 2^m q^{-m}$ follows from $2^m \leq q^m$. □

Corollary 1. *Instantiate Theorem 7 with the Shamir construction, or an ideal MDS construction, at $k = t + d + 1$ and $m \geq d$. For every $B \geq 1$ the per-item communication factor is $1 + m/B$ relative to the same base PIR over the same field, and the batched realization error is at most* $\min\{1, (k\frac{q^{h-1}-1}{q-1}+1)2^m q^{-m}\}$, *independently of B.*

Proof. These constructions have $u = t+1$, $h = d+1$ and $k - d = u$. Apply Theorem 7 and $F_{B,m,q} \leq 2^m q^{-m}$. □

C Extension to MDS-Based PIR

Polynomial evaluation is not essential to the compiler. The argument for Shamir-based PIR applies to ideal threshold sharing obtained from an MDS code, with the secret occupying one coordinate. Setting that coordinate to zero defines the audit code.

Proposition 6 (Audit codes from MDS linear secret sharing). *Let $\mathcal{D} \subseteq \mathbb{F}_q^{\{0\}\cup[k]}$ be a $[k+1, t+1, k-t+1]$ MDS code, and let the associated secret-sharing scheme share $s \in \mathbb{F}_q$ by drawing a uniform codeword $c \in \mathcal{D}$ with $c_0 = s$ and giving c_j to server j. Let Π^{base} be the induced PIR, in which the client shares $e_i \in \mathbb{F}_q^n$ coordinatewise and each server returns the inner product of its share vector with x, and let* Audit *share the zero vector in the same way. Then Π^{base} has reconstruction threshold $u = t+1$, its audit code is the shortening $\mathcal{C} = \{c|_{[k]} : c \in \mathcal{D},\ c_0 = 0\}$, a $[k, t, k-t+1]$ MDS code, and Π^{base} is d-audit-compatible if and only if $k \geq t+d+1$.*

Proof. Every $t+1$ coordinates of an MDS code of dimension $t+1$ form an information set. Consequently, fixing the secret coordinate leaves every set of t server shares uniform and independent of that secret. Sharing the selection vector and zero vector coordinatewise gives the query distributions required by Definition 3 on $\mathcal{Q} = \mathbb{F}_q^n$. Linear evaluation and reconstruction from any $t+1$ shares establish the remaining requirements of Definition 2. The zero-secret shortening has dimension t and distance $k-t+1$. Choose H as a full-row-rank parity-check matrix of this code. An audit answer vector is a linear combination of its codewords and is therefore annihilated by H, as required. Column independence is equivalent to $k \geq t+d+1$. □

C.1 A Non-Reed–Solomon Instantiation

Remark 2 (An explicit non-GRS example). We instantiate Proposition 6 with sharing and audit codes outside the generalized Reed–Solomon class, over fields large enough to meet a target statistical error.

Let $\mathbb{F}_8 \subset \mathbb{F}_q$ with $q = 2^{3s}$ and $s \geq 2$, and let γ generate $\mathbb{F}_8^*$. Set $\delta = \gamma^6$ and

$$\mathcal{B} = \{1, \gamma, \gamma^2, \gamma^3, \gamma^4, \gamma^5\} = \mathbb{F}_8^* \setminus \{\delta\}.$$

Use evaluation points $\{0\} \cup \mathcal{B}$, with 0 as the secret coordinate. Choose any $\eta \in \mathbb{F}_q \setminus \mathbb{F}_8$, and let $\mathcal{D}_\eta \subseteq \mathbb{F}_q^7$ be the single-twist twisted Reed–Solomon code with hook 1 and twist degree 4 of [3]. Its defining polynomials are

$$f(x) = a_0 + a_1 x + \eta a_1 x^4 + a_2 x^2 + a_3 x^3, \qquad a_0, a_1, a_2, a_3 \in \mathbb{F}_q,$$

and the sharing code is

$$\mathcal{D}_\eta = \big\{(f(\alpha))_{\alpha \in \{0\}\cup\mathcal{B}} : a_0, a_1, a_2, a_3 \in \mathbb{F}_q\big\}.$$

Its shortening at the secret coordinate, that is the set of f with $a_0 = 0$ restricted to $\mathcal{B}$, is the induced audit code

$$\mathcal{C}_\eta = \{(a_1(x + \eta x^4) + a_2 x^2 + a_3 x^3)|_{\mathcal{B}}\} \subseteq \mathbb{F}_q^6.$$

(a) MDS parameters. The sharing code $\mathcal{D}_\eta$ is $[7, 4, 4]$ MDS and the audit code $\mathcal{C}_\eta$ is $[6, 3, 4]$ MDS, for every $\eta \notin \mathbb{F}_8$ and every such $\mathbb{F}_q$. Both statements rest on the same computation. Let $f \neq 0$ be as displayed. If $a_1 = 0$ then $\deg f \leq 3$, so f has at most three roots, and a codeword of $\mathcal{D}_\eta$ has weight at least $7 - 3 = 4$; if in addition $a_0 = 0$ then $f = x^2(a_2 + a_3 x)$ has at most one root in $\mathcal{B}$, so a codeword of $\mathcal{C}_\eta$ has weight at least 5. Let now $a_1 \neq 0$, so $\deg f = 4$ exactly. If f had four roots in $\{0\} \cup \mathcal{B}$ then $f = \eta a_1 \prod_{i=1}^4 (x - \beta_i)$. When 0 is one of the roots we have $a_0 = 0$ and $f = x(a_1 + a_2 x + a_3 x^2 + \eta a_1 x^3)$, whose second factor has the three roots $\beta_i \in \mathcal{B}$, so comparing constant terms in characteristic 2 gives $a_1 = \eta a_1 \beta_1 \beta_2 \beta_3$ and hence $\eta = (\beta_1 \beta_2 \beta_3)^{-1} \in \mathbb{F}_8^*$. When 0 is not a root, all four β_i lie in $\mathcal{B}$ and comparing the coefficients of x gives $a_1 = \eta a_1 e_3(\beta)$, so $e_3(\beta) \neq 0$ and $\eta = e_3(\beta)^{-1} \in \mathbb{F}_8$, where e_3 is the third elementary symmetric polynomial. Both conclusions contradict $\eta \notin \mathbb{F}_8$. Hence no nonzero codeword of $\mathcal{D}_\eta$ has weight at most 3, and the same argument confined to $\mathcal{B}$ gives the claim for $\mathcal{C}_\eta$; the Singleton bound makes both distances exact.

(b) Separation from generalized Reed–Solomon codes. The audit code $\mathcal{C}_\eta$ is not monomially equivalent to a generalized Reed–Solomon code, including an extended one, for every $\eta \notin \mathbb{F}_8$ and every extension of $\mathbb{F}_8$. The Schur square of a code monomially equivalent to a $[6, 3]$ generalized Reed–Solomon code has dimension $2 \cdot 3 - 1 = 5$, and Schur-square dimension is invariant under monomial equivalence; this is the distinguisher of [2]. Take the basis $g_1 = x + \eta x^4$, $g_2 = x^2$, $g_3 = x^3$ of $\mathcal{C}_\eta$ and let $M(\eta)$ be the 6×6 matrix whose rows are the six pairwise products $g_1^2, g_1 g_2, g_1 g_3, g_2^2, g_2 g_3, g_3^2$ evaluated on $\mathcal{B}$. Since $\mathcal{B} \subseteq \mathbb{F}_8^*$ we may reduce exponents modulo 7, so on $\mathcal{B}$

$$\begin{aligned} g_1^2 &= \eta^2 x + x^2, & g_2^2 &= x^4, \\ g_1 g_2 &= x^3 + \eta x^6, & g_2 g_3 &= x^5, \\ g_1 g_3 &= \eta + x^4, & g_3^2 &= x^6. \end{aligned}$$

Adding g_2^2 to $g_1 g_3$ and ηg_3^2 to $g_1 g_2$ leaves the determinant unchanged and replaces those two rows by the constant η and by x^3; factoring η out of the constant row and expanding the remaining first row by multilinearity gives

$$\det M(\eta) = \eta\big(\eta^2 V_{\{0,1,3,4,5,6\}} + V_{\{0,2,3,4,5,6\}}\big),$$

where V_E denotes the determinant of $(\alpha^e)_{e \in E,\ \alpha \in \mathcal{B}}$. A generalized Vandermonde determinant factors as $V_E = V(\mathcal{B}) s_\lambda(\mathcal{B})$ with $V(\mathcal{B})$ the ordinary Vandermonde determinant and λ the partition obtained from E; here $\lambda = (1^4)$ and $\lambda = (1^5)$, giving

$$\begin{aligned} V_{\{0,1,3,4,5,6\}} &= V(\mathcal{B}) e_4(\mathcal{B}), \\ V_{\{0,2,3,4,5,6\}} &= V(\mathcal{B}) e_5(\mathcal{B}). \end{aligned}$$

In characteristic 2,

$$\prod_{\alpha\in\mathbb{F}_8^*}(x+\alpha) = x^7+1 = (x+\delta)\sum_{i=0}^{6}\delta^i x^{6-i}.$$

Thus $e_i(\mathcal{B}) = \delta^i$ and

$$\det M(\eta) = V(\mathcal{B})\delta^4\eta(\eta^2+\delta), \qquad V(\mathcal{B})\delta^4 \in \mathbb{F}_8^*.$$

This vanishes only for $\eta = 0$ and for the unique η with $\eta^2 = \delta$, and both lie in $\mathbb{F}_8$ because squaring restricts to a bijection of $\mathbb{F}_8$. Hence $\dim \mathcal{C}_\eta^{*2} = 6 \neq 5$ for every $\eta \notin \mathbb{F}_8$. Since shortening a generalized Reed–Solomon code again yields one, $\mathcal{D}_\eta$ is outside the class as well.

(c) Induced parameters. Proposition 6 gives $k = 6$, $t = 3$, $u = t+1 = 4$ and audit code $\mathcal{C}_\eta$, a $[6, 3, 4]$ MDS code; thus $d = 2$, $k = t+d+1$ and $k-d = 4 \geq u$. The induced scheme satisfies Definition 3 with $\dim \mathcal{C}_\eta = t = 3$, $h = 3 = d+1$ and $m = d = 2$. The dimension $t = 3$ is the smallest at which the audit code can leave the class at all: every MDS code of dimension at most 2 is monomially equivalent to a, possibly extended, generalized Reed–Solomon code, since scaling its columns puts a generator matrix in the form $\left(\begin{smallmatrix} 1 & \cdots & 1 \\ \alpha_1 & \cdots & \alpha_k \end{smallmatrix}\right)$ with distinct α_j, with at most one additional column $(0,1)^\top$ representing the point at infinity.

(d) Error parameters. At $q = 64$, Theorem 5 bounds the realization error by

$$\left(6\cdot\frac{64^2-1}{63}+1\right)\frac{4}{64^2} = \frac{391}{1024} \approx 0.382 < 1.$$

The guarantee of Theorem 5 is therefore already non-vacuous at the smallest admissible field. Taking $q = 2^{135}$, which contains $\mathbb{F}_8$ because 3 divides 135, and any $\eta \notin \mathbb{F}_8$, Lemma 3 with $m-h+2 = 1$ gives $\varepsilon_{\text{abort}} \leq 2k2^m q^{-1} = 48\cdot 2^{-135} < 2^{-129}$. Claims (a) and (b) were also verified by exhaustive computation over $\mathbb{F}_{64}$: for all 56 values $\eta \notin \mathbb{F}_8$ both codes are MDS and $\dim \mathcal{C}_\eta^{*2} = 6$, for every $\eta \in \mathbb{F}_8^*$ the sharing code fails to be MDS, and a generalized Reed–Solomon code of the same length and dimension has Schur square of dimension 5.

D Communication Analysis

D.1 Optimizing the Number of Audit Queries and the Field Size

For the Shamir-based construction with $k = t+d+1$, increasing m increases the number of queries but reduces the field size required by Lemma 3. For one retrieval and a fixed database size in bits, the balanced communication estimate in Section 6.2 is proportional to

$$g(m) = (m+1)\sqrt{\frac{\lambda+m+1+\log_2 k}{m-d+1}}, \qquad m \geq d.$$

The factor $m+1$ counts the retrieval and audit queries. The square-root factor comes from the field width in the balanced database layout. This estimate uses continuous matrix dimensions, while Equation (2) gives the cost with integer dimensions and padding.

Lemma 5 (Optimal Number of Audit Queries). *For $\lambda = 128$, $1 \le d \le 6$ and $2d+1 \le k \le 16$, the unique integer minimizer of $g(m)$ is $m = 2d-1$. The same choice minimizes the estimate when the field width is rounded up to an integer.*

Proof. We first locate the minimum when m is treated as a real variable. Write $x = m - d + 1$ and $C = \lambda + d + \log_2 k$. Then

$$g(m)^2 = \frac{(x+d)^2(C+x)}{x}, \qquad \frac{d}{dx}\log g(m)^2 = \frac{2}{x+d} + \frac{1}{C+x} - \frac{1}{x}.$$

The derivative has the same sign as $2x^2 + Cx - dC$. This polynomial has one positive root,

$$x^\star = \frac{\sqrt{C^2 + 8dC} - C}{4}.$$

Thus the estimate decreases up to this root and increases afterwards. Subject to $m \ge d$, its real minimizer is $\max\{d, d-1+x^\star\}$, and its integer minimum is attained at one of the adjacent admissible integers.

For the stated parameter range, evaluating the estimate gives the unique integer minimizer $m = 2d-1$ in all 54 pairs (k, d). We also check the rounded estimate

$$(m+1)\sqrt{\left\lceil \frac{128 + m + 1 + \log_2 k}{m - d + 1} \right\rceil}.$$

Both estimates are at least $m+1$, so any m with $m+1$ exceeding the value at $2d-1$ cannot improve on that choice. This gives a finite search through at most $m = 59$. The rounded estimate also has the unique minimizer $2d-1$ for all 54 pairs. □

D.2 Communication of EKN22-TCC

We derive the EKN22-TCC row of Table 2 using the homomorphic-MAC construction of [9, Theorem 7, Section 8.1 and Appendix F]. For the linear answer map used here, this construction avoids the subset enumeration and repeated executions of their generic transformation in Corollary 6 and is the more efficient of the two baselines for this record layout.

The comparison uses $(k, t, d) = (5, 2, 2)$ and retrieves one 32 KiB record from a 1 GiB database, with target error 2^{-40}. There are $\ell_Q = 32{,}768$ records of $R = 262{,}144$ bits each. Over a base field of width w, a record occupies $\ell_A(w) = \lceil R/w \rceil$ field elements, padding the final element as needed.

For the MAC construction, let the base and authentication fields be $\mathbb{F}_{2^w} \subseteq \mathbb{F}_{2^W}$, where W is a positive integer multiple of w. We omit self-pairs and transmit

only the nonconstant coefficients of the authentication polynomials. The input constant term is already the base query, and an accepted output authentication polynomial must have the claimed answer as its constant term. Omitting self-pairs also preserves the correction argument. An honest candidate fails at most two checks, whereas an incorrect candidate is checked by all three honest servers. Therefore the same acceptance threshold applies using the $k(k-1) = 20$ ordered pairs of distinct servers. The answer map has degree one, so the resulting bit counts are

$$\begin{aligned} U_{\mathrm{MAC}} &= k\ell_Q w + 2k(k-1)\ell_Q W, \\ D_{\mathrm{MAC}} &= k\ell_A(w)w + 2k(k-1)\ell_A(w)W. \end{aligned} \tag{5}$$

To select the authentication field size, we refine the error calculation in Appendix F of the cited work. Since a pair is accepted only if the checks pass for every answer coordinate, a union bound over coordinates is unnecessary. Fix a corrupted candidate i and an honest checker j, and condition on all coins except α_{ij}, including the adversary's random tape. The original independence argument makes the entire claimed answer vector and its authentication polynomials independent of α_{ij} under this conditioning. If the answer vector is incorrect, choose its first incorrect coordinate before revealing α_{ij}. Passing the constant-term check makes the difference between the forged and correct authentication polynomials at that coordinate a nonzero polynomial, of degree at most one. The pair can pass only if this polynomial vanishes at the uniform nonzero point α_{ij}, with probability at most $1/(2^W - 1)$. An accepted incorrect candidate must pass a check with at least one honest server. A union bound over at most $2(5-2) = 6$ corrupted–honest pairs therefore gives

$$\varepsilon_{\mathrm{MAC}} \leq \frac{6}{2^W - 1}.$$

This is a refinement of the published analysis for the same joint verification rule; it does not assume independent errors across record coordinates. Choosing $w = W = 43$ gives $\ell_A(w) = 6{,}097$, 27 padding bits per record, and error at most $6.83 \cdot 10^{-13} < 2^{-40}$. Equation (5) then gives 63,406,080 upload bits and 11,797,695 download bits. The total is therefore 8.96 MiB, as reported in the table.

E Optimality Proofs and Extensions

This appendix supplies the proofs and additional cases for Section 4.5. The general server bound uses the privacy and download assumptions. We also give a sufficient field size for the Shamir-based construction to attain the minimum number of audit queries established in the main text.

E.1 The General Server Bound

We first prove the download requirement for perfect reconstruction, then extend it to statistical error.

Lemma 6 (Reconstruction from a coalition-sized set requires linear download). *Let $\Pi = (\mathsf{Query}, \{\mathsf{Ans}_j\}, \mathsf{Rec})$ be a single-round, perfectly t-private k-server PIR for databases $x \in \mathbb{F}_q^n$. Let $J \subseteq [k]$ with $|J| \le t$ and suppose that, for every index i and every database x, the client recovers x_i from its own randomness together with the correctly computed answers $(a_j)_{j\in J}$. Then $\sum_{j\in J} |a_j| \ge n$ field elements.*

Proof. By perfect t-privacy the distribution of $(q_j)_{j\in J}$ does not depend on the requested index, so its support $\mathcal{V}$ is the same for every i. Fix $v \in \mathcal{V}$ and put $\Phi_v : x \mapsto (\mathsf{Ans}_j(v_j, x))_{j\in J}$. For every i there is client randomness consistent with index i and query vector v, and by hypothesis the client then recovers x_i from its randomness and $\Phi_v(x)$. If $\Phi_v(x) = \Phi_v(x')$ for $x \ne x'$, running the client with the randomness that recovers a coordinate on which they differ produces the same output in both cases, a contradiction; so Φ_v is injective and $\sum_{j\in J} |a_j| \ge n$. □

To state the download condition, let server j return b_j field elements, padding shorter answers to this length. The length b_j depends only on public parameters. The maximum total download from any t servers is

$$D_t = \max_{\substack{J\subseteq[k]\\ |J|\le t}} \sum_{j\in J} b_j.$$

For reconstruction without error, the condition is simply $D_t < n$. Allowing error ε requires the stricter condition $D_t < n f_q(\varepsilon)$, where $f_q(z) = 1 - z - H_2(z)/\log_2 q$ and H_2 is the binary entropy function. As in the base PIR model, query generation uses no database-dependent advice or preprocessing.

Theorem 8 (Lower Bound on the Number of Servers). *Any single-round, perfectly t-private PIR satisfying Definition 1 with error ε and download $D_t < n f_q(\varepsilon)$ must satisfy*

$$k \ge t + d + 1.$$

Proof (Theorem 8). Suppose $k - d \le t$, fix T with $|T| = d$, a legal corrupted set since $d \le t$, and put $J = [k] \setminus T$, so $|J| = k - d \le t$. Let every server in T return $\perp$; then $\mathcal{M} = T$ has size d in every execution, so Proposition 1(ii) gives $\Pr[\mathsf{out} = x_i] \ge 1 - \varepsilon$ for every i and x.

By t-privacy the marginal distribution D of $(q_j)_{j\in J}$ does not depend on i. For $v \in \mathrm{supp} D$ let $p_i(v)$ be the probability that the client outputs x_i, conditioned on $(q_j)_{j\in J} = v$ and the index being i, with $x \leftarrow \mathbb{F}_q^n$ uniform; some $v^\star$ satisfies $\frac{1}{n}\sum_i p_i(v^\star) \ge 1 - \varepsilon$. Fix it and set $A = \Phi_{v^\star}(x)$ with Φ_v as in Lemma 6; each honest answer ranges over $\mathbb{F}_q^{b_j}$, so A takes at most q^{D_t} values, using $|J| \le t$. The client randomness conditioned on $\{(q_j)_{j\in J} = v^\star,\ \text{index} = i\}$ is independent of x, hence of (x_i, A). Including it in the decoder's input therefore does not change $H(x_i \mid A)$, and Fano's inequality gives $H(x_i \mid A) \le H_2(e_i) + e_i \log_2 q$

with $e_i = 1 - p_i(v^\star)$; with $\bar{e} = \frac{1}{n}\sum_i e_i \le \varepsilon$ and concavity of H_2,

$$\begin{aligned} n\log_2 q - D_t \log_2 q \le H(x) - H(A) &\le H(x \mid A) \\ &\le \sum_{i=1}^{n} H(x_i \mid A) \le n\big(H_2(\bar{e}) + \bar{e}\log_2 q\big), \end{aligned}$$

so $D_t \ge n\, f_q(\bar{e}) \ge n\, f_q(\varepsilon)$. For the second inequality, the download hypothesis implies $f_q(\varepsilon) > 0$. Since f_q is convex and $f_q(1) = 0$, it is decreasing on $[0, \varepsilon]$. This contradicts the download hypothesis. Hence $k - d \ge t + 1$. For self-delimiting answers of length at most b_j, replace D_t by $D_t + t$. This follows because a self-delimiting alphabet of length at most b_j has $|\mathcal{A}_j| \le \sum_{\iota \le b_j} q^\iota \le q^{b_j+1}$, which multiplies the count by at most q^t. □

Remark 3 (Scope of the download hypothesis). For the balanced instantiation of Section 6.1, each server returns $m+1$ blocks of ℓ_A symbols, so the hypothesis is

$$D_t = t(m+1)\ell_A < n\, f_q(\varepsilon),$$

where n counts database elements before padding. For $\ell_A = \Theta(\sqrt{n})$ and fixed t, m, it holds for sufficiently large databases and small ε. For a batch, fixing all but one requested index gives a single-item protocol to which the theorem applies, but its full download must be charged: replace $m+1$ by $B+m$.

The proof does not establish a server lower bound when a coalition-sized set can download the entire database. It also relies on robustness under missing responses and therefore does not establish a bound for identification alone when robustness is not required.

E.2 Field Size with $m = d$ Audit Queries

For the Shamir-based construction, Proposition 3 identifies the audit code as a generalized Reed–Solomon code of dimension t. At $k = t + d + 1$, this gives $h = k - t = d + 1$. Taking $m = d$ in Theorem 5 and applying Lemma 3 yields security error at most $2^{-\lambda}$ whenever

$$\log_2 q \ge \lambda + d + 1 + \log_2 k.$$

The field must also provide the k distinct nonzero evaluation points required by the construction. Choosing such a field establishes that $m = d$ suffices for the Shamir-based construction, matching the lower bound in Section 4.5.