Reliable Homomorphic Matching for Fuzzy Labeled PSI at Scale

Erkam Uzun*

***Abstract*—**

Fuzzy Labeled Private Set Intersection (FLPSI) lets a receiver learn the labels of enrolled records that are similar to its query, and nothing else. FLPSI can be built in several ways. Constructions based on a set-threshold reduction reach practical performance: a query matches a record when the two agree on a threshold number of components. These constructions delegate the private matching to an inner set-threshold kernel. We study its homomorphic form, which combines leveled-BFV homomorphic encryption (HE), a garbled circuit, and secret sharing to decide the match under encryption and release the record's label. We identify a composition gap in this kernel, an instance of a protocol-level problem: efficiency is bought with a per-trial false-accept probability, but one query runs a trial for every record, so the error compounds with the database size into the kernel's realization soundness error (RSE), the rate at which it accepts a query the plaintext matcher would reject. The RSE is a reliability property of the cryptographic matching layer, not the matcher's accuracy. On a spurious accept the kernel also returns a value the plaintext matcher would never produce. A sound kernel must contribute zero or negligible RSE of its own. We formalize this requirement as a composable security property, give a closed-form bound on the receiver's advantage, and close the gap with CSTPSI, a kernel that runs independent token rounds and raises the per-trial bound to a matching power. We prove CSTPSI secure in the semi-honest model. The closed-form bound sets the round count: two token rounds suffice for million-scale databases and three for billion-scale at the 10^{-6} engineering threshold. Our evaluation confirms the prediction. At a million records the baseline kernel's RSE reaches 100% while CSTPSI holds it at 0 in every measured configuration. CSTPSI decouples threshold-checking and upload costs from label size. For large labels at small to moderate scale it is more than $20\times$ faster than the baseline kernel, with up to 93% less communication. It converges to the baseline only at the million-scale database size. Our implementation, with a one-command reproducibility harness, is publicly available.

***Index Terms*—private set intersection, fuzzy labeled PSI, garbled circuits, leveled homomorphic encryption, secret sharing**

I. Introduction

Private set intersection (PSI) lets two parties, a sender and a receiver, compute the intersection of their private sets without revealing anything else. A long line of work [1]–[6] has made exact-equality PSI fast enough for wide deployment, for example in contact discovery and compromised-credential checking. Many applications, however, need similarity instead of exact equality: biometric templates never repeat exactly across captures, and a password is risky even when it is only close to a leaked one. *Fuzzy Labeled PSI* (FLPSI) serves this regime. The sender holds a database of enrolled records, each carrying a label; the receiver holds a query and learns the labels of the records that are similar to it, and nothing else, while the sender learns nothing at all.

This paper is about a gap that opens when the matching inside FLPSI is made fast enough for practice. Secure realizations of a similarity matcher rarely evaluate the closeness predicate exactly; trading exactness for efficiency at scale is the norm across secure similarity search, in PSI and beyond [7]–[9]. In FLPSI the trade takes a specific form: the predicate is replaced by a cryptographic test that is exact on a true match but wrongly accepts an unrelated pair with a small per-trial probability. The bound looks harmless until one notices that a single online query is not one test but many, run across the whole database. The per-trial probability then compounds into a per-query rate that grows with the database, and at deployment scale the kernel accepts a query that matches no record. This is a soundness defect of the realization: it accepts inputs the plaintext matcher would reject, and we call the rate at which it does so the *realization soundness error* (RSE). The RSE is a reliability property of the realization, not an accuracy metric, so it cannot be weighed against the matcher's intrinsic false-match rate (FMR), a plaintext floor that any realization inherits and that is out of scope here. The problem is protocol-level: it appears in any realization that makes this trade. We give the model and the attack in §II.

We study this RSE in the setting where it can be isolated and measured: the set-threshold reduction [7], [8], one of the several ways to build FLPSI. It encodes each record and each query into a tuple of N component items so that similar inputs agree on many components, declares a match when at least k of the N positions agree, and runs the private matching in a discrete inner kernel. That last property is what matters here: with the matching confined to a kernel, the kernel's RSE separates from the matcher's FMR. Within this reduction, our focus is the kernel's homomorphic form, which we call a *leveled-HE-based set-threshold kernel*: a composition of leveled-BFV homomorphic encryption (HE), a garbled-circuit (GC) oblivious pseudorandom function (OPRF), and Shamir secret-share reconstruction.[1] We choose this form because one of the most practical FLPSI constructions is built on it

*This work extends research initiated during the author's doctoral studies at the Georgia Institute of Technology. All contributions are original and were developed independently using the author's personal time and resources, without affiliation to any current or former employer.

[1]The formal sections (§VI onward) treat the kernel as a two-party protocol and keep the standard phrasing "protocol Π_T".

and shows the gap most sharply. Our reference example is STLPSI, the inner sub-protocol of the FLPSI'21 protocol [7]; we refer to it as the baseline kernel, or the baseline protocol, in the rest of the paper. One caveat guides the whole study. Secure-computation remedies are tied to the schema of the construction they harden, so a single fix cannot cover every realization and should not try. We therefore formalize the gap once, in the set-threshold setting, and close it on this representative kernel with a new construction, CSTPSI. The result is meant as a reusable template rather than a one-off patch.

The trial count is concrete in this kernel. For each query it checks every k-subset of the N component positions in each of the n_{part} database partitions, $\binom{N}{k} n_{part}$ independent trials in all, with n_{part} growing linearly in the database size D. Each trial is exact on a true match but wrongly accepts an unrelated pair with probability $1/F$, a coincidental collision in the Shamir field of size F. The per-query RSE aggregates these trials, so it rises with D toward one.

What the kernel returns on a spurious accept confirms the defect is one of soundness. The released value is *off-curve*. It is a field element outside the label range $[0, D)$, independent of every record, so it is not a stored label and the accept leaks nothing (§IV-B explains why). The same kernel also runs without labels, where it returns only a match bit. There a spurious accept is simply a wrong answer, even under a perfect matcher, with no off-curve value to mark it. Either way a sound kernel must have zero or negligible RSE of its own. Prior constructions either inherit this gap silently or re-tune parameters around it per deployment; none gives a composable treatment, which we survey in §III.

We give that composable treatment in the two-party semi-honest model, with five contributions:

- **A composition soundness gap, identified and modeled.** We model the set-threshold kernel's per-trial spurious accept as a deployment-scale soundness defect, and cast it as a concrete attack by a semi-honest receiver (§II).
- **A composable RSE definition and bound.** We define security for the set-threshold setting so that the kernel's added probability of accepting an unmatched query is the quantity of interest. The definition yields a closed-form bound (Lemma 1) that makes the workload composition explicit, and it recovers the per-element analysis of exact-match labeled PSI as a special case.
- **CSTPSI: a composable validator construction, proved secure.** CSTPSI runs T independent token rounds, each an independent validator of a candidate match, so each trial's success probability falls from $1/F$ to $1/F^T$ and the per-query bound becomes $\binom{N}{k} n_{part}/F^T$ (Theorem 2). We prove it secure in the semi-honest model through a simulation-based reduction (§VIII).
- **Protocol engineering achieves soundness restoration for free.** Amplification adds token rounds, and a naive realization would pay for each with a fresh garbled circuit and encrypted-query upload, so cost would grow with the round count and the label size. CSTPSI instead introduces several improvements without changing the ideal functionality or the simulation proof. The receiver (i) evaluates the garbled circuit once per session, (ii) sends the query once, and (iii) caches the modular inverses that reconstruction reuses, while the sender caches the expanded encrypted query-power window. These optimizations reduce the per-round work to plaintext-ciphertext inner products and decouple the threshold-check and upload costs from the label size (§VII).
- **Implementation and artifact.** A complete implementation with a one-command harness that regenerates every table and figure in this paper, publicly available.[2]

Our measurements confirm both the gap and the fix. We first use synthetic data that realizes a perfect matcher with zero FMR, so every accept is purely the kernel's RSE and its scaling with the database is clear: at the FLPSI'21 parameters the baseline kernel's RSE climbs from under 1% at 1K and 4.5% at 10K records to 56% at 100K and 100% at 1M, while CSTPSI at $T = 2$ admits no spurious accept in any measured configuration through 1M records, below the 10^{-6} ($\approx 2^{-20}$) engineering threshold; the bound shows $T = 3$ suffices at billion scale. We then run the attack on two real datasets of different scales, LFW and Deep1B, where the baseline's RSE climbs to near one at a million records (0.999 on Deep1B) while CSTPSI holds it at zero (§IX-D). The added security costs little. With its cached garbled circuit and encrypted query, CSTPSI matches or outperforms the baseline across the tested range. It is more than $20\times$ faster for large labels on small-to-moderate databases, converges to within about 2% of the baseline at a million records, and sends up to 93% less data in the regimes where it leads.

II. Threat Model and Assumptions

We state here the system this paper studies and the powers we grant the adversary. We then draw the line the rest of the paper rests on: between the matcher's own error floor, which the protocol cannot and should not change, and the error the realization makes, which the protocol can and must close. Finally, we say what counts as an attack.

A. System and Adversary Model

The system is an FLPSI deployment built on the set-threshold reduction. A *sender* holds a database of enrolled records, each carrying a label. A *receiver* holds a query and wishes to learn the labels of the records that are similar to it, and nothing else. The matching is delegated to an inner *leveled-HE-based set-threshold kernel*, whose job is to realize the plaintext k-of-N matcher under encryption: a query matches a record when the two agree on at least k of their N components, and on a match the kernel releases the record's label. The receiver submits one query per online interaction. In scope is what the kernel does on each submitted query: which queries it accepts and which values it returns. Out of

[2] https://github.com/euzun/cstpsi

scope is the choice of representation that turns raw inputs into the N components.

We adopt the standard semi-honest two-party model.

- Both parties are probabilistic polynomial-time *semi-honest* adversaries with security parameter λ: each follows the protocol specification but may attempt to learn additional information from its view of the transcript.
- We analyse a single execution in isolation; guarantees under multiple concurrent executions are out of scope.
- We rely on the standard assumptions of the building blocks introduced in §VII: the IND-CPA security of BFV leveled homomorphic encryption [10], the pseudorandomness of AES-128 [11], [12], and the information-theoretic threshold property of Shamir secret sharing [13].
- The sender draws a fresh OPRF key per execution; the receiver generates the HE keys and shares only the public key with the sender. No trusted third party is involved.

The model makes no assumptions about the origin or domain of database items. Rows may derive from biometric templates, attribute arrays, medical records, or any structured input amenable to a k-of-N component decomposition.

B. Two Error Sources

This paper separates two error sources with different owners, both introduced in §I. The *false-match rate* (FMR) belongs to the matcher. It is a plaintext floor that we inherit and leave out of scope. The *realization soundness error* (RSE) belongs to the kernel. It is the rate at which the kernel accepts a query the matcher would reject, and it is the quantity we target. The two are not interchangeable.

Why the RSE matters is easy to undervalue, so we state it plainly. The kernel's one job is to realize the matcher faithfully, so that its accept-or-reject decision can be trusted as the matcher's own. A spurious accept breaks that trust. It cannot be weighed against the FMR, because a large FMR is the matcher honestly reporting chance-similar records, while a spurious accept is the kernel reporting a match that was never there. Filtering the values afterward does not help, because a fixed fraction of them lands in the label range by chance and passes a range check silently. The stakes are starkest without labels. CSTPSI also runs as unlabeled fuzzy PSI, where the kernel returns only a match bit, so a spurious accept is a wrong answer outright, even under a perfect matcher, with nothing to inspect or filter. A deployable kernel must therefore contribute zero or negligible RSE of its own. The attack below shows why the requirement is strict, because at scale the untreated error swamps the match decision.

C. The Composition Attack on FLPSI

We now turn this soundness gap into a concrete attack: a semi-honest receiver, issuing only ordinary queries, drives the kernel's realization soundness error from negligible to near-certain as the database grows. We give first the adversary and procedure, and then what a spurious accept yields.

a) Assets and adversary.: The asset is the soundness of the match decision: every accept should mark a record the plaintext matcher would also accept, and every released value should be a genuine label of such a record. The adversary is a semi-honest receiver that submits queries through the normal interface. It need not deviate from the protocol, and it need not hold any input similar to an enrolled record. It only has to issue ordinary queries and read what the kernel returns.

b) The procedure.: The attack is nothing but the trial structure of §I carried to scale. The receiver issues ordinary queries; each drives $\binom{N}{k} n_{part}$ independent trials, and aggregated over a database of size D the per-query realization soundness error climbs toward one. This is a multiple-comparisons effect: a per-trial rate harmless in isolation turns near-certain across the simultaneous tests of one query. After passing a dtabase size, the kernel accepts essentially every query, even with no similarity to any record.

c) Why this is a protocol-level problem.: The compounding follows from the workload, not from any flaw unique to one scheme. Any realization that buys efficiency with a per-trial spurious accept, and that runs a trial for every record, carries the same gap: the per-query rate grows with the database and the match decision eventually fails. The set-threshold kernel is where the gap can be isolated and measured, since the RSE there separates cleanly from the matcher's FMR, but the shape of the problem is general. We formalize the per-query rate as a closed-form bound (Lemma 1) and measure the break on real data, where the baseline kernel's RSE reaches near one at a million records while our construction holds it at zero (§IX-D). The rest of the paper closes this gap inside the kernel and sizes the fix to the database scale.

III. Related Work

This section surveys prior work on private set intersection (PSI) along the two axes most relevant to our construction, *fuzziness* of the matching predicate and *labeling* of the sender's records, and positions CSTPSI relative to STLPSI, the closest prior threshold-labeled construction and our reference baseline. Exact-equality PSI [1]–[6] is deployed at consumer scale but does not apply directly to similarity matching, so we leave it aside and focus on the fuzzy line.

The fuzzy PSI (FPSI) literature splits along two reductions of the closeness predicate. The first is the set-threshold (k-of-N) reduction of our setting [7], [8], [19]–[21]. The second computes the predicate directly as a metric inequality $d_p(x, y) \leq \tau$ for an ℓ_p distance ($p \in \{1, 2, \infty\}$) or a Hamming threshold [18], [22]–[32]. Garimella et al. (CRYPTO'22) [27] suppress the realization soundness error (RSE) at the batched-function-secret-sharing level by repeating ℓ independent bFSS sharings (ℓ=280 as instantiated). Gao et al. [28] achieve strictly linear communication via a "fuzzy mapping" assumption, with a reported 30–305$\times$ speedup over the prior EUROCRYPT 2024 baseline [18]. Yang et al. [31] push ℓ_p-distance FPSI into the symmetric-key regime (a reported 12–145$\times$ improvement), and report a false-positive leakage in a recent linear-complexity AHE-based construction [33]. Meng

TABLE I: Audit-thread positioning. "Thr.": k-of-N matching; "Lbl.": label retrieval on match; "RSE": analyzes spurious-accept scaling for its threshold or label primitive.

Protocol	Venue / Year	Mechanism	Thr.	Lbl.	RSE	Role in audit thread
Resende and Aranha [14]	FC'18	cuckoo-filter tags	no	no	per-element only	flagged-by-successors
Kales et al. [15]	USENIX'19	cuckoo-filter tags	no	no	per-element only	deployed-with-bound
Cong et al. [16]	CCS'21	HE polynomial eval	no	yes	per-execution	recomposes deployed bound
Garimella et al. [17]	CRYPTO'21	OKVS amplification	no	no	data-structure level	algebraic precedent ($p \mapsto p^c$)
Uzun et al. [7]	USENIX'21	HE + GC, single token	yes	yes	no	no scale RSE analysis (baseline we extend)
Chakraborti et al. [8]	USENIX'23	subsample reconciliation	yes	no	no	no scale RSE analysis
van Baarsen and Pu [18]	EUROCRYPT'24	DDH spatial hashing	no	no	no	flags leakage in prior distance-aware FPSI
CSTPSI (this work)	-	HE + GC, multi-token	yes	yes	yes (Lem. 1, Thm 2)	general formalization + constructive fix

et al. [32] target unbalanced biometric search under L_∞ with sub-linear communication, and further works push structure-aware and distance-aware-OT directions [29], [30]. None of these addresses the fuzzy labeled PSI (FLPSI) setting we work in, where the receiver learns the labels of similar records, though all attack the same fuzzy-intersection problem under different efficiency and metric profiles.

Several primitives sit near CSTPSI's functionality but solve a different problem. *Cardinality-threshold PSI* [34], [35] returns the exact-equality intersection only when its cardinality exceeds a threshold t and otherwise returns nothing; the threshold there is over $|X \cap Y|$ across the entire sets, not over component agreement within a single fuzzy item. *Labeled PSI from HE* (APSI [16], [36]) returns labels for exact-match items; the per-element $1/F$ false-positive analysis at the heart of its cuckoo/bin design is the lineage from which the bound of Lemma 1 generalizes. *Searchable encryption*, *private information retrieval*, and *circuit-PSI* [37], [38] share parts of the toolkit but address orthogonal functionalities: encrypted-index query, single-row recovery, and arbitrary-circuit set evaluation, respectively.

The thread that motivates this work cuts across all of these lines, and Table I collects it. The works in the table target different problem settings, so the comparison is qualitative positioning rather than a performance ranking. The common pattern is a per-trial error that a single query runs against every record, so its small per-trial probability compounds with the database size (§I). Several constructions do not analyze this scaling: the unbalanced PSI of Resende and Aranha, built on cuckoo-filter tags [14], the contact-discovery protocols of Kales et al. [15], the distance-aware PSI of Chakraborti et al. [8], and the k-out-of-N labeled fuzzy PSI of Uzun et al. [7], the baseline kernel we extend. Others control the error per deployment or per execution: the APSI line sizes its hash range and cuckoo bounds against the workload [16], [36], and modern OKVS-PSI absorbs it into the field size [6]. A third group, including van Baarsen and Pu [18] and Yang et al. [31], frames the error as information disclosure and flags false-positive leakage in prior fuzzy-PSI constructions. Closest to a fix, Garimella et al. (CRYPTO'21) [17] amplify an OKVS overfitting probability from p to p^c inside the encoding structure, and the ℓ-fold bFSS repetition of Garimella et al. (CRYPTO'22) [27] above is the same move at the function-secret-sharing layer. Both strengthen one trial inside a sub-primitive. This is the wrong layer for the soundness gap: neither composes the guarantee across the many trials a query runs against a full database, and neither exposes the amplification as a tunable protocol parameter with a matching security definition. CSTPSI supplies exactly these two missing pieces, which makes it, to our reading, the first composable treatment of the gap, demonstrated in the threshold-labeled setting.

IV. Preliminaries

This section introduces the protocol class we analyse, formalizes the realization soundness error (RSE) composition problem that any protocol in this class faces at deployment scale, and rules out the obvious parameter-level remedies. Our solution, CSTPSI, is introduced in §V and formalized in §VI.

A. Set-Threshold Fuzzy PSI (k-of-N matching)

We work over a protocol class commonly called *Set-Threshold fuzzy PSI* or *k-of-N matching*: each row of the sender's database and each receiver query is an N-tuple of items from a fixed domain $\mathcal{D}$, and a match is declared when at least k of N item positions agree under the closeness predicate Cl_τ [7], [8]. Table II collects the symbols used throughout the rest of the paper. As a representative member of this class we take STLPSI, the labeled fuzzy PSI primitive of Uzun et al. [7], as the baseline protocol since it carries the RSE composition problem we analyse in this section in its sharpest form.

Attaching a label to each row turns this matching primitive into *Fuzzy Labeled PSI* (FLPSI, §I), the functionality this paper studies: the receiver learns the labels of the rows it matches, and nothing else. The sender inputs a database $Db = \{(x_e, \ell_{x_e})\}_{e=1}^{D}$ in which each row x_e is an N-tuple of items and each row carries an opaque label $\ell_{x_e} \in \mathbb{F}_F$. The receiver inputs a single fuzzy query $Y = (y_1, \ldots, y_N)$ from $\mathcal{D}^N$. The functionality returns to the receiver the label ℓ_{x_e} of every row e that agrees with the query on at least k of N item positions, and $\perp$ to the sender. Labels are *opaque* from STLPSI's perspective: the primitive treats them as arbitrary field elements, so any structural decoration the labels need (a validation prefix, multiple polynomials for cross-validation, additional verification rounds) must be assembled by the caller before invoking the primitive and stripped after the output.

Each row's N items are blinded under an OPRF ($b_{e,i}$ on the sender side, b_j on the receiver side); per item position i, the sender packs at most s_{part} rows into a single interpolating

TABLE II: Notation and parameters used throughout the paper. Values are fixed across experiments; ranges are swept in §IX.

Symbol	Meaning (fixed value, where applicable)
N	Component items per database row (64)
k	Matching threshold of agreeing items (2)
T	Independent token rounds $(1, 2, 3)$
K	Label-chunk rounds for 23-bit and 16/32/64-byte labels $(1, 6, 12, 23)$
D	Database size in rows (1K–1M)
s_{part}	Maximum rows per partition (32)
n_{part}	Number of partitions, $\lceil D/s_{part} \rceil$
F	BFV plaintext modulus and Shamir field $\mathbb{F}_F$ (8519681, $\sim$23-bit)
m	BFV polynomial-modulus degree (4096)
$\mathcal{D}, \mathcal{L}$	Item domain and label space
X	Sender's items $\{x_1, \ldots, x_D\}$, $x_e \in \mathcal{D}$
ℓ_{x_e}	Label of sender item x_e (in $\mathcal{L}$)
Y	Receiver's query $(y_1, \ldots, y_N) \in \mathcal{D}^N$
Cl_τ	Closeness predicate
κ	Sender's AES-OPRF key
b_j	Blinded query item: $\mathrm{AES}_\kappa(y_j) \bmod F$
$P_{t,e}(x)$	Per-row Shamir polynomial in round t
$f_{t,i,p}(x)$	Per-partition interpolating polynomial
KR	Shamir secret reconstruction (see §VII-A)

polynomial $f_{t,i,p}$ of degree $s_{part} - 1$ over $\mathbb{F}_F$, evaluated at the receiver's blinded query inputs in a single homomorphic round; the receiver recovers each polynomial value and tries every $\binom{N}{k}$ unordered k-subset of item positions per partition via k-point Lagrange reconstruction at $x = 0$[3]. Each token round uses a fresh degree-$(k-1)$ Shamir polynomial $P_{t,e}$ per row whose constant term is the public token 0; reconstructions that hit 0 trigger a candidate match. The full construction with all design rationale is in §VII.

B. The RSE Composition Problem

To our knowledge, FLPSI protocols evaluated to date have not analyzed RSE scaling at deployment scale (§III). We observe that the per-query RSE of this class scales linearly with the number of database partitions $n_{part} = \lceil D/s_{part} \rceil$, and therefore degrades rapidly as D grows into the database scales that motivate fuzzy PSI in the first place: biometric watchlist matching at the hundred-thousand to ten-million scale, and credential-leak monitoring well above that scale.

The per-query RSE is a soundness defect, not just an accuracy nuisance. On a non-matching pair the kernel reconstructs an *off-curve* value, uniform in $\mathbb{F}_F$ and independent of any row's payload, so it is not a stored label but a value the plaintext matcher would never return. This is distinct from stored-label exposure, which happens only on a genuine $\geq k$ match, the matcher's intrinsic false-match rate that lies outside our scope. Lemma 1 formalizes the scaling and bounds the probability of such a wrong accept; its proof shows the reconstructed value is uniform.

[3]Practical STLPSI deployments instantiate the threshold at $k = 2$, yielding a 2-point Lagrange operation; the bounds in this section apply at general k.

Lemma 1 (Composition Bound on the RSE)**.** *Let Π_T be a labeled fuzzy PSI protocol with $T \geq 1$ independent Shamir-based token rounds and parameters (N, k, s_{part}, F). For a database of D entries with $n_{part} = \lceil D/s_{part} \rceil$, and a random query Y sharing fewer than k items with every enrolled entry, the probability that Π_T wrongly accepts and returns any value is at most*

$$\Pr[\text{accept}] \;\leq\; \binom{N}{k} n_{part} \,/\, F^T + T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B}), \quad (1)$$

where $\mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B})$ is the AES pseudorandomness advantage of an efficient adversary $\mathcal{B}$; under standard assumptions this term is bounded by 2^{-128} up to negligible constants. At $T = 1$, the dominant term is tight to first order when $\binom{N}{k} n_{part}/F \ll 1$: $\binom{N}{k} n_{part}/F = 1-(1-1/F)^{\binom{N}{k} n_{part}} + O\big((\binom{N}{k} n_{part}/F)^2\big)$. Since $\binom{N}{k} n_{part}$ grows with D, this ratio is not small at deployment scale, where the linear form overestimates and the exact $1-(1-1/F)^{\binom{N}{k} n_{part}}$ should be used (§IX).

Proof sketch. Replace AES_κ with a random function R at the cost of $T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B})$ by a standard PRF/PRP hybrid argument across the T token rounds. Conditioned on the R-output, fix a partition p containing no entry with $\geq k$ matches against Y, and fix a k-subset $I = \{i_1, \ldots, i_k\}$ of item positions. In round t, the sender's per-entry Shamir polynomial $P_{t,e}$ is sampled with fresh random coefficients, so the partition polynomial $f_{t,i,p}$ interpolates the partition through s_{part} ordinate values $\{P_{t,e}(i)\}_{e \in p}$ that are independent uniform draws over $\mathbb{F}_F$ (each is a fixed nonzero linear combination of the row's fresh secret-zero Shamir coefficients, which reduces to a nonzero scalar multiple of the single fresh coefficient when $k = 2$). Evaluated at any point b outside the s_{part} interpolation x-coordinates, $f_{t,i,p}(b)$ is a fixed nonzero linear combination of those independent uniform values, hence itself uniform in $\mathbb{F}_F$. By hypothesis no entry in p shares $\geq k$ items with Y, so for every entry $e \in p$ at least one position $i_j \in I$ is a non-node ($b_{i_j} \neq b_{e,i_j}$); the k values fed to the reconstruction therefore include at least one such uniform value. The modular-bias and zero-collision terms (the $0 \mapsto 1$ remap induces a per-value bias of $2/F$ on the value 1; the $2^{128} \to F$ reduction induces a $O(F/2^{128}) \approx 10^{-32}$ bias) are absorbed into the asymptotic. The k-point Lagrange reconstruction at $x = 0$ is an affine combination of these k values that gives nonzero weight to a uniform non-node value, and is therefore uniform in $\mathbb{F}_F$; equality with the public token 0 occurs with probability $1/F$. Across T rounds with bin-local fresh coefficient samples, the per-round events are independent, giving $(1/F)^T$. This per-round independence rests on the Shamir coefficients being sampled independently across rounds, an assumption on the randomness source that Theorem 2 makes explicit. A union bound over the $\binom{N}{k}$ candidate k-subsets in each of n_{part} partitions yields (1); the union bound itself requires only sub-additivity, not independence across bins. □ □

Lemma 1 generalizes the per-element false-positive analysis of prior exact-matching PSI work, e.g. APSI [36], which estab-

lishes $1/F$ per query item per bin: setting $N = k = n_{part} = 1$ recovers their bound from ours. The novelty here is lifting that single-element analysis to the threshold setting, where the combinatorial factor $\binom{N}{k}$ and the bin amplification n_{part} jointly drive RSE linearly in D. Our measurements over a reference codebase confirm this qualitative shape; the full configuration table and the comparison against the exact bound appear in §IX.

C. Why Existing Remedies Fall Short

Two parameter-level remedies suggest themselves before a structural change: (i) growing the partition size s_{part} to shrink the count $n_{part} = \lceil D/s_{part} \rceil$, and (ii) enlarging the plaintext modulus F to shrink the per-trial $1/F$ probability. Both target Lemma 1 directly, but neither is viable at practical labeled fuzzy PSI parameters.

a) Growing the partition size s_{part}.: Doubling s_{part} halves n_{part} and the bound, but the per-partition interpolating polynomial has degree $s_{part} - 1$, so communication and homomorphic-evaluation cost grow linearly in s_{part} as well. The trade-off is linear-for-linear, and a cryptographically negligible bound forces s_{part} to a value where cross-partition amortization collapses.

b) Enlarging the plaintext modulus F.: In BFV-style encryption the plaintext modulus must be prime and satisfy $F \equiv 1 \pmod{2\,m}$ for SIMD (single-instruction multiple-data) batch encoding, where m is the polynomial-modulus degree. At the m used in practical labeled-PSI deployments the prime budget at 128-bit security is exhausted near a fixed bit-width; enlarging F further forces m to the next power of two, roughly doubling per-operation evaluation cost and ciphertext size at the same depth. This more than offsets the improvement to the bound at every practical label size (concrete costs in §IX).

The structural fix therefore lies elsewhere: re-folding the caller/primitive boundary so the amplification round count is a parameter of the primitive, not a re-orchestration of its caller. We introduce this fix as CSTPSI in §V.

V. Our Solution: CSTPSI

We introduce *Composable Set-Threshold PSI* (CSTPSI), the primitive that takes the realization soundness error (RSE) mitigation inside its boundary by absorbing the caller-managed share-construction preamble into the protocol itself. Five design choices drive the construction: where share construction lives, how many token rounds the primitive admits, how the garbled-circuit OPRF interacts with the round count, how the receiver's encrypted query material is amortized across rounds, and how the receiver reuses its reconstruction work across rounds.

a) Taking share construction into the protocol.: Set-Threshold fuzzy PSI primitives that treat labels as opaque field elements delegate share construction to the caller, which makes any structural change to the share machinery (adding verification rounds for amplification, cross-validation encodings, longer-label transport) a caller-side surgery rather than a parameter change. CSTPSI accepts raw labels at its interface and constructs Shamir polynomials [13] internally, parameterized by a configurable round count T. Variations on the share machinery then become parameter choices.

b) The role of T (multi-token amplification).: CSTPSI exposes T, the number of independent token rounds, as a first-class protocol parameter. Each added token round acts as an independent *validator* of a candidate match: the round re-checks the same candidate with fresh Shamir coefficients, so a spurious accept must fool all T validators at once. Fresh coefficients per round drive the per-trial spurious-accept probability geometrically to $1/F^T$. The validator lives at the protocol-round level, so it is composable across the whole query workload, and we *size* the number of validators to the database scale through the closed-form bound (Lemma 1). Corollary 1 gives the sufficient T for any target scale, and $T = 2$ already makes the per-query bound engineering-negligible at million-scale D (§VIII-B). The full amplification statement is Theorem 2.

c) Once-per-session GC-OPRF.: The OPRF used to blind query and database items is realized as a two-party garbled-circuit AES evaluation, which mutually conceals the query items from the sender and the key from the receiver. A naive realization runs this circuit per round, inflating the receiver's online cost by a factor $T + K$ and making the GC step the dominant component of online latency at any reasonable K. CSTPSI executes the garbled circuit [11] only *once per online session*, since the blinded queries $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ are deterministic in the query items and the sender's key, hence independent of the round. The general principle of minimizing protocol-conversion steps in a mixed arithmetic–boolean computation traces to ABY [12].

d) Once-per-session encrypted query powers with sender caching.: The sender's per-partition polynomial evaluation $f_{t,i,p}(b_j) = \sum_{\ell=0}^{d-1} c_{t,i,p,\ell} \cdot \mathrm{Enc}(b_j^\ell)$, with $d = s_{part}$, requires ciphertexts $\mathrm{Enc}(b_j^\ell)$ for every $\ell \in [0, d)$. Following the baby-step–giant-step (BSGS) windowing scheme used in APSI-style labeled PSI [7], [16], [36], the receiver transmits only a sparse window of $O(\sqrt{d})$ encrypted powers; the sender expands this window to the full power set via $O(\sqrt{d})$ ciphertext-ciphertext multiplications at session setup, consuming a small constant number of BFV multiplicative levels. CSTPSI then *caches* the expanded power bundle on the sender side for the duration of the online session, so the per-round sender work reduces to plaintext-ciphertext (coefficients $\times$ power) inner products with no per-round ciphertext-ciphertext multiplications. The expansion cost is paid once and amortized over all $T + K$ rounds. [4]

e) Cached reconstruction on the receiver.: At reconstruction the receiver runs a two-point Lagrange combination at $x = 0$ for each candidate pair $\{i, j\}$ in a partition. Its weights depend only on the blinded query coordinates b_i, b_j, which the single garbled circuit fixes for the whole session, so they

[4]The trade-off between BSGS and a full-receiver-side pre-encryption variants depends on hardware and deployment regime (network bandwidth vs computation); Appendix A describes both variants.

are identical across all $T + K$ rounds. CSTPSI computes the modular inverses in these weights once per pair and reuses them across every round, replacing a per-round field inversion with a cached lookup.

CSTPSI inherits the semi-honest two-party model and building-block assumptions of §II. Within that model the T token rounds also blunt a receiver who probes with crafted queries to force a spurious accept, since each candidate must then collide in all T rounds rather than one (probability $1/F^T$, not $1/F$); a sender that omits entries merely substitutes inputs and is not a separate attack.

VI. Formal Definition of CSTPSI

We formally define the CSTPSI primitive: its syntax, the ideal functionality $\mathcal{F}_{\text{CSTPSI}}$ it realizes, and its correctness clause. We start from a closeness abstraction (Definition 1) that fixes the item-level fuzzy-matching relation, then give the protocol syntax (Definition 2) over it [7], [39].

Definition 1 (Closeness Domain)**.** *A* closeness domain *for set-threshold parameters (N, k) with $1 \le k \le N$ is a pair $\Lambda = (\mathcal{D}, \mathsf{Cl}_\tau)$, where $\mathcal{D}$ is an item set and $\mathsf{Cl}_\tau \colon \mathcal{D}^N \times \mathcal{D}^N \to \{0,1\}$ is the symmetric set-threshold closeness function defined by*

$$\mathsf{Cl}_\tau(Y, X) = 1 \iff \left|\{\, j \in [N] : y_j = x_j \,\}\right| \ge k.$$

We read $\mathsf{Cl}_\tau(Y, X) = 1$ as "Y is close to X": the two N-tuples agree on at least k of the N item positions.

Definition 2 (Composable Set-Threshold PSI – CSTPSI)**.** *A* CSTPSI *protocol Π_T is a two-party interactive protocol between a sender S and a receiver R, parameterized by a closeness domain $\Lambda = (\mathcal{D}, \mathsf{Cl}_\tau)$ per Def. 1, a label space $\mathcal{L}$, the set-threshold parameters (N, k) with $1 \le k \le N$, the token-round count $T \ge 1$, and a security parameter λ.*

Inputs. S holds a labeled database $Db = \{(x_e, \ell_{x_e})\}_{e=1}^{D}$ with each $x_e \in \mathcal{D}^N$ an N-tuple of component items and each $\ell_{x_e} \in \mathcal{L}$ the row's payload label. R holds a query $Y \in \mathcal{D}^N$. Both parties agree on $(\Lambda, \mathcal{L}, N, k, T, \lambda)$ and on any public parameters the concrete realization requires.

Outputs. R outputs the multiset $\mathcal{R} = \{\, \ell_{x_e} : \mathsf{Cl}_\tau(Y, x_e) = 1 \,\}$; S outputs $\perp$.

Ideal functionality. The CSTPSI protocol Π_T realizes $\mathcal{F}_{\text{CSTPSI}}$ semi-honestly with the simulator gap quantified in Theorem 1. The token-round count T is a first-class protocol parameter, not a tuning knob hidden inside the realization. The protocols $\Pi_1, \Pi_2, \ldots$ all realize this same functionality, but with *different* simulator gaps; an operator picks T to meet a target gap at a target deployment scale.

Correctness. The correctness clause is deterministic with respect to true matches. We deliberately do not adopt an ε-correctness clause that would allow the protocol to be wrong on spurious accepts with negligible probability [7]. An ε-correctness clause creates a loophole: it discharges the obligation to account for spurious accepts in either the correctness analysis (where they would surface as deviations from $\mathcal{F}_{\text{CSTPSI}}$) or the security analysis (where they would surface as simulator-gap mass). We instead treat the spurious-accept event as a *soundness* event: a quantified deviation from $\mathcal{F}_{\text{CSTPSI}}$'s authoritative output, in which the real protocol accepts a query the ideal functionality would reject and returns a value the ideal functionality would never produce. Lemma 1 and Theorem 2 bound this event, and it appears as a soundness contribution to the simulator gap of Theorem 1. Accounting for it this way closes the loophole and composes cleanly with downstream applications that treat $\mathcal{F}_{\text{CSTPSI}}$'s output as authoritative.

Functionality 1 (Ideal functionality $\mathcal{F}_{\text{CSTPSI}}$)**.** *Parameters. Structural parameters (N, k) with $1 \le k \le N$; closeness setting $(\mathcal{D}, \mathsf{Cl}_\tau)$; label space $\mathcal{L}$.*
Inputs. From S, a labeled database $Db = \{(x_e, \ell_{x_e})\}_{e=1}^{D}$ with $x_e \in \mathcal{D}^N$ and $\ell_{x_e} \in \mathcal{L}$; from R, a query $Y = (y_1, \ldots, y_N) \in \mathcal{D}^N$.
Computation. For each entry e, mark it a match iff $\mathsf{Cl}_\tau(Y, x_e) = 1$.
Outputs. To R, the multiset $\mathcal{R} = \{\, \ell_{x_e} : \mathsf{Cl}_\tau(Y, x_e) = 1 \,\}$; to S, $\perp$.
Leakage. None beyond what $\mathcal{R}$ and $|Y|, |Db|$ reveal: no information on non-matching entries and no per-component side channel.

Definition 3 (CSTPSI correctness)**.** *A CSTPSI protocol Π_T is* correct *if, for every database Db and every query Y such that there exists an enrolled entry e with $\mathsf{Cl}_\tau(Y, x_e) = 1$, the output multiset $\mathcal{R}_{\Pi_T}$ computed by the honest execution of Π_T contains ℓ_{x_e} with probability 1. Equivalently, the kernel false-reject rate of Π_T, conditioned on a true k-of-N component agreement, is zero (the representation's recall miss reported in §IX-D is a plaintext property of the sketch, identical for the matcher, STLPSI, and CSTPSI).*

The CSTPSI security definition and the proof that Π_T realizes $\mathcal{F}_{\text{CSTPSI}}$ are presented in §VIII and Appendix B.

VII. Building Blocks of CSTPSI

We instantiate CSTPSI from three standard primitives: Shamir secret sharing, the BFV leveled homomorphic encryption (HE) scheme, and a two-party AES-128 garbled-circuit OPRF. We then assemble CSTPSI from these primitives in §VII-D as a self-contained protocol Π_T, and state the full protocol in §VII-D. Notations follow Table II.

A. Shamir Secret Sharing

A k-out-of-N Shamir secret sharing scheme [13] over a finite field $\mathbb{F}_F$ is a pair of algorithms $(\mathsf{KS}, \mathsf{KR})$.[5] KS takes a secret $s \in \mathbb{F}_F$, samples a random degree-$(k-1)$ polynomial $P(x) = s + a_1 x + \cdots + a_{k-1} x^{k-1}$ with $a_i \xleftarrow{\$} \mathbb{F}_F$, and

[5] We fix $k = 2$. Raising k inflates the per-pair work and the RSE bound $\binom{N}{k} n_{part}/F^T$ through $\binom{N}{k}$ (2016 at k=2, 41664 at k=3) but leaves the per-trial accept probability at $1/F$; only T lowers that, to $1/F^T$ (Theorem 2).

Inputs. *Sender* S inputs $Db = \{(x_e, \ell_{x_e})\}_{e=1}^{D}$, where each $x_e \in \mathcal{D}^N$ is an N-tuple of items and each $\ell_{x_e} \in \mathcal{L}$. *Receiver* R inputs $Y = (y_1, \ldots, y_N)$ with $y_j \in \mathcal{D}$. Both parties agree on the parameters (N, k, T, F, s_{part}) and on BFV public parameters.

1) **[Init]** S samples an AES-128 OPRF key $\kappa \xleftarrow{\$} \{0,1\}^{128}$.
2) **[S Offline: partition]** S partitions Db into $n_{part} = \lceil D/s_{part} \rceil$ partitions of at most s_{part} rows each, rejecting within-partition duplicate items so step 5's interpolation is well-defined.
3) **[S Offline: blind]** For each row e and each item position $i \in [N]$, S computes $b_{e,i} = \mathrm{AES}_\kappa(x_{e,i}) \bmod F$ (with the convention $0 \mapsto 1$) and replaces $x_{e,i}$ with $b_{e,i}$ in place.
4) **[S Offline: share]** For each row e and each round $t \in \{0, \ldots, T+K-1\}$, S draws a fresh degree-$(k-1)$ Shamir polynomial $P_{t,e}(x) = s_{t,e} + a_{t,e}\, x$ with $a_{t,e} \xleftarrow{\$} \mathbb{F}_F$, where $s_{t,e} = 0$ for $t \in [0, T)$ and $s_{t,e}$ is the $(t{-}T{+}1)$-th chunk of ℓ_{x_e} for $t \in [T, T+K)$, and evaluates $P_{t,e}(i)$ for each $i \in [N]$.
5) **[S Offline: interpolate & pack]** For each round t, query position i, and partition p, S interpolates the unique degree-$(d{-}1)$ polynomial $f_{t,i,p}$ with $d = s_{part}$ through the points $\{(b_{e,i}, P_{t,e}(i)) : e \in p\}$ over $\mathbb{F}_F$ and SIMD-batches the coefficients of m/N partitions into one SEAL plaintext.
6) **[GC-OPRF: 1-GC]** R and S run a single two-party AES-128 garbled circuit [11]: R inputs $(y_1, \ldots, y_N)$, S inputs κ, R learns $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ for each $j \in [N]$ (with the convention $0 \mapsto 1$), and S learns $\perp$. This step is executed exactly once per online session and is reused across all $T+K$ subsequent rounds.
7) **[Online Query]** For each $j \in [N]$, R homomorphically encrypts a BSGS window of $O(\sqrt{d})$ powers of b_j and transmits it once; S expands it to the full power set $\{\mathrm{Enc}(b_j^\ell)\}_{\ell \in [0,d)}$ via $O(\sqrt{d})$ ciphertext multiplications and *caches* the bundle, reused by all $T+K$ rounds without re-expansion.
8) **[Per-Round Homomorphic Evaluation]** For each round $t \in \{0, \ldots, T+K-1\}$ in turn: for each (i, p), S homomorphically evaluates the partition polynomial $\mathrm{Enc}\big(f_{t,i,p}(b_j)\big) = \sum_{\ell=0}^{d-1} c_{t,i,p,\ell} \cdot \mathrm{Enc}(b_j^\ell)$ in batched form, and returns k SEAL ciphertexts per round (one per query position i, j consumed by reconstruction). R decrypts and stores the round-t evaluations $f_{t,i,p}(b_j)$ for every (i, j, p).
9) **[Receiver Reconstruction: token check]** For each partition p and each unordered pair $\{i, j\} \in \binom{[N]}{k}$, R computes $\mathsf{KR}(i, j, t)$ with the two-point Lagrange weights at $x = 0$. R records the pair $(p, \{i, j\})$ as a token candidate iff $\mathsf{KR}(i, j, t) = 0$ for *every* token round $t \in [0, T)$.
10) **[Receiver Reconstruction: label recovery]** For each token candidate $(p, \{i, j\})$, R computes $\mathsf{KR}(i, j, t)$ for every label round $t \in [T, T+K)$ to recover the K chunks of the label, and reassembles them into $\ell_{x_e} \in \mathcal{L}$.

Output. R outputs the multiset $\{\ell_{x_e} : e \text{ matched}\}$. S outputs $\perp$.

Fig. 1: CSTPSI protocol Π_T. Lemma 1 gives the realization soundness error (RSE) bound at any T; the multi-token-round mitigation drives the bound to negligible for $T \geq 2$ at the deployment scales of §IX.

returns the share set $\{(j, P(j))\}_{j \in [N]}$. KR takes any k of these shares and recovers s; fewer than k shares leave s information-theoretically uniform in $\mathbb{F}_F$.

B. BFV Homomorphic Encryption

The BFV scheme[6] is a semantically secure leveled homomorphic encryption scheme over the plaintext ring $\mathbb{Z}_F[X]/(X^m + 1)$, supporting SIMD batching in m parallel slots over $\mathbb{F}_F$ at the parameter regime presented in Table II. CSTPSI's homomorphic evaluation is a degree-$(s_{part}{-}1)$ univariate polynomial per query position, evaluated batched across $\lfloor m/N \rfloor$ partitions per ciphertext.

C. Garbled-Circuit OPRF

We realize the OPRF that blinds query and database items as a two-party AES-128 evaluation in semi-honest Yao garbled circuits, instantiated via the EMP toolkit [11].[7] The sender holds the 128-bit key κ, the receiver inputs the query items $(y_1, \ldots, y_N)$, and the receiver alone learns $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ for $j \in [N]$, with the convention $0 \mapsto 1$ on the zero pre-image; the sender learns $\perp$. The sender blinds its own database locally: $b_{e,i} = \mathrm{AES}_\kappa(x_{e,i}) \bmod F$.

[6]We instantiate the BFV scheme using Microsoft SEAL 4.1.2 [10].

[7]Standard semi-honest 2PC realization of the AES functionality.

D. Full Protocol Construction

Figure 1 formally states the CSTPSI protocol Π_T (parametrized by the token-round count T) in full, built from the three primitives above, and realizes the ideal functionality $\mathcal{F}_{\mathrm{CSTPSI}}$ in Functionality 1. Appendix C complements this with a message-sequence walkthrough (Figure 5) across the database, sender, and receiver.

CSTPSI runs in an *offline* phase (sender-only) and an *online* phase (two-party), adopting the standard PSI / HE optimizations for compressing communication and reducing HE circuit depth that have become canonical in the labeled-PSI line [7], [16], [36], [40]–[46]: *partitioning* the database into s_{part}-sized chunks, baby-step–giant-step (*BSGS*) windowing of the encrypted query powers, *SIMD batching* across partitions, and *noise flooding* plus *modulus switching* on the returned ciphertexts. We refer to those works for technique details and describe only how CSTPSI assembles them below.

We instantiate the design choices motivated in §V: per-round share construction (step 4), a single AES garbled circuit

Definition 4 (Semi-honest realization)**.** *Let Π_T be a CSTPSI protocol with security parameter λ and let $P \in \{S, R\}$ denote a corrupted party. The* real-world view

$$\mathsf{Real}_{\Pi_T,P}(\lambda; Y, Db)$$

is the tuple $(\mathsf{view}_P, y_R, y_S)$, where view_P collects P's input, randomness, and received messages over an honest execution of Π_T at security parameter λ on inputs (Y, Db), and (y_R, y_S) are the parties' outputs. The ideal-world view

$$\mathsf{Ideal}_{\mathcal{F}_{\mathrm{CSTPSI}},\mathsf{Sim},P}(\lambda; Y, Db)$$

is obtained by computing $(y_R, y_S) \leftarrow \mathcal{F}_{\mathrm{CSTPSI}}(Y, Db)$ and outputting $(\mathsf{Sim}(\lambda, P, x_P, y_P), y_R, y_S)$, where x_P is P's input and Sim *is a probabilistic polynomial-time simulator. We say that Π_T* semi-honestly realizes *$\mathcal{F}_{\mathrm{CSTPSI}}$ if, for every corrupted party $P \in \{S, R\}$, there exists a simulator Sim_P such that for every valid input pair (Y, Db),*

$$\mathsf{Real}_{\Pi_T,P}(\lambda; Y, Db) \overset{c}{\approx} \mathsf{Ideal}_{\mathcal{F}_{\mathrm{CSTPSI}},\mathsf{Sim}_P,P}(\lambda; Y, Db).$$

The maximum advantage of a polynomial-time distinguisher between the two distributions, taken over both P and all valid inputs, is the simulator gap *of Π_T.*

reused across all rounds (step 6), and one cached BSGS query-power window (step 7), concretely in Figure 1.

Offline. First, R and S agree on the BFV public parameters (Table II); R generates the BFV key pair and shares the public key with S. S then proceeds as follows. S samples an AES key κ, blinds each database item to $b_{e,i} = \mathrm{AES}_\kappa(x_{e,i}) \bmod F$, and, for every row e and every round $t \in \{0, \ldots, T + K - 1\}$, draws a fresh degree-$(k-1)$ Shamir polynomial $P_{t,e}(x) = s_{t,e} + a_{t,e}\, x$ with $s_{t,e} = 0$ for token rounds $t \in [0, T)$ and $s_{t,e}$ equal to the $(t-T+1)$-th chunk of ℓ_{x_e} for label rounds $t \in [T, T+K)$. Per query position $i \in [N]$ and per partition p, S interpolates the unique degree-$(s_{part}-1)$ polynomial $f_{t,i,p}$ over $\mathbb{F}_F$ through $\{\, (b_{e,i},\, P_{t,e}(i)) : e \in p \,\}$ and SIMD-packs $\lfloor m/N \rfloor$ partitions per SEAL plaintext.

Online. R obtains the blinded query $(b_1, \ldots, b_N) = (\mathrm{AES}_\kappa(y_1), \ldots, \mathrm{AES}_\kappa(y_N)) \bmod F$ via a single garbled-circuit AES execution, transmits a BSGS sparse window of encrypted query powers once, and decrypts $f_{t,i,p}(b_j)$ as S returns the round-t ciphertexts (re-randomized via noise flooding and shrunk via modulus switching). Reconstruction applies the Shamir reconstructor KR (of §VII-A) to every unordered pair $(i, j) \in \binom{[N]}{k}$ within each partition; the explicit $k = 2$ form is given in step 9 of Figure 1. Its Lagrange weights at $x = 0$ depend only on the blinded coordinates b_i, b_j, which the single garbled circuit fixes for the session, so R computes their modular inverses once per pair and reuses them across all $T + K$ rounds, sparing a per-round field inversion. The T token rounds, whose Shamir secret is the public token 0, are the independent validators of §V; Theorem 2 bounds their joint spurious accept at $1/F^T$.

Theorem 1 (Semi-honest security of Π_T)**.** *Assuming a semantically secure homomorphic encryption scheme, the pseudorandomness of AES-128, and a semi-honest-secure garbled-circuit protocol, the protocol Π_T realizes the ideal functionality $\mathcal{F}_{\mathrm{CSTPSI}}$ of Functionality 1 in the sense of Definition 4 against semi-honest adversaries with simulator gap at most*

$$\binom{N}{k} \frac{n_{part}}{F^T} \;+\; T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B}) \;+\; \mathsf{negl}(\lambda),$$

where the first term is the realization soundness error (RSE) bound of Theorem 2 (via Lemma 1), and the $\mathsf{negl}(\lambda)$ term absorbs the HE-semantic-security and GC simulator (privacy) gaps.

VIII. Security Analysis

The threat model and the ideal functionality $\mathcal{F}_{\mathrm{CSTPSI}}$ are stated in §II and Functionality 1 respectively. We adopt the standard simulation-based, real/ideal semi-honest security definition (Definition 4). This section states the formal definition, the security theorem for Π_T of §VII-D, and its proof sketch; the simulator constructions and supporting propositions are deferred to Appendix B.

A. Main Theorem

The simulator gap splits into two parts. The first is a *soundness* contribution, the RSE mass $\binom{N}{k} n_{part}/F^T$. On a spurious accept the real protocol deviates from $\mathcal{F}_{\mathrm{CSTPSI}}$ by accepting a query that shares fewer than k items and returning an off-curve value the ideal functionality would never produce. The second is a negligible *privacy* contribution, the HE, GC, and AES-PRF transcript-indistinguishability terms. Because the off-curve value is independent of the stored records (§IV-B), it leaks no sender data, so the gap's only operator-controlled term is a soundness quantity, not a privacy leak.

Proof sketch. Sender privacy (Appendix B-A): the simulator Sim_R populates non-match reconstructions uniformly in $\mathbb{F}_F$; the only event distinguishing it from the real view is the soundness (RSE) event of Lemma 1, bounded by $\binom{N}{k} n_{part}/F^T$ plus the AES-PRF hybrid cost. Although Lemma 1 is stated for a query that matches no entry, its proof bounds each non-matching entry independently, so the same union bound covers the non-matching pairs of an arbitrary query. Receiver privacy (Appendix B-B): the sender's view comprises one GC transcript and one ciphertext bundle of encrypted query powers, both simulable from $(Db, |Y|)$ under the HE semantic security and the standard semi-honest GC simulator. Propositions 1 and 2 (Appendix B-C) ensure the bound carries across the $T + K$ homomorphic rounds and the once-per-session GC optimization. □ □

a) Scope of the guarantee.: The simulator gap above is the difference between the real protocol and the ideal functionality. A natural match, where two unrelated records share at least k items by chance, is returned by the ideal functionality as well. It therefore appears in both worlds and

cancels in this gap. What remains is the protocol's RSE on queries that share fewer than k items, which the bound drives to a negligible level. The matcher's intrinsic false-match rate (FMR) is a property of the predicate, not of the protocol, so it is out of scope here and is the same as in plaintext.

B. Multi-Token-Round Amplification

The bound of Lemma 1 (§IV-B) gives the single-round $T = 1$ probability that Π_T accepts a non-matching query. We now show that the amplification mechanism the construction admits, T independent token rounds with independently sampled Shamir coefficients per round, drives this bound exponentially in T, with no degradation of the false reject rate.

Theorem 2 (Multi-Token Amplification)**.** *For $T \geq 2$ with independently sampled Shamir coefficients in each token round and a cryptographically secure source of randomness for those samples, the RSE of Lemma 1 satisfies*

$$\Pr[\text{accept}] \leq \binom{N}{k} n_{part} \,/\, F^T + T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B}). \quad (2)$$

Moreover, the false reject rate of Π_T matches that of Π_1: every true match lies in $\bigcap_t S_t$ deterministically, where S_t denotes the set of pairs (i, j) whose round-t reconstruction $\mathsf{KR}(i, j, t)$ equals the public token 0. Concrete parameter thresholds are stated in Corollary 1.

Proof sketch. The bound follows immediately from Lemma 1 once across-round independence is established. Independence of the T per-round reconstructions reduces to independence of the Shamir coefficient samples; in the reference implementation of §VII-D, this requires that each call for creating a secret share draw its coefficients from a cryptographically secure source.[8] False-reject preservation: at a true match (i, j) within an enrolled entry e, the receiver tries the same pair (i, j) in every round t; the sender's per-entry Shamir polynomial $P_{t,e}$ is sampled independently per round, but in every round it has constant term 0, so the KR interpolation at $x = 0$ through the points $(b_{e,i}, P_{t,e}(i))$ and $(b_{e,j}, P_{t,e}(j))$ deterministically yields 0. Hence $(i, j) \in S_t$ for all t, and $(i, j) \in \bigcap_t S_t$, with no probability of failure. □ □

The simulator gap of Theorem 1 is thus a tunable parameter, not a fixed weakness. Its only operator-controlled term, $\binom{N}{k} n_{part}/F^T$, is the soundness term of the composition of spurious accepts; by Corollary 1 an extra token round drives it below $2^{-\lambda}$ at the target scale, where it is dominated by the $\mathsf{negl}(\lambda)$ HE and GC terms and the AES-PRF cost. A non-negligible gap at $T = 1$ is therefore an artifact of running below the intended security parameter, not an inherent limitation.

The sender-privacy simulator construction in Appendix B-A uses Theorem 2 as its quantitative bound on this soundness term; together with the receiver-privacy argument and the multi-round propositions of Appendix B, this yields the bound of Theorem 1.

[8] In the reference implementation each thread's Shamir-coefficient generator is seeded from the operating-system CSPRNG (`arc4random_buf` on macOS/BSD, `/dev/urandom` on Linux), replacing the additive-feedback `random()` that an earlier prototype used.

Corollary 1 (Sufficient Number of Token Rounds)**.** *For RSE at most $2^{-\lambda}$ with security parameter λ, it suffices to choose*

$$T \geq \left\lceil \frac{\lambda \ln 2 + \ln \binom{N}{k} + \ln\lceil D/s_{part} \rceil}{\ln F} \right\rceil. \quad (3)$$

At the CSTPSI parameters $(N, k, F, s_{part}) = (64, 2, 8\,519\,681, 32)$, $T = 2$ suffices at million-scale databases under the engineering RSE threshold ($\lambda \approx 20$); $T = 3$ suffices to million-scale at the cryptographic standard $\lambda = 40$ and to billion-scale under the engineering threshold; and billion-scale at $\lambda = 40$ needs $T = 4$. The PRF advantage term contributes an additive $T \cdot 2^{-128}$ that does not affect any of these thresholds.

C. Leakage Profile

a) Receiver learns: (i) the labels of true matches, as required by $\mathcal{F}_{\mathrm{CSTPSI}}$; (ii) the approximate database size via the number of SIMD partitions transmitted (low severity; this parameter may be public); (iii) at $T = 1$, spurious accepts that return off-curve values at the rate of Lemma 1, the soundness term of Theorem 1 (driven to negligible at $T \geq 2$); and (iv) on each matched partition, the count and the identities of the agreeing component positions (revealed only when the count reaches k and a label is recovered).

The component-position leakage of (iv) is inherent to the set-threshold reduction and is shared by every protocol in this class; what distinguishes CSTPSI is that the multi-token amplification confines it to genuine matches. At $T = 1$ the same indices leak on each spurious accept, that is, for a $\binom{N}{k} n_{part}/F$ fraction of non-matching rows; driving that rate to negligible at $T \geq 2$ is precisely what restricts the leakage to true matches.

b) Sender learns: nothing of cryptographic value. The constant GC online cost eliminates the label-size timing side channel that a naive multi-round-GC composition would exhibit.

c) Network observer learns: the round count $T+K$, from which an observer can infer the chosen label size unless that parameter is treated as public, and fixed-size message bodies that carry no item-specific information.

IX. Evaluation

A. Experimental Setup

We implement CSTPSI in C++ from scratch; the source and a one-command reproduction harness are publicly available under the PolyForm Noncommercial License.[9] All experiments run on a single workstation with a 12-core (6 performance, 6 efficiency) processor and 36 GB of memory under a current desktop operating system. The sender and receiver

[9] https://github.com/euzun/cstpsi

TABLE III: Per-query online time (s) and communication (MiB), grouped by thread count (1, 4, 8). CSTPSI runs at $T = 2$ token rounds, STLPSI at $T = 1$. Faster and lower values are bolded; (-) marks cases where CSTPSI sends more than the baseline.

		Online Time, 1 thr (s)			**Online Time, 4 thr (s)**			**Online Time, 8 thr (s)**			**Comm. (MiB)**		
D	Label	STL	CST	Spd.	STL	CST	Spd.	STL	CST	Spd.	STL	CST	Save
1K	23-bit	3.3	**1.7**	1.9×	3.3	**1.7**	2.0×	3.4	**1.7**	2.0×	6	**3**	50%
	16-B	11.3	**1.7**	6.5×	11.5	**1.7**	6.7×	11.4	**1.7**	6.7×	20	**3**	85%
	32-B	21.3	**1.9**	11.5×	21.0	**1.7**	12.0×	21.1	**1.8**	12.0×	36	**4**	89%
	64-B	38.8	**2.0**	19.4×	40.1	**1.9**	21.2×	40.1	**1.9**	21.2×	67	**5**	93%
10K	23-bit	3.4	**1.9**	1.8×	3.3	**1.7**	1.9×	3.5	**1.8**	1.9×	6	**4**	33%
	16-B	11.8	**2.2**	5.3×	11.8	**1.9**	6.2×	11.5	**1.8**	6.4×	22	**6**	73%
	32-B	21.9	**2.7**	8.1×	21.9	**2.1**	10.6×	21.5	**1.9**	11.1×	41	**9**	78%
	64-B	40.6	**3.5**	11.5×	41.2	**2.4**	17.3×	39.7	**2.2**	18.4×	75	**14**	81%
100K	23-bit	4.9	**4.0**	1.2×	3.9	**2.4**	1.6×	3.8	**2.3**	1.7×	**14**	16	-
	16-B	16.6	**7.7**	2.2×	12.9	**3.5**	3.7×	12.7	**3.0**	4.2×	49	**37**	24%
	32-B	31.4	**12.1**	2.6×	24.8	**4.9**	5.0×	23.2	**3.9**	5.9×	91	**63**	31%
	64-B	56.7	**20.0**	2.8×	44.7	**7.4**	6.0×	45.6	**5.9**	7.8×	169	**111**	34%
1M	23-bit	**19.6**	25.9	0.8×	**8.3**	9.1	0.9×	**6.7**	6.8	1.0×	**92**	132	-
	16-B	65.3	**62.7**	1.0×	28.3	**20.5**	1.4×	23.0	**14.6**	1.6×	**322**	349	-
	32-B	119.9	**106.9**	1.1×	52.1	**34.1**	1.5×	41.9	**23.9**	1.8×	**597**	608	-
	64-B	219.7	**187.8**	1.2×	95.6	**59.1**	1.6×	79.9	**42.2**	1.9×	1103	**1084**	2%

run as separate processes communicating over the loopback interface with ZeroMQ [47]; network conditions are simulated at 10 Gbps bandwidth and 0.02 ms latency. Each case uses 300 true-positive and 300 true-negative query draws, reduced to 100 of each for the heaviest configurations (32- and 64-byte labels and all $D = 1$M cases); timing, communication, and memory are reported as the median over independent runs. The cryptographic stack is SEAL [10] (BFV at ≈128-bit security) for the homomorphic layer, EMP-toolkit [11] for the AES-128 oblivious-PRF garbled circuit, and FLINT [48] with GMP and MPFR for polynomial arithmetic; threading uses OpenMP.

We use two data regimes. The synthetic regime (Sections IX-C and IX-E) realizes a perfect matcher with zero false-match rate. Every true-negative query is built to share fewer than k items with every enrolled record, so no natural match can occur and every accept is purely the kernel's realization soundness error. This isolates the RSE and shows clearly how it scales with the database size. The real regime (Section IX-D) uses two public datasets at different scales, LFW [49] and Deep1B [50], to demonstrate the attack on a real matcher and to show that CSTPSI restores the RSE to zero.

B. Benchmarked Protocols and Headline Result

We also implemented the STLPSI protocol of Uzun et al. [7] from scratch and exposed it within the same tool as an alternative mode, so that both protocols run on identical code paths and hardware. The two modes are:

- **CSTPSI**: the deployment configuration, which evaluates the garbled circuit once per session, transmits the encrypted query once, and applies the multi-token-round mitigation that drives the realization soundness error (RSE) to its amplified bound.
- **STLPSI**: the baseline, with a single token round; the oblivious-PRF garbled circuit and the encrypted query are recomputed and retransmitted on every protocol round.

Both modes share the standard labeled-PSI optimizations (partitioning, SIMD batching, baby-step–giant-step query-power windowing), differing only in whether the garbled circuit and encrypted query are cached across rounds (CSTPSI) or recomputed each round (STLPSI); the reported speedups thus reflect this protocol-level caching, not an unoptimized baseline.

The protocol parameters and the swept ranges are collected in Table II (§IV); the security level is ≈128 bits.

a) Headline result: At a million records ($D = 1$M, 8 threads) the no-caching STLPSI baseline ($T = 1$) has an RSE of one, admitting a spurious accept on every spoofing query, while CSTPSI ($T = 2$) admits none. CSTPSI buys this soundness essentially for free. Across the tested range it matches or beats the baseline, running more than $20\times$ faster and sending up to 93% less data at best (at $D = 1$K with a 64-byte label) and staying within about 2% of it at worst. Corollary 1 projects an engineering-negligible RSE ($\leq 10^{-6}$) through a million records, which the run confirms.

C. Correctness and Realization Soundness Error

We measure the protocol's RSE, not the matcher FMR. Every true-negative query is drawn to share fewer than k items with every enrolled record, so any accept is a protocol artifact and not a natural match; the intrinsic FMR is a plaintext property of the representation, reported as the FMR floor in §IX-D.

Fig. 2 plots the measured $T = 1$ RSE against the bound $\binom{N}{k} n_{part}/F^T$ of Lemma 1. It grows with D as predicted, from under 1% at 1K through 4.5% at 10K and 56% at 100K to 100% at 1M, tracking the exact bound; the linear

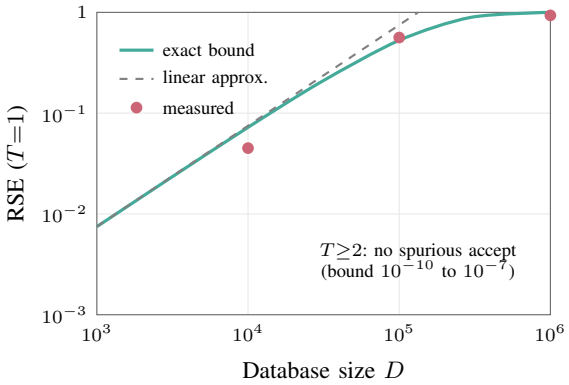

Fig. 2: Empirical RSE versus database size D (log–log axes).

approximation overestimates above $D \approx 10^4$. The 1K point is omitted from the log plot, as no spurious accept was observed in its sample (consistent with the $\sim$1% bound). A single token round removes the error: no spurious accept was observed at $T \geq 2$ through $D = 1$M, where the bound falls from 10^{-10} at 1K to 10^{-7} at 1M, and Corollary 1 gives the sufficient T for any target scale. Every true-positive query is accepted (false-reject rate 0 throughout), so the fix costs no recall.

D. Soundness on Real Datasets

The synthetic study isolates the kernel's RSE; we now confirm the same effect on real data at two scales. Deep1B carries the scale story up to a million records, and LFW checks end-to-end accuracy on a real biometric matcher. We measure the per-query RSE as the fraction of queries with at least one spurious accept. Such an accept returns an off-curve value, not an enrolled label (§IV-B), so this fraction measures the composition gap directly.

a) Datasets and setup: Deep1B [50] includes 96-dimensional deep object descriptors. We enroll D records up to 10^6 and issue, per repeat, 100 genuine queries (each close to one enrolled record and in the published query vectors) and 100 random queries (no planted similarity), over 10 repeats. LFW [49] is a widely used dataset for face-recognition accuracy. It supplies 12,209 face embeddings; we use the 128-dimensional vectors. We built a lightweight FLPSI wrapper that consumes these embedding vectors and reduces them to N-item sets, with the exact parameters and conversion flow prescribed in FLPSI'21 [7]. The wrapper and the input embeddings for this experiment are in our released codebase. We enroll $D = 1500$ records and issue, per repeat, a genuine query for each enrolled record and 100 impostor queries, over 10 repeats. Both datasets pass through the same locality-sensitive hash to a 256-bit sketch and use $T = 2$ for CSTPSI.

TABLE IV: Deep1B kernel soundness vs. DB size: the baseline RSE climbs to one. CSTPSI stays at zero (FRR identical).

D	10^3	10^4	10^5	10^6
k-of-N FRR	0.112	0.112	0.112	0.112
STLPSI RSE (T=1)	0.010	0.080	0.504	0.999
CSTPSI RSE (T=2)	**0.000**	**0.000**	**0.000**	**0.000**

TABLE V: LFW end-to-end accuracy at the plaintext k-of-N operating point. CSTPSI matches the matcher's false-accept and false-reject rates; STLPSI inflates the false-accept rate.

Configuration	FAR	FRR
k-of-N (plaintext)	0.337	0.324
STLPSI (T=1)	0.352	0.324
CSTPSI (T=2)	0.337	0.324

b) Deep1B: the RSE approaches one: The baseline (STLPSI, $T = 1$) RSE climbs with D, from 0.010 at 10^3 to 0.080, 0.504, and 0.999 at 10^6, matching the synthetic curve of Fig. 2; at a million records almost every query produces a spurious accept, about seven off-curve values per query.[10] CSTPSI holds the RSE at zero throughout, and the false-reject rate is identical for the plaintext matcher, STLPSI, and CSTPSI (0.112), so the fix costs no recall. The random and genuine queries behave alike, confirming the effect follows from the database size, not the data, as Lemma 1 predicts.

The matcher's own false matches run far higher. Its FMR floor rises from about 0.95 to 267 real matches per query as D grows from 10^3 to 10^6, but that floor is a plaintext property of the representation, shared by the matcher, STLPSI, and CSTPSI, and out of scope here (§II-B). What the protocol owns is the RSE, which STLPSI drives to one and CSTPSI removes entirely. The floor being the larger number is exactly why the two must be separated. CSTPSI zeros the protocol's contribution, bringing the deployed system down to the matcher's floor, which amplification cannot and should not go below (§X).

c) LFW: a real biometric matcher: Table V reports end-to-end behavior on faces. LFW is a stringent test because the plaintext matcher is noisy at this operating point: at the k-of-N threshold it already sits at a high false-accept rate of 0.337 and false-reject rate of 0.324. The test is faithfulness, not accuracy. Whatever the plaintext operating point, the kernel must reproduce it. CSTPSI does: it matches the k-of-N false-accept and false-reject rates (0.337 and 0.324) to the digit, so the cryptographic layer adds nothing to the matcher's own error. STLPSI, without the soundness fix, instead pushes the false-accept rate above the matcher floor, to 0.352. Deep1B carries the scale story; LFW shows the kernel tracks a real matcher's operating point under encryption, however noisy that matcher is.

E. Performance of CSTPSI

Having established that the deployment configuration zeroes the RSE, we show it does so without sacrificing performance. Table III reports the head-to-head of CSTPSI against the STLPSI baseline across database sizes, label sizes, and thread counts, with the per-case speedup and communication saving.

Scaling with database size. CSTPSI's edge over the baseline is largest on small and moderate databases (Fig. 3).

[10]The split counts a reconstruction as a label when its value falls in $[0, D)$ and as off-curve otherwise. An off-curve value falls in $[0, D)$ by chance with probability D/F, which is 11.7% at $D = 10^6$, so the reported count is a lower bound on the true off-curve rate.

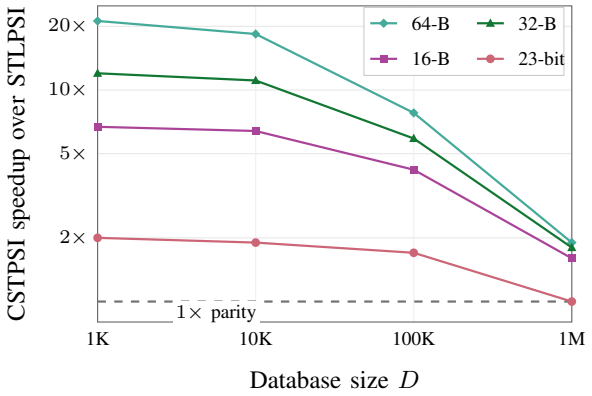

Fig. 3: Per-query online-time speedup of CSTPSI over STLPSI (8 threads) versus database size D, for each label size (both axes log scale; the dashed line marks $1\times$, parity).

Its homomorphic work per round grows with D, so as the database grows this per-round cost catches up with the one-time saving from running GC once. Through $D = 100$K CSTPSI is faster at every label size. At $D = 1$M it is still faster at all but the smallest label: with a 23-bit label the extra security round of T=2 outweighs GC saving, and the two protocols run within about 2% of each other. CSTPSI thus matches or beats the baseline at every operating point we measure, with no regime in which it is more than marginally slower.

Scaling with label size. STLPSI re-runs GC and retransmits the encrypted query on every round, where they grow with the label. CSTPSI runs GC once and sends the query once, so its advantage widens as labels grow (the upper curves in Fig. 3). The effect peaks at $D = 1$K with a 64-byte label, where CSTPSI is more than $20\times$ faster than the baseline at 8 threads (Fig. 3) and sends about 93% less data.

Scaling with threads. Threads help most where homomorphic evaluation dominates, that is at large databases. At $D = 100$K the speedup from 1 to 8 threads reaches about $2.7\times$. At $D = 1$K the garbled circuit runs single-threaded, leaving little to parallelize, so extra threads change little.

Communication. Per-query communication has two parts, the receiver's query upload ($R\to S$) and the sender's encrypted response ($S\to R$); Table III and Fig. 4 give the per-case breakdown. The key effect is that caching the encrypted query fixes $R\to S$ at a constant 2.7 MiB per query in CSTPSI, independent of database and label size, whereas STLPSI re-sends it on each of its $1 + K$ rounds; the response $S\to R$ instead scales with D in both protocols and dominates at scale. Caching therefore wins decisively where the upload dominates, on small to moderate databases with large labels (up to 93% less data), and the advantage narrows or reverses at the largest databases, where the extra $T = 2$ response round makes CSTPSI send about 40% more for a 23-bit label at $D = 1$M. These extra bytes are modest in absolute terms and are the price of the security round whose benefit Section IX-C quantifies.

Peak RAM. Peak resident memory is dominated by the sender, which holds the partitioned database, and grows with D. It rises from well under 1 GB at small databases to about 5.3–5.6 GB at $D = 100$K (64-byte) and 8.0–9.1 GB at $D = 1$M (23-bit, $T = 2$), a modest overhead over the baseline; the receiver's footprint stays under 1 GB throughout.

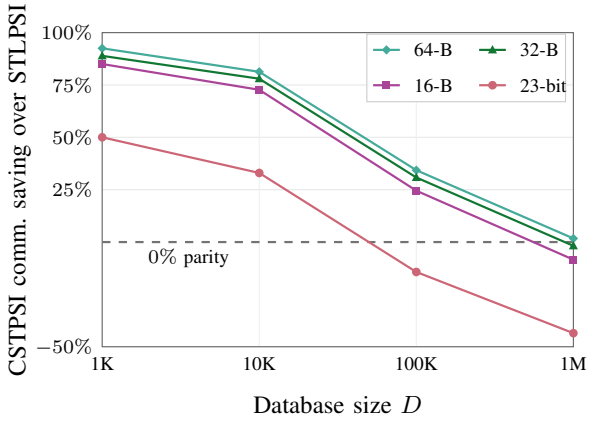

Fig. 4: Communication saving of CSTPSI over STLPSI versus database size D, for each label size (0% is parity; negative values mean CSTPSI sends more).

F. Optimization Breakdown

We add CSTPSI's caching optimizations one at a time and report each step as a factor on the per-query online time, with no absolute timings, so the contribution reads independently of any single case. The base is CSTPSI at $T = 2$ with no caching, which is exactly the baseline run at $T = 2$. Relative to the original $T = 1$ baseline this base pays only the security fix, one extra homomorphic round out of $T+K$, a tax of about 10% at the 23-bit label and 3% at the 64-byte label. On top of the base we add three caching steps in sequence (Table VI): a single garbled circuit per session, with the receiver still re-sending its encrypted query powers each round; then the sender caching the expanded power window, so the receiver sends it only once; then the receiver caching its reconstruction inverses.

TABLE VI: Cumulative speedup of each caching step over the unoptimized base (CSTPSI at T=2, no caching), per database size (factors).

	Speedup over base at $D =$			
Caching step (cumulative)	1K	10K	100K	1M
+ once-per-session GC	21.44×	18.59×	7.91×	1.95×
+ sender power-window cache	22.06×	19.06×	7.98×	1.95×
+ receiver reconstruction cache	22.07×	19.13×	8.09×	1.99×

Every entry in Table VI is a speedup factor over the no-caching base. The garbled-circuit step dominates. Evaluating it once instead of on each of the $1 + K$ rounds removes a $K/(1 + K)$ fraction of the oblivious-PRF cost, from 50% at the 23-bit label to 96% at the 64-byte label. Because that cost dominates the baseline at small and moderate D, this single step alone recovers the security fix's tax many times over. The sender power-window cache then uploads the encrypted query

powers once rather than on every round. The reconstruction cache replaces a per-pair field inversion with a cached lookup, which makes the receiver reconstruction about $2.5\times$ faster. That saving grows with D, from under a millisecond at 1K to about 0.8 seconds at 1M, yet it stays a small share of the online time, which the homomorphic evaluation dominates. End to end, the optimizations turn the added $T = 2$ round from a cost into a net speedup over the baseline across the tested range.

G. Artifact and Reproducibility

We provide our complete source together with single-command shell scripts that regenerate every table and figure in this section from fixed seeds, and an interactive tool that runs the protocol end to end. The artifact and documentation are publicly available at https://github.com/euzun/cstpsi.

X. Discussion

Discussion. CSTPSI's central property is that caching decouples the dominant online costs from the label size, so its advantage over the baseline widens as labels grow (§IX-E). One deployment nuance is worth noting. On a fast (10 Gbps) link the homomorphic evaluation, not transmission, dominates latency, so the caching primarily reduces communication volume rather than wall-clock time. Throughout, two token rounds keep the realization soundness error (RSE) below engineering thresholds for databases up to one million records.

Generalization. The composition gap is not specific to this kernel. Any realization that buys efficiency with a per-trial false accept and runs one trial per record inherits it, and the set-threshold kernel is simply where the gap separates cleanly from the matcher's FMR floor and can be measured (§II). The remedy generalizes as a design principle. When a matcher's soundness rests on a per-trial event whose mass scales with the workload, make the check a first-class, independent, in-protocol validator rather than an after-the-fact output filter, and size the number of validators to the deployment scale through a closed-form bound (Lemma 1); soundness then composes with the workload instead of degrading with it, since a spurious accept must fool all T validators at once. CSTPSI is the threshold-labeled fuzzy-PSI instance, and we state this as a measured observation about a class of protocols, not a proof of the general case.

Limitations. CSTPSI removes the kernel's RSE, but it does not change the matcher's intrinsic FMR. Two unrelated records can still share at least k items by chance, and the expected number of such natural matches per query grows with the database size. The application controls this rate through the representation and the choice of N and k, not through the protocol. CSTPSI therefore brings the deployed system down to this FMR floor, and amplification cannot go below it. Amplification is free on the false-reject side, because a true match shares the actual items and reconstructs in every round, so adding token rounds removes only spurious accepts. Reducing the FMR itself would instead need independent encodings across the rounds, which would raise the false-reject rate, and we leave this to future work. A second limit is communication. The sender-to-receiver response grows by one field element per round, so it scales with the label length, and the construction targets small-to-moderate labels rather than bulk payloads (at one field element of about 2.8 bytes per round, a one-megabyte label would need roughly $350{,}000$ rounds). A third limit is the threshold, fixed at $k = 2$. A larger k inflates the factor $\binom{N}{k}$ (for example, from $\binom{64}{3} = 41{,}664$ to $\binom{64}{5} \approx 7.6 \times 10^6$) and raises the single-round RSE, so it needs more token rounds for the same target. Seen positively, amplification is what makes a larger k usable at all, because without it the single-round rate at $k = 3$ already approaches certainty on modest databases. Finally, our analysis is semi-honest, and malicious security is future work. Amplification raises the cost of cryptographic probing, where an adversary crafts queries to trigger field collisions, but it does not raise the cost of natural-match probing. Token and label rounds are independent (Proposition 1) but are sent sequentially, so pipelining them for throughput is also future work.

XI. Conclusion

Standard per-trial analysis for FLPSI fails under realistic workloads, where a spurious accept is not a mere accuracy artifact but a soundness defect: the kernel accepts a query the plaintext matcher would reject, and returns a value it never would. We expose this workload dependence through a composable bound (Lemma 1, Theorem 2) and close the resulting gap with CSTPSI, whose cached query and garbled circuit keep its added security affordable: it holds the realization soundness error (RSE) within engineering thresholds at a million records with two token rounds, and the bound extends that guarantee to a billion records with a third. Our evaluation, on synthetic and real data, records no spurious accept under amplification.

XII. LLM Usage Statement

The authors used an LLM solely for language editing, grammar correction, figure drafting assistance, table formatting, and minor presentation improvements. The LLM did not contribute any conceptual ideas, protocol designs, methodological innovations, experimental strategies, security analyses, or scientific insights. All intellectual and technical contributions originated entirely from the authors. Any material provided to the LLM for editing or formatting contained no sensitive, personal, confidential, or ethically restricted information.

References

[1] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology – EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, vol. 3027. Springer, 2004, pp. 1–19.

[2] B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on OT extension," in *23rd USENIX Security Symposium (USENIX Security 14)*. USENIX Association, 2014, pp. 797–812.

[3] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, "Efficient batched oblivious PRF with applications to private set intersection," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 818–829.

[4] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "PSI from PaXoS: Fast, malicious private set intersection," in *Advances in Cryptology – EUROCRYPT 2020*, ser. Lecture Notes in Computer Science, vol. 12106. Springer, 2020, pp. 739–767.

[5] P. Rindal and P. Schoppmann, "VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE," in *Advances in Cryptology – EUROCRYPT 2021*, ser. Lecture Notes in Computer Science, vol. 12697. Springer, 2021, pp. 901–930.

[6] S. Raghuraman and P. Rindal, "Blazing fast PSI from improved OKVS and subfield VOLE," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2022, pp. 2505–2517, ePrint 2022/320, https://eprint.iacr.org/2022/320.

[7] E. Uzun, S. P. Chung, V. Kolesnikov, A. Boldyreva, and W. Lee, "Fuzzy labeled private set intersection with applications to private real-time biometric search," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2021, pp. 911–928.

[8] A. Chakraborti, G. Fanti, and M. K. Reiter, "Distance-aware private set intersection," in *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, 2023, arXiv:2112.14737.

[9] H. Chen, I. Chillotti, Y. Dong, O. Poburinnaya, I. Razenshteyn, and M. S. Riazi, "SANNS: Scaling up secure approximate k-nearest neighbors search," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 2020, pp. 2111–2128.

[10] Microsoft Research, "Microsoft SEAL (release 4.x)," https://github.com/microsoft/SEAL, 2023.

[11] X. Wang, A. J. Malozemoff, and J. Katz, "EMP-toolkit: Efficient multiparty computation toolkit," https://github.com/emp-toolkit, 2016.

[12] D. Demmler, T. Schneider, and M. Zohner, "ABY: A framework for efficient mixed-protocol secure two-party computation," in *22nd Annual Network and Distributed System Security Symposium (NDSS)*, 2015.

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[14] A. C. D. Resende and D. de Freitas Aranha, "Faster unbalanced private set intersection," in *Financial Cryptography and Data Security (FC 2018)*, ser. Lecture Notes in Computer Science. Springer, 2018, ePrint 2017/677.

[15] D. Kales, C. Rechberger, T. Schneider, M. Senker, and C. Weinert, "Mobile private contact discovery at scale," in *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, 2019.

[16] K. Cong, R. C. Moreno, M. B. da Gama, W. Dai, I. Iliashenko, K. Laine, and M. Rosenberg, "Labeled PSI from homomorphic encryption with reduced computation and communication," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021, pp. 1135–1150.

[17] G. Garimella, B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "Oblivious key-value stores and amplification for private set intersection," in *Advances in Cryptology – CRYPTO 2021*, ser. Lecture Notes in Computer Science, vol. 12828. Springer, 2021, pp. 395–425, iACR ePrint 2021/883.

[18] A. van Baarsen and S. Pu, "Fuzzy private set intersection with large hyperballs," in *Advances in Cryptology – EUROCRYPT 2024*, ser. Lecture Notes in Computer Science, M. Joye and G. Leander, Eds., vol. 14655. Springer, 2024, pp. 340–369, ePrint 2024/330, https://eprint.iacr.org/2024/330.

[19] L. Chmielewski and J.-H. Hoepman, "Fuzzy private matching," in *2008 Third International Conference on Availability, Reliability and Security*. IEEE, 2008, pp. 327–334.

[20] Q. Ye, R. Steinfeld, J. Pieprzyk, and H. Wang, "Efficient fuzzy matching and intersection on private datasets," in *International Conference on Information Security and Cryptology*. Springer, 2009, pp. 211–228.

[21] I. Calapodescu, S. Estehghari, and J. Clier, "Compact fuzzy private matching using a fully-homomorphic encryption scheme," Aug. 29 2017, uS Patent 9,749,128.

[22] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *International conference on information security and cryptology*. Springer, 2009, pp. 229–244.

[23] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International symposium on privacy enhancing technologies symposium*. Springer, 2009, pp. 235–253.

[24] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "Scifi-a system for secure face identification," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 239–254.

[25] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti *et al.*, "Privacy-preserving fingercode authentication," in *Proceedings of the 12th ACM workshop on Multimedia and security*, 2010, pp. 231–240.

[26] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *European Symposium on Research in Computer Security*. Springer, 2011, pp. 190–209.

[27] G. Garimella, M. Rosulek, and J. Singh, "Structure-aware private set intersection, with applications to fuzzy matching," in *Advances in Cryptology – CRYPTO 2022, Part I*, ser. Lecture Notes in Computer Science, vol. 13507. Springer, Heidelberg, 2022, pp. 323–352, ePrint 2022/1011, https://eprint.iacr.org/2022/1011.

[28] Y. Gao, L. Qi, X. Liu, Y. Luo, and L. Wang, "Efficient fuzzy private set intersection from fuzzy mapping," in *Advances in Cryptology – ASIACRYPT 2024*, 2024, code: https://github.com/ql70ql70/Fuzzy-Private-Set-Intersection-from-Fuzzy-Mapping.

[29] L. Piske, J. Singh, N. Trieu, V. Kolesnikov, and V. Zikas, "Distance-aware OT with application to fuzzy PSI," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '25. ACM, 2025, pp. 4679–4691, full version: ePrint 2025/996.

[30] C. Zhang, Y. Chen, Y. Cao, Y. Bai, S. Li, J. Lin, A. Wang, and X. Wang, "Fast fuzzy PSI from symmetric-key techniques," 2025, iACR ePrint 2025/885, https://eprint.iacr.org/2025/885; venue not stated in manuscript.

[31] X. Yang, M. Hao, C. Weng, R. H. Deng, Y. Wen, and T. Zhang, "Efficient fuzzy private set intersection from secret-shared OPRF," in *Proceedings of the 2026 IEEE Symposium on Security and Privacy (S&P)*, 2026, arXiv:2604.14909v1.

[32] S. Meng, X. Wang, X. Zhou, and B. Liang, "Unbalanced fuzzy private set intersection for L_∞ distance: Achieving sublinear communication with large set size," in *35th USENIX Security Symposium (USENIX Security 26)*. USENIX Association, 2026, code: https://doi.org/10.5281/zenodo.18223806.

[33] C. Dang, X. Zhou, and B. Liang, "Efficient fuzzy PSI based on prefix representation," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2025, pp. 2204–2218.

[34] S. Ghosh and M. Simkin, "The communication complexity of threshold private set intersection," in *Advances in Cryptology - CRYPTO 2019*, 2019, pp. 3–29.

[35] S. Badrinarayanan, P. Miao, S. Raghuraman, and P. Rindal, "Multi-party threshold private set intersection with sublinear communication," in *Public-Key Cryptography - PKC 2021*, 2021.

[36] H. Chen, Z. Huang, K. Laine, and P. Rindal, "Labeled PSI from fully homomorphic encryption with malicious security," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2018, pp. 1223–1237.

[37] N. Chandran, D. Gupta, and A. Shah, "Circuit-PSI with linear complexity via relaxed batch OPPRF," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2022, no. 1, 2022, pp. 353–372.

[38] B. Pinkas, T. Schneider, O. Tkachenko, and A. Yanai, "Efficient circuit-based PSI with linear communication," in *Advances in Cryptology – EUROCRYPT 2019*, ser. Lecture Notes in Computer Science, vol. 11478, 2019, pp. 122–153.

[39] A. Boldyreva and N. Chenette, "Efficient fuzzy search on encrypted data," in *Fast Software Encryption (FSE)*, ser. LNCS, vol. 8540. Springer, 2014, pp. 613–633.

[40] H. Chen, K. Laine, and P. Rindal, "Fast private set intersection from homomorphic encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1243–1255.

[41] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 13:1–13:36, 2014, preliminary version: ITCS 2012.

[42] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

[43] B. Pinkas, T. Schneider, G. Segev, and M. Zohner, "Phasing: Private set intersection using permutation-based hashing," in *24th USENIX Security Symposium (USENIX Security)*, 2015, pp. 515–530.

[44] B. Pinkas, T. Schneider, C. Weinert, and U. Wieder, "Efficient circuit-based PSI via cuckoo hashing," in *Advances in Cryptology – EUROCRYPT 2018*, ser. LNCS, vol. 10822. Springer, 2018, pp. 125–157.

[45] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography (TCC)*, ser. LNCS, vol. 3378. Springer, 2005, pp. 303–324.
[46] L. Ducas and D. Stehlé, "Sanitization of FHE ciphertexts," in *Advances in Cryptology – EUROCRYPT 2016*, ser. LNCS, vol. 9665. Springer, 2016, pp. 294–310.
[47] "ØMQ (ZeroMQ): An open-source universal messaging library," https://zeromq.org, 2024.
[48] W. Hart *et al.*, "FLINT: Fast Library for Number Theory (release 3.4)," https://flintlib.org, 2024.
[49] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.
[50] A. Babenko and V. Lempitsky, "Efficient indexing of billion-scale datasets of deep descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2055–2063.

APPENDIX A
QUERY-POWER TRANSMISSION VARIANTS

The sender's per-partition polynomial evaluation $f_{t,i,p}(b_j) = \sum_{\ell=0}^{d-1} c_{t,i,p,\ell} \cdot \mathrm{Enc}(b_j^\ell)$ with $d = s_{part}$ requires ciphertexts $\mathrm{Enc}(b_j^\ell)$ for every $\ell \in [0, d)$. Two variants populate this set; both are compatible with the cross-round caching of §V.

A. Variant A (full-receiver-side).

The receiver knows b_j in plaintext (from the GC-OPRF of step 6), computes b_j^ℓ for $\ell = 0, \ldots, d-1$ modulo F, encrypts each, and transmits the full set of d ciphertexts. Sender per-round work: d `multiply_plain` operations at depth one. No ciphertext-ciphertext multiplications.

B. Variant B (BSGS).

The receiver transmits an $O(\sqrt{d})$-element baby-step–giant-step sparse window following APSI-style labeled PSI [16]; the sender expands to the full power set via $O(\sqrt{d})$ ciphertext-ciphertext multiplications plus relinearization at session setup, consuming a small constant number of additional BFV multiplicative levels.

C. Trade-off.

Variant B saves $\sim \sqrt{d}\times$ in communication (at $d = 32$, ~ 0.24 MiB vs ~ 1.3 MiB per query-item bundle) but pays a session-setup tax: relinearization in SEAL runs roughly $20-40\times$ a `multiply_plain` on Intel servers with AVX-512+AES-NI, and disproportionately slower on hardware without those instructions (ARM, AMD pre-Zen3 without AVX-512, embedded). Variant B's extra depth also forces a longer coeff_modulus chain, paying a $\sim 1.3\times$ ciphertext-size overhead on every subsequent operation. Variant B therefore dominates when the network is the bottleneck and the hardware is Intel-AVX-tuned (WAN deployments on modern servers); Variant A dominates under co-located benchmarks or non-Intel hardware.

D. Reference artifact.

The benchmarks of §IX co-locate sender and receiver over a ZeroMQ loopback transport, where Variant B's communication savings are negligible while its cipher-cipher session-setup tax remains. The artifact therefore implements Variant A, aligning with the benchmarked regime.

E. Caching contribution and follow-up.

Independent of the variant, CSTPSI's cross-round caching of the power bundle avoids $(T + K - 1)$ session-setup passes per query session; under Variant B this is $(T+K-1)\cdot O(\sqrt{d})$ ciphertext-ciphertext multiplications avoided, on the order of seconds at typical K on the reference hardware. A full Variant B implementation with a decoupled ablation flag and cross-hardware measurement is queued for follow-up.

APPENDIX B
FULL SECURITY PROOF OF Π_T

This appendix gives the simulator constructions $\mathsf{Sim}_R, \mathsf{Sim}_S$ and the supporting propositions that underpin the proof sketch of Theorem 1. Throughout, the real-world view $\mathsf{Real}_{\Pi_T,P}$ and the ideal-world view $\mathsf{Ideal}_{\mathcal{F}_{\mathrm{CSTPSI}},\mathsf{Sim}_P,P}$ are as in Definition 4 of §VIII. We proceed party by party: for each corrupted party we describe the real view step by step (indicating which steps of Π_T place messages into the view), construct the simulator step by step (mirroring those messages), and argue indistinguishability per step. The substantive (soundness) term of the simulator gap, the realization soundness error (RSE) composition bound from Lemma 1 amplified by Theorem 2, falls out of the reconstruction step on the receiver-side walk. Two supporting propositions (multi-round soundness and 1-GC transcript equivalence) are stated at the end of the appendix and cited from inside the walks.

A. Sender Privacy: Simulating the Receiver

Sim_R takes as input $(\lambda, R, Y, \mathcal{F}_{\mathrm{CSTPSI}}(Y, Db))$: the security parameter, the corrupted party label, the receiver's query Y, and the ideal-world output (the multiset of matched labels). R's view consists of its input, output, randomness, and the messages it receives; the simulator's task is to emulate those messages consistently with the ideal output.

*a) **Describing the receiver's view.**:* We walk through the steps of CSTPSI protocol Π_T (Figure 1) to describe view_R.

Steps 1–5 are sender-internal: S samples κ, partitions, blinds with AES_κ, constructs Shamir shares, and interpolates and SIMD-packs the per-partition polynomials. R receives no messages, so Sim_R does nothing to simulate them.

In Step 6, R and S run a single 1-GC AES execution: R inputs $(y_1, \ldots, y_N)$, S inputs κ, and R obtains $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ (with the $0 \mapsto 1$ convention) for every $j \in [N]$. R receives GC messages whose transcript is computationally indistinguishable from a simulated transcript by the security of the underlying semi-honest GC protocol [11]. The GC output $\{b_j\}_{j\in[N]}$ enters view_R and must be emulated. By Proposition 2, the single 1-GC execution is transcript-equivalent to the naive $(T + K)$-GC composition, so this analysis suffices for the entire session.

In Step 7, R prepares the BSGS sparse window $\{\mathrm{Enc}(b_j^\ell) : \ell \in W\}$ under its own BFV public key and transmits it once. R sends here but does not receive; S's window expansion is server-internal and not part of view_R.

In Step 8, for each of the $T + K$ rounds R receives BFV ciphertexts that decrypt to the per-partition polynomial

evaluations $f_{t,i,p}(b_j)$ for every (i, j, p). These plaintext reconstructions are the informative messages Sim_R must emulate, and they are the locus of the RSE (soundness) term of the simulator gap.

In Steps 9–10, R combines the decrypted evaluations into the per-pair reconstructions $\mathsf{KR}(i, j, t)$ (the two-point Lagrange combination of Step 9, instantiated at $k = 2$) and outputs labels. R receives no further messages; the work is local to its view.

For sender privacy, R must learn nothing about non-matching database rows beyond the ideal-world output. The key observation is that for any non-matching pair (i, j), under the PRF security of AES_κ and the threshold-$k = 2$ property of Shamir secret sharing, the real $\mathsf{KR}(i, j, t)$ is uniform on $\mathbb{F}_F$; the sole exception is the RSE composition event where a non-match collides to zero in all T token rounds, bounded by Lemma 1. This exception is the soundness term of the simulator gap, not a leak: on it the value the receiver reconstructs is an off-curve field element independent of the stored records (a two-point Lagrange value at $x = 0$ with the token channel constrained to 0 and the record channel free), so even there no sender data is revealed and the defect is one of soundness.

*b) **Constructing the receiver's simulator.**:* We mirror the real view step by step.

For Steps 1–5, Sim_R does nothing (no messages to R).

For Step 6, Sim_R invokes the semi-honest GC simulator [11] on receiver-side input Y and a freshly sampled simulated output $\widetilde{b}_j \xleftarrow{\$} \mathbb{F}_F$ for each $j \in [N]$, yielding a simulated GC transcript together with simulated blinded items $\{\widetilde{b}_j\}$ used downstream.

For Step 7, Sim_R generates the BSGS power bundle exactly as R would on inputs $\{\widetilde{b}_j\}$ using fresh BFV encryption randomness.

For Step 8, Sim_R populates the decrypted evaluations $\mathsf{KR}(i, j, t)$ using the ideal-world output: for each unordered pair $\{i, j\}$, each partition p, and each round t,

- if $\mathcal{F}_{\mathrm{CSTPSI}}$ reports a match at $(p, \{i, j\})$, set $\mathsf{KR}(i, j, t) = 0$ in every token round $t \in [0, T)$ and to the corresponding Lagrange-encoded label chunk for every label round $t \in [T, T + K)$, consistent with the reported label;
- otherwise sample $\mathsf{KR}(i, j, t) \xleftarrow{\$} \mathbb{F}_F$ independently across t.

Sim_R then constructs simulated BFV ciphertexts encrypting plaintext polynomials whose batched coefficients decrypt to these values under R's public key, using fresh encryption randomness; by Proposition 1 the per-round ciphertexts can be simulated independently across the $T + K$ rounds.

For Steps 9–10, R does only local computation; Sim_R runs the same computation on the simulated values, which by construction yields the ideal-world output.

*c) **Indistinguishability**:* We argue step by step to show the indistinguishability of $\mathsf{Real}_{\Pi_T, R}$ and $\mathsf{Ideal}_{\mathcal{F}_{\mathrm{CSTPSI}}, \mathsf{Sim}_R, R}$.

Steps 1–5. R receives nothing; the two views are identical.

Step 6. The real GC transcript on inputs (Y, κ) is computationally indistinguishable from the simulated transcript by the semi-honest GC simulator [11]. The real outputs $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ are PRF-uniform on $\mathbb{F}_F$ under the AES PRF assumption, up to negligible mod-F bias; the simulated $\widetilde{b}_j$ are exactly uniform. The gap on this step is bounded by the GC simulator advantage plus the AES PRF advantage of the single GC execution, amortized across all rounds by Proposition 2; the lemma's PRF hybrid charges $T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}$ in total across the T token rounds.

Step 7. R sends; the sent ciphertexts in both worlds are generated from the (then-fixed) blinded items under fresh BFV encryption randomness. Conditioned on Step 6 indistinguishability, the two bundles are identically distributed up to BFV randomness.

Step 8. The substantive gap. We split by pair type.

For matching pairs $(p, \{i, j\})$ consistent with the ideal-world output, the real reconstructions $\mathsf{KR}(i, j, t)$ are determined by the Shamir scheme: $s_{t,e} = 0$ for token rounds and the label chunk for label rounds. These match Sim_R's planted values exactly.

For non-matching pairs $(p, \{i, j\})$, the real $\mathsf{KR}(i, j, t)$ at every round is uniform on $\mathbb{F}_F$ conditioned on R's ideal-world output (by the threshold-$k = 2$ property of Shamir secret sharing combined with PRF-uniformity of $\{b_j\}$ from Step 6); the simulated values are uniform by construction. We argue this uniformity marginally per non-matching pair, which suffices for the semi-honest sketch in the style of FLPSI'21 [7]; the within-partition correlations among the $\binom{N}{k}$ reconstructions that share one low-degree interpolant are not needed for the bound and are not exploited by the simulator. The two distributions agree on non-matches except on the RSE composition event where the real reconstruction collides to zero across all T token rounds at some non-matching pair (mis-classified by R as a token candidate). The probability of this event is at most $\binom{N}{k} n_{part}/F^T$: the per-trial single-round collision probability $1/F$ is driven down to $1/F^T$ over the T independent token rounds (Theorem 2), and this is then union-bounded over the $\binom{N}{k} n_{part}$ trials per query (Lemma 1).

The BFV ciphertexts carrying these values are computationally indistinguishable from fresh encryptions by IND-CPA security of BFV [10]. By Proposition 1, the ciphertext bundle is transmitted once and reused, so the ciphertext-level gap is a single IND-CPA advantage rather than $T + K$ of them.

Steps 9–10. Local computation on values that are now indistinguishable; the outputs agree by construction.

Summing the per-step gaps, the receiver-side simulator gap is

$$\binom{N}{k} \frac{n_{part}}{F^T} \; + \; T \cdot \mathsf{Adv}^{\mathrm{prf}}_{\mathrm{AES}}(\mathcal{B}) \; + \; \mathsf{negl}(\lambda),$$

where the negligible term collects the GC simulator advantage and the BFV IND-CPA advantage. At the parameters of Corollary 1, the soundness term is below the engineering RSE threshold for $T \geq 2$ at million-scale and below the cryptographic standard for $T \geq 3$ up to ten-million scale. It reaches the cryptographic standard at billion-scale only at $T = 4$ (Corollary 1).

B. Receiver Privacy: Simulating the Sender

Sim_S takes as input $(\lambda, S, Db, |Y|)$: the security parameter, the corrupted party label, the sender's database, and the leakage $|Y|$ (which equals N and is public). $\mathcal{F}_{\mathrm{CSTPSI}}$ returns $\perp$ to S, so no ideal-world output enters the simulator.

a) ***Describing the sender's view.:*** We walk through the steps of CSTPSI protocol Π_T (Figure 1) to describe view_S.

Steps 1–5 are sender-internal: S computes on its own inputs and randomness; no messages arrive. Sim_S follows the protocol honestly on its own input Db; no simulation work is required.

In Step 6, S participates in the 1-GC AES execution with input κ and output $\perp$. S receives GC messages whose transcript is computationally indistinguishable from a simulated transcript by the semi-honest GC simulator [11]. By Proposition 2, the single GC transcript is identical in distribution to the marginal of $T + K$ independent GC executions on the same inputs, so this analysis covers the full session.

In Step 7, S receives the receiver's encrypted BSGS sparse window $\{\mathrm{Enc}(b_j^\ell)\}_{j,\,\ell\in W}$: $|W| \cdot N$ BFV ciphertexts under R's public key. This is the only ciphertext message S receives in the entire session. S's subsequent window expansion and per-round homomorphic evaluations are server-internal: S produces the response ciphertexts but does not receive them, so they are not part of view_S. By Proposition 1, the receiver's ciphertext bundle is transmitted once and reused across the $T+K$ rounds; the sender-side ciphertext-level gap is therefore a single IND-CPA advantage rather than $T + K$ of them.

In Steps 8 and 9–10, S either acts internally (Step 8) or is not involved (reconstruction is receiver-local). view_S gains no new messages.

b) ***Constructing the sender's simulator.:*** We again mirror the real view step by step.

For Steps 1–5, Sim_S runs the protocol honestly on the sender's own input Db; the resulting offline state is exactly what S would compute in the real world.

For Step 6, Sim_S invokes the semi-honest GC simulator [11] on sender-side input κ and output $\perp$, producing a simulated GC transcript.

For Step 7, Sim_S samples $|W| \cdot N$ random plaintexts in $\mathbb{F}_F$ of the appropriate batched length and encrypts them under fresh BFV randomness with R's public key, producing a simulated query bundle.

For Steps 8 and 9–10, Sim_S does nothing additional: no messages are received by S in these steps.

c) ***Indistinguishability:*** We argue step by step to show the indistinguishability of $\mathsf{Real}_{\Pi_T,S}$ and $\mathsf{Ideal}_{\mathcal{F}_{\mathrm{CSTPSI}},\mathsf{Sim}_S,S}$.

Steps 1–5. Honest execution on the sender's own input matches the real view exactly.

Step 6. Computational indistinguishability of the GC transcript follows from the semi-honest GC simulator guarantee [11]. By Proposition 2, this covers the full $(T+K)$-round session at the cost of one GC simulator invocation.

Step 7. Any distinguisher between the real query-power bundle and the simulated random-plaintext bundle yields a BFV IND-CPA distinguisher of the same advantage [10], since both bundles consist of BFV ciphertexts under R's public key and differ only in their plaintexts. By Proposition 1, the bundle is transmitted once, so a single IND-CPA reduction suffices for the entire session.

Steps 8 and 9–10. No messages enter view_S; the two views agree trivially.

The sender-side simulator gap reduces to the BFV IND-CPA advantage, the semi-honest GC simulator advantage, and the information-theoretic threshold property of Shamir secret sharing [13], summing to $\mathsf{negl}(\lambda)$.

C. Supporting Propositions

The two propositions below are invoked from inside both step-walks above.

Proposition 1 (Multi-round security)**.** *The transcript of the $T + K$ homomorphic-evaluation rounds in Π_T leaks no more, up to a negligible advantage, than what a sender or receiver would learn from $T + K$ independent semi-honest executions of the single-token-round configuration Π_1.*

Argument. The receiver's encrypted query powers are transmitted exactly once at the start of the online phase (Step 7) and reused as ciphertext operands in every subsequent round (Step 8). Their inclusion in the sender's view contributes only to the first round; subsequent rounds' use of the same ciphertext bundle adds nothing under the BFV IND-CPA assumption applied to the single transmission. The Shamir polynomials $\{P_{t,e}\}_t$ for distinct t (Step 4) are sampled with independent fresh coefficients (Theorem 2, conditional on a cryptographically secure pseudorandom number generator (CSPRNG) as the randomness source); the joint distribution of $\{f_{t,i,p}\}_t$ therefore factors as the product of per-round distributions, and the receiver's view across rounds factors correspondingly into independent single-round views. Sender-side, no per-round ciphertext is added to view_S, so no additional simulation gap arises. □ □

Proposition 2 (1-GC transcript identity)**.** *The transcript of the single GC execution in Step 6 of Π_T is identical in distribution to the transcript of $T + K$ independent GC executions on the same input $(y_1, \ldots, y_N)$ and key κ. Hence the optimization is transcript-equivalent to the naive $(T+K)$-GC composition and inherits its semi-honest security without additional reductions.*

Argument. The blinded query items $b_j = \mathrm{AES}_\kappa(y_j) \bmod F$ with the $0 \mapsto 1$ remap are deterministic in $(y_1, \ldots, y_N, \kappa)$; evaluating the same circuit on the same inputs yields identical outputs whether executed once or $T + K$ times. The single-execution transcript is therefore a marginal of the multi-execution distribution and is simulated by the standard semi-honest GC simulator. □ □

APPENDIX C
PROTOCOL WALKTHROUGH

Figure 5 gives a message-sequence view of CSTPSI (Π_T, Fig. 1) across the database, sender, and receiver.

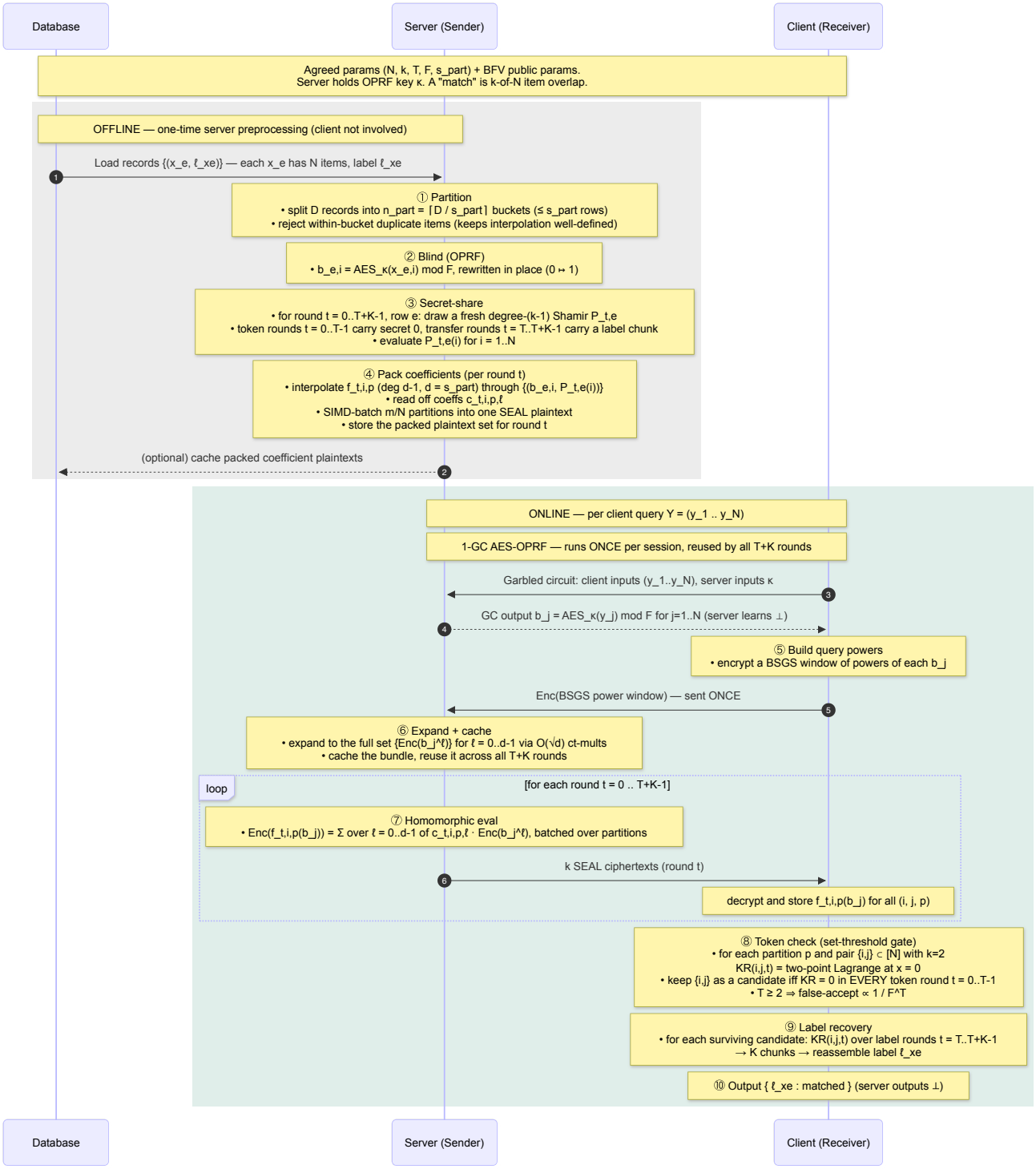

Fig. 5: Message-sequence walkthrough of CSTPSI (Π_T, Fig. 1) with all optimizations enabled and $T \geq 2$. **Offline** (grey) is sender-only and one-time: partition, OPRF-blind, secret-share over $T + K$ rounds, and pack the interpolated coefficients into SIMD-batched SEAL plaintexts. **Online** (teal) amortizes a single garbled circuit and a single cached query-power transmission across all $T + K$ rounds; only the per-round homomorphic evaluation repeats. A pair is admitted only if it reconstructs the token in *every* one of the T token rounds, yielding the $\propto 1/F^T$ realization soundness error (RSE) bound of Lemma 1.