AICR · OPEN PROBLEM

An Optimal Two-Dimensional Locality-Preserving Hash for Shifts

One printed algorithm, one experimentally observed error rate, and a matching lower bound — and no analysis

Status. Statement: AI-written from Elette Boyle, Itai Dinur, Niv Gilboa, Yuval Ishai, Nathan Keller and Ohad Klein, *Locality-Preserving Hashing for Shifts with Connections to Cryptography*, IACR ePrint 2022/028; ITCS 2022. Not yet checked by a human. Open, and asserted rather than asked: the source prints the algorithm, conjectures its error rate, reports experiments agreeing with the conjecture, states that it could not analyse it, and says settling it is left for future work. The matching lower bound is the source's own Theorem 1.4, so a proof would close the two-dimensional case exactly.

Category. *Foundations*, *Information-Theoretic*.

Conjecture (informal). *Two parties each hold a huge random array, and one array is the other shifted by one step along some axis. Neither may read more than d of the entries, they cannot talk, and each must output a position, such that the two positions differ by exactly the shift. The one-dimensional version of this is well understood, and its optimal error probability is $\widetilde{O}(1/d^2)$; the two-dimensional version is not. The source paper proves that no two-dimensional scheme can have error below $\Omega(1/d)$, gives one achieving $\widetilde{O}(d^{-7/8})$, and then prints a different, symmetric algorithm — a sequence of deterministic random walks over the plane with geometrically growing step sizes — whose error its experiments put at $\widetilde{O}(1/d)$, matching the lower bound. It could not prove that. This conjecture is that the printed algorithm really does achieve it.*

1 The setting

What an LPHS is for. A locality-preserving hash for shifts turns a long random string into a single position, so robustly that shifting the input by one moves the output by exactly one. The reason cryptographers care is a two-way connection, proved in the source's

Section 3, between LPHS and the *distributed discrete logarithm* problem in the generic group model: two parties holding v and $g \cdot v$ must non-interactively output shares differing by the known amount, which is the bottleneck step in group-based homomorphic secret sharing. An LPHS is exactly that problem with the group replaced by a random string, and the source's reduction moves algorithms and lower bounds in both directions. The k-dimensional generalization, where the input is indexed by $\mathbb{Z}_n^k$ and each axis shift must move the output by the corresponding unit vector, buys packed homomorphic secret sharing and a sublinear-time location-sensitive encryption, and is where the open question sits.

What is known in each dimension. In one dimension the answer is tight: $\delta = \widetilde{O}(d^{-2})$ is achievable, from the Iterated Random Walk algorithm of Dinur, Keller and Klein via the DDL connection, and $\Omega(d^{-2})$ is necessary. In k dimensions the source proves the lower bound $\delta = \Omega(d^{-2/k})$ (its Theorem 1.4, quoted as Theorem 1), which for $k = 2$ reads $\Omega(d^{-1})$, and gives a sequence of upper bounds: $\widetilde{O}(d^{-1/2})$ by two-dimensional MinHash, $\widetilde{O}(d^{-2/3})$ by combining MinHash on one axis with the one-dimensional optimum on the other, $\widetilde{O}(d^{-4/5})$ by applying the one-dimensional optimum on both axes, and $\widetilde{O}(d^{-7/8})$ by a three-stage algorithm whose analysis uses martingale techniques. None of these is $\widetilde{O}(d^{-1})$, and the source is explicit that the asymmetric treatment of the two axes is what costs it: its algorithms "do not have optimal dependence of δ on d".

The algorithm this page is about, and what the source says of it. Section 4.1.4 of the source, titled "Conjectured optimal algorithm", prints RANDOM-WALK-HASH (Definition 2) and states: "We conjecture that the following symmetric algorithm (RANDOM-WALK-HASH) has the optimal performance of $\delta = \widetilde{O}(1/d)$. However, we were not able to rigorously analyze it." The introduction is more explicit about the status: "we present another 2-dimensional LPHS algorithm, which seems harder to analyze, but for which we conjecture that the error rate is at most $\widetilde{O}(d^{-1})$", that "Our experiments suggest that the error rate of this algorithm is indeed" that, and that "the analysis (and especially deterministic resolution of cycles in the random walk) is quite involved, and settling our conjecture is left

open for future work". It also records what a proof would mean: "This bound is essentially the best one can hope for given the lower bound discussed below."

Why it resists analysis, in the source's own account. The heuristic the source would like to run is short. Model each RW-STAGE as a random walk on $\mathbb{Z}^2$ with independent steps uniform in $\{-L,\dots,L\}$ on each axis; assume that once the two parties' walks collide they stay collided; then the ℓ-th stage, counting MIN-HASH as the first, fails with probability about $d'^{-2^{-\ell}}$, and the product over the $I = \lg\lg d$ stages is $2^{O(I)}/d' = \widetilde{O}(1/d)$. The modelling assumption is what fails: "In practice we cannot guarantee independence, since the random walk occasionally runs into loops." Lines 8–15 of Algorithm RW-STAGE are the canonical loop-breaking rule, and they are what makes the steps neither independent nor deterministic functions of fresh input symbols. The source names the fix it would want and cannot have: "If the algorithm would make monotone queries along (at least) one axis (as the one-dimensional algorithm), then it would avoid loops and its analysis would be much simpler. Unfortunately, we do not know how to design such an algorithm with similar performance."

2 Notation and parameters

- n is the side length, Σ_b an alphabet of b-bit symbols, and the input is $x \in \Sigma_b^{\mathbb{Z}_n^2}$, a two-dimensional array with entries indexed by $\mathbb{Z}_n^2$ and drawn uniformly at random.
- $x \lll (r_1, r_2)$ is the cyclic shift of x by (r_1, r_2); e_1, e_2 are the unit vectors of $\mathbb{Z}_n^2$.
- d bounds the number of entries either party may query, and δ the error probability. $\widetilde{O}$ hides factors polylogarithmic in d.
- Integer operations are floored, and $\sqrt{\cdot}$ means the floor of the square root, as in the source.

Definition 1 (Two-dimensional LPHS, the source's Definition 2.1). A function $h : \Sigma_b^{\mathbb{Z}_n^2} \to \mathbb{Z}_n^2$ is a *2-dimensional* (d,δ)-*LPHS* if h can be computed by making d adaptive queries of

the form $x[i,j]$ to its input, and for each unit vector e_i,

$$\Pr_x\left[\, h(x) \neq h(x \lll e_i) + e_i \,\right] \leq \delta,$$

where x is uniform over $\Sigma_b^{\mathbb{Z}_n^2}$ and the same randomness of h is used in both invocations.

Theorem 1 (The source's Theorem 1.4, at $k = 2$). *For $n = \Omega(d)$, every 2-dimensional (d,δ)-LPHS satisfies $\delta = \Omega(d^{-1})$.*

Definition 2 (The source's Algorithms 1, 7 and 8). MIN-HASH(x,d) returns $\arg\min_{i,j\in[0,\sqrt{d}]}\{x[i,j]\}$.

RW-STAGE(z,d,L,i,j), for $L \in \mathbb{N}$ and a starting point $(i,j) \in \mathbb{Z}_n^2$, first fixes independent random functions $\psi_1,\psi_2 : \Sigma_b \to \{-L,\ldots,L\}$ and then:

1. $P \leftarrow$ list();
2. for $s \leftarrow 0,\ldots,d-1$: set $P[s] \leftarrow (i,j)$, let $v \leftarrow z[i,j]$, and set $(i,j) \leftarrow (i+\psi_1(v),\ j+\psi_2(v))$; then, if $(i,j) \in P$, let t be the unique index with $P[t] = (i,j)$, set $k \leftarrow \arg\min_{u\in[t,s]}\{z[P[u]]\}$ and $(i,j) \leftarrow P[k]$, and while $(i,j) \in P$ increment j;
3. return $\arg\min_{(i',j')\in P}\{z[i',j']\}$.

RANDOM-WALK-HASH(z,d) sets $I \leftarrow \lg\lg(d)$ and $d' \leftarrow d/I$, then

$$\begin{aligned}
(i_0,j_0) &\leftarrow \text{MIN-HASH}(z,d'),\\
(i_1,j_1) &\leftarrow \text{RW-STAGE}(z,d',d'^{1/4},i_0,j_0),\\
(i_2,j_2) &\leftarrow \text{RW-STAGE}(z,d',d'^{3/8},i_1,j_1),\\
(i_3,j_3) &\leftarrow \text{RW-STAGE}(z,d',d'^{7/16},i_2,j_2),\\
&\vdots\\
(i_I,j_I) &\leftarrow \text{RW-STAGE}\big(z,d',\sqrt{d'/2},i_{I-1},j_{I-1}\big),
\end{aligned}$$

and returns (i_I,j_I). The step size at stage ℓ is $d'^{1/2-2^{-(\ell+1)}}$, which is $d'^{1/4}, d'^{3/8}, d'^{7/16},\ldots$ as printed and tends to $\sqrt{d'}$ from below as ℓ grows.

3 The conjecture

Conjecture 1 (RANDOM-WALK-HASH is an optimal two-dimensional LPHS). *There is a function $c(\cdot)$, polylogarithmic in its argument, such that for all large enough n and b, the algorithm RANDOM-WALK-HASH$(\cdot, d)$ of Definition 2 is a 2-dimensional (d,δ)-LPHS (Definition 1) with*

$$\delta \;\le\; \frac{c(d)}{d}.$$

Remark 1 (What the statement does and does not fix). Two parameters are left as "large enough" because the source leaves them so: its statement of the conjecture names no condition on n or b, while the neighbouring results carry hypotheses of the shape $n = \Omega(d)$ and $2^b \ge d^4$ (Lemma 4.3) or $b \ge 3\log n$ (Lemma 4.1). A reviewer should pin these down before attempting a proof; they are the kind of hypothesis that an analysis supplies rather than assumes, and no reading of them changes what the conjecture is about. Note also that the claim is about *this* algorithm, not about the existence of some $\widetilde{O}(1/d)$ two-dimensional LPHS: the existence version is weaker, is the natural thing to want, and is not what the source conjectures.

Remark 2 (The two claims a proof would go through). The source reduces its heuristic to two facts about random walks on $\mathbb{Z}^2$ with steps uniform in $\{-L,\dots,L\}$ per axis, which it states and does not prove: writing T for the meeting time of two such walks started at ℓ_1 distance D, that $\mathbb{E}[\min(d', T)] = O(\sqrt{d'}\,(L + D/L))$; and that the expected ℓ_1 distance between the start and final point of such a walk of d' steps is $O(L\sqrt{d'})$. Granting those and the independence assumption, the stage-by-stage failure probabilities are $O(d'^{-1/4})$, $O(d'^{-1/8})$, $O(d'^{-1/16})$,…, whose product over $I = \lg\lg d$ stages is $\widetilde{O}(1/d)$. So the conjecture is not short of a strategy; it is short of a way to run that strategy against an algorithm whose steps are deterministic functions of the input it has already read, and whose loop-breaking rule correlates them.

Remark 3 (Refuting it is a different kind of work from proving it). Because Theorem 1 already forbids anything below $\Omega(1/d)$, a

refutation cannot come from a better lower bound on LPHS in general — it has to be a lower bound on *this algorithm*, showing that the loop-breaking rule or the correlation between stages costs a power of d. The experiments the source reports are evidence against that, at whatever sizes it ran them; the paper does not say what those sizes were, which is itself worth checking before trusting the direction.

4 Bibliography

[BDGIKK22] Elette Boyle, Itai Dinur, Niv Gilboa, Yuval Ishai, Nathan Keller and Ohad Klein. *Locality-preserving hashing for shifts with connections to cryptography.* IACR ePrint 2022/028; ITCS 2022, LIPIcs 215, article 27. The source. The conjecture and the algorithm are Section 4.1.4, PDF p. 24–25; the introduction's statement of it and the remark that settling it is left open are on printed p. 6–7 (PDF p. 7); the LPHS definition is Definition 2.1, PDF p. 9; the lower bound is Theorem 1.4, printed p. 7 (PDF p. 7), proved for $k = 2$ in Section 4.2; Min-Hash is Algorithm 1, PDF p. 17.

[DKK18] Itai Dinur, Nathan Keller and Ohad Klein. *An optimal distributed discrete log protocol with applications to homomorphic secret sharing.* CRYPTO 2018, Part III, pp. 213–242. Cited by the source as [23]; the Iterated Random Walk algorithm whose LPHS translation gives the optimal one-dimensional bound $\delta = \widetilde{O}(d^{-2})$, and the negative results the one-dimensional lower bound comes from.