Efficient Information-Theoretic Distributed Point Function with General Output Groups

Junru Li[1], Pengzhen Ke[2], Liang Feng Zhang[2*]

[1]Institute of Mathematical Science, Shanghaitech University, Middle Huaxia Road, Shanghai, 201210, China.

[2*]School of Information Science and Technology, Shanghaitech University, Middle Huaxia Road, Shanghai, 201210, China.

*Corresponding author(s). E-mail(s): zhanglf@shanghaitech.edu.cn;
Contributing authors: lijr2@shanghaitech.edu.cn;
kepzh@shanghaitech.edu.cn;

Abstract

An n-server information-theoretic *Distributed Point Function* (DPF) allows a client to secret-share a point function $f_{\alpha,\beta}(x)$ with domain $[N]$ and output group $\mathbb{G}$ among n servers such that each server learns no information about the function from its share (called a *key*) but can compute an additive share of $f_{\alpha,\beta}(x)$ for any x. DPFs with small key sizes and general output groups are preferred. In this paper, we propose a new transformation from share conversions to information-theoretic DPFs. By applying it to share conversions from Efremenko's PIR and Dvir-Gopi PIR, we obtain both an 8-server DPF with key size $O(2^{10}\sqrt{\log N \log\log N} + \log p)$ and output group $\mathbb{Z}_p$ and a 4-server DPF with key size $O(\tau \cdot 2^{6}\sqrt{\log N \log\log N})$ and output group $\mathbb{Z}_{2^\tau}$. The former allows us to partially answer an open question by Boyle, Gilboa, Ishai, and Kolobov (ITC 2022) and the latter allows us to build the first DPFs that may take any finite Abelian groups as output groups. We also discuss how to further reduce the key sizes by using different PIR, how to reduce the number of servers by resorting to statistical security or using nice integers, and how to obtain DPFs with t-security. We show the applications of the new DPFs by constructing new efficient PIR protocols with result verification.

Keywords: Distributed point function, Private information retrieval, Secret sharing, Information-theoretic cryptography

1 Introduction

An n-server *Distributed Point Function* (DPF) [1] converts any point function $f_{\alpha,\beta}$ (i.e., $f_{\alpha,\beta}(x) = \beta$ for $x = \alpha$ and $f_{\alpha,\beta}(x) = 0$ for all $x \neq \alpha$) into n shares $k_0, \ldots, k_{n-1}$ such that every share k_i leaks no information about the function but enables the computation of an additive share of $f_{\alpha,\beta}(x)$. In particular, both the additive shares and $f_{\alpha,\beta}(x)$ belong to an Abelian group, which is called the *output group*. The communication efficiency of a DPF may be measured by its *key size*, i.e., the maximum size of the n shares $k_0, \ldots, k_{n-1}$. Ideally, we would like DPFs with general output groups, small key sizes, and a small number of servers.

DPFs can be computational or information-theoretic. Computational DPFs [1] base their security on cryptographic assumptions (e.g., the existence of one-way functions) such that every share k_i leaks no information about the point function to a polynomial-time server. Information-theoretic DPFs [2] can tolerate any computationally unbounded server and have better computation efficiency. These merits make them especially useful in constructing efficient cryptographic protocols such as private information retrieval (PIR) [3].

The study of information-theoretic DPFs was initiated by Boyle et al. [2]. For point functions with domain $[N]$ and output group $\mathbb{Z}_{p^\tau}$ ($p \geq 3$ is a prime, $\tau \geq 1$), they constructed a 4-server *perfectly secure* DPF with key size $O(\tau \log(p) \cdot 2^{2p\sqrt{\log N \log\log N}})$; for point functions with domain $[N]$ and output group $\mathbb{Z}_p$ ($p \geq 2$ is a prime), they constructed a 3-server *statistically secure* DPF with key size $O(\log(p) \cdot 2^{2p\sqrt{\log N \log\log N}})$. Both DPFs were based on share conversions [4–6], which may be derived from the PIR of [7].

Note that the DPFs of [2] have several restrictions. First, their key sizes are all *exponential* in p. When p is large, they will incur unaffordable communication overhead. In fact, Boyle et al. [2] leave it as an *open question* to remove this exponential dependence of key sizes in p. In particular, it is even not known how to reduce the p in the exponent to $\mathsf{poly}(\log p)$. Second, the DPFs of [2] cannot handle point functions with an output group of the form $\mathbb{Z}_{2^\tau}$ for any $\tau > 1$. Consequently, it is impossible for Boyle et al. [2] to handle point functions with *any* finite Abelian group as output group. In many real-life applications, either an output group of the form $\mathbb{Z}_{2^\tau}$ with $\tau > 1$ (e.g., PIR with result verification [8]) or an output group of the form $\mathbb{Z}_p$ with a very large p (e.g., statistical analysis [9, 10]) is needed. Therefore, it is interesting to lift the above restrictions with new techniques.

1.1 Our Contributions

In this paper, we focus on the open question raised by Boyle et al. [2] and construct new DPFs with either smaller key sizes or more general output groups.

As the first contribution, we extend the definition of 1-private n-server DPFs of [2] that requires every key k_i leaks no information about a point function to that of t-private n-server DPFs that can tolerate the collusion of any t servers ($t \geq 1$). We then give a general transformation from share conversions that satisfy certain nice properties to perfectly secure DPFs. This transformation is novel and of independent

interest. In particular, we give a t-private DPF transformed from Woodruff-Yekhanin PIR [11].

As the second contribution, we build a share conversion from the matching vectors (MVs) based PIR scheme of [12] and apply the transformation to the share conversion to obtain a perfectly secure 8-server DPF with key size $O(2^{10\sqrt{\log N \log\log N}} + \log p)$, for point functions with domain $[N]$ and output group $\mathbb{Z}_p$ ($p \geq 2$ is a prime). We then adopt the idea of [2] to this DPF and construct a $2^{-\Omega(\lambda)}$-statistically secure 4-server DPF with key size $O(\lambda \cdot 2^{10\sqrt{\log N \log\log N}} + \lambda \log p)$, for the same point functions. These DPFs remove the p from the exponent and partially answer the open question of [2]. We also extend these constructions with the nice integers from [13, 14] and give both n-server ($n \leq 2^{r+1}$) perfectly secure DPFs and n-server ($n \leq 2^r$) statistically secure DPFs with key size $O(2^{c(r)\sqrt[r]{\log N (\log\log N)^{r-1}}} + \log p)$, for point functions with domain $[N]$ and output group $\mathbb{Z}_p$, where $r \geq 2$ and $c(r)$ is a constant. Since the set of functions we need to share is of size $N(p-1)+1$, the key size should be at least $\Theta(\log N + \log p)$. Our construction almost reach the optimal key size when p is quite larger than N. For applications like secure aggregation [9] or secure writing [15, 16], we need a DPF with a large prime output group, our new constructions greatly improved the efficiency.

As the third contribution, we build a share conversion from the MVs based PIR scheme of [7] and apply the transformation to the share conversion to obtain a perfectly secure 4-server DPF with key size $O(\tau \log p \cdot 2^{c(p)\sqrt{\log N \log\log N}})$, for point functions with domain $[N]$ and output group $\mathbb{Z}_{p^\tau}$, where $c(p) = 6$ for $p = 2$ and $c(p) = 2p$ for $p \geq 3$. In particular, for $p = 2$ and $\tau > 1$, our DPFs fill the gap left by Boyle et al. [2] and thus lead to DPFs with any finite Abelian groups as output groups. In the problem of private set intersection [17], each element is given a weight and the DPF is used to sum the weights of the elements in the intersection. In this application, we may assign weight 2^j to the j-th element. Choosing $\mathbb{Z}_{2^\tau}$ as the output group can reduce the storage cost. Besides, DPFs with arbitrary output groups are more flexible building blocks for constructing function secret sharing schemes [10, 18].

1.2 Application to PIR-RV

A t-private n-server PIR protocol allows one to privately retrieve an item DB_i of a database $\mathrm{DB} = (\mathrm{DB}_1, \ldots, \mathrm{DB}_N)$ from n servers, each of which stores a copy of DB, such that the collusion of any t servers learn no information about $i \in [N]$. Such a protocol is said to be a (v, ϵ)-secure PIR with result verification (PIR-RV) [8] if it additionally allows one to verify if the correct value of DB_i has been recovered, except with a small probability ϵ, when at most v of the servers provide wrong answers.

As the fourth contribution of this paper, we construct a 1-private $2(\zeta+1)$-server $(2\zeta, \frac{1}{2^\tau})$-secure PIR-RV protocol with communication complexity $O(\zeta^3\tau \cdot 2^{6\sqrt{\log N \log\log N}})$ for any positive integers ζ and τ together with a 1-private $4(\zeta+1)$-server $(4\zeta, \frac{1}{p})$-secure PIR-RV protocol with communication complexity $O(\zeta^5 \cdot 2^{6\sqrt{\log N \log\log N}} + \zeta^5 \log p)$, by using DPFs. Compared with the 2-server $(1, \frac{1}{p})$-secure PIR-RV [8] with communication complexity $O(\log p \cdot \sqrt{N})$, ours support more servers and is asymptotically more efficient. Our construction is the most efficient information-theoretic PIR-RV protocols to date and secure even if a majority of the servers are

malicious, which is a property not achieved by [8]. Our PIR-RV is constructed from any perfect secure DPF in our DPF framework, so future improvements in DPF could lead to improvements in PIR-RV protocols, which is also meaningful.

1.3 Our Techniques

A share conversion allows one to convert the shares of a secret under a secret sharing scheme (SSS) to the shares of a *related* secret under another SSS. Our transformation requires a share conversion $\mathsf{Conv}(\cdot,\cdot,\cdot)$ from a (t,n)-threshold SSS to an additive SSS. Given a point function $f_{\alpha,\beta}(x)$, we secret-share α as $(\mathbf{c}_0,\ldots,\mathbf{c}_{n-1})$ with the threshold SSS. The function Conv is chosen such that there exist functions ϕ,Φ and ψ (where ϕ is a homomorphism and Φ is bilinear) with the following property: for any x, there exists a value σ such that

$$f_{\alpha,\beta}(x) = \phi\left(\Phi\left(\big(\sigma\cdot\beta\big)\diamond\psi(\alpha), \sum_{\ell=0}^{n-1}\mathsf{Conv}(\ell,x,\mathbf{c}_\ell)\right)\right), \tag{1}$$

where $\diamond$ stands for the action of $\sigma\cdot\beta$ on a module element $\psi(\alpha)$. By splitting $(\sigma\cdot\beta)\diamond\psi(\alpha)$ as the sum of $t+1$ random values $h_0,\ldots,h_t$ and distributing every $(h_j,\mathbf{c}_\ell)$ to a different server, the algebraic properties of ϕ and Φ allow us to express $f_{\alpha,\beta}(x)$ as the sum of $n(t+1)$ terms, each of which can be computed by exactly one of the servers. Consequently, this transformation gives a t-private $n(t+1)$-server perfectly secure DPF. Underlying the DPFs of Boyle et al. [2] is a formula similar to (1), which however lacks the component ϕ. In our language, their transformation is a special case of ours with ϕ being the identity function. The idea of introducing ϕ to (1) is the core technique of this work.

Like [2], our 4-server perfectly secure DPF with output group $\mathbb{Z}_{p^\tau}$ is based on a share conversion from the PIR [7]. The share conversion of [2] needs MVs over $\mathbb{Z}_m$ for $m=2p^\tau$. As MVs exist only if m has ≥ 2 different prime divisors, their construction cannot allow $p=2$. A natural idea is to change m to qp^τ for a prime $q\neq p$. However, this simple idea turns out not working. We bypass this difficulty by replacing the ring $\mathbb{Z}_{2p^\tau}$ with $\mathbb{Z}_{p^\tau}[\gamma]/(\gamma^q-1)$ and applying a homomorphism ϕ from this ring to $\mathbb{Z}_{p^\tau}$, which gives (1). For the 8-server perfectly secure DPF with output group $\mathbb{Z}_p$, we use a share conversion from the PIR [12], which can use MVs over a much smaller ring $\mathbb{Z}_m$ and in particular allow us to remove the exponential dependence in p of the key size.

1.4 Organization

In Section 2, we introduce basic notion, definitions and techniques that will be used in our constructions. Section 3 presents our transformation from share conversion to information-theoretic DPFs. In Section 4, we build share conversions on several existing PIR schemes and apply the transformation to obtain our new DPFs. In Section 5, we show the applications of our DPFs to PIR-RV protocols. Finally, Section 6 contains our concluding remarks.

2 Preliminaries

Let $\mathbb{Z}^+$ the set of all positive integers. For any $N \in \mathbb{Z}^+$, we denote $[N] = \{1, \ldots, N\}$. For any $m, h \in \mathbb{Z}^+$, we denote by $\mathbb{Z}_m$ the ring of integers modulo m and denote by $\mathbb{Z}_m^h$ the set of all vectors of length h over $\mathbb{Z}_m$. For any $\mathbf{u} = (u_1, \ldots, u_h), \mathbf{v} = (v_1, \ldots, v_h) \in \mathbb{Z}_m^h$, we denote $\langle \mathbf{u}, \mathbf{v} \rangle_m = \sum_{i=1}^h u_i v_i$. For any prime power q, we denote by $\mathbb{F}_q$ the finite field of q elements and denote by $\mathbb{F}_q^*$ its multiplicative group. Let $\mathbf{u} = (u_1, \ldots, u_h)$. For any vector $\mathbf{z} = (z_1, \ldots, z_h)$, we denote $\mathbf{z}^{\mathbf{u}} = z_1^{u_1} \cdots z_h^{u_h}$. For any γ, we denote $\gamma^{\mathbf{u}} = (\gamma^{u_1}, \ldots, \gamma^{u_h})$. We use $\delta_{\alpha,x}$ to denote the Kronecker symbol, i.e., $\delta_{\alpha,x} = 1$ when $x = \alpha$ and $\delta_{\alpha,x} = 0$ when $\alpha \neq x$.

Bilinear functions. Let $\mathbb{R}$ a commutative ring with identity. Let $\mathbb{H}$ be an $\mathbb{R}$-module (see Section 2.1 for basics about rings and modules). We denote by $r \diamond h$ the action of a ring element $r \in \mathbb{R}$ on a module element $h \in \mathbb{H}$. Let $\mathcal{C}$ be a finite Abelian group. A function $\Phi : \mathbb{H} \times \mathcal{C} \to \mathbb{R}$ is said to be *bilinear* if for any $h_1, h_2 \in \mathbb{H}, \mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}, r_1, r_2 \in \mathbb{R}$, $\Phi(r_1 \diamond h_1 + r_2 \diamond h_2, \mathbf{c}_1 + \mathbf{c}_2) = \sum_{i=1}^2 r_i \cdot \Phi(h_i, \mathbf{c}_j)$.

Probability. We denote by U_ℓ the uniform distribution over $\{0,1\}^\ell$. For any two distributions D_1, D_2 over the same sample space Ω, we denote by $\mathsf{SD}(D_1, D_2) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr_{D_1}[\omega] - \Pr_{D_2}[\omega]|$ their *statistical distance*.

Point functions. Let $N \in \mathbb{Z}^+$ and let $\mathbb{G}$ be an Abelian group. For any $\alpha \in [N]$ and $\beta \in \mathbb{G}$, the *point function* $f_{\alpha,\beta}$: $[N] \to \mathbb{G}$ is defined by $f_{\alpha,\beta}(x) = \beta \cdot \delta_{\alpha,x}$.

2.1 Rings, Modules and the Structure of Finite Abelian groups

Definition 1. (Commutative ring with identity [19]) *A commutative ring $\mathbb{R}$ with identity is a set together with two binary operations $+$ and $\cdot$ satisfying the following axioms:*

1. *$(\mathbb{R}, +)$ is an Abelian group, we call this group the additive group of $\mathbb{R}$;*
2. *$(a \cdot b) \cdot c = a \cdot (b \cdot c)$ and $a \cdot b = b \cdot a$ for all $a, b, c \in \mathbb{R}$;*
3. *$(a + b) \cdot c = (a \cdot c) + (b \cdot c)$;*
4. *there is an element $1 \in \mathbb{R}$ with $1 \cdot a = a \cdot 1 = a$ for all $a \in \mathbb{R}$.*

Definition 2. (Module [19]) *Let $\mathbb{R}$ be a commutative ring with identity. A (left) $\mathbb{R}$-module is an Abelian group $\mathbb{H}$ with an action of $\mathbb{R}$ on $\mathbb{H}$ denoted by $r \diamond h$, for all $r \in \mathbb{R}$ and $h \in \mathbb{H}$ which satisfies*

1. *$(r + s) \diamond h = r \diamond h + s \diamond h$ for all $r, s \in \mathbb{R}$ and $h \in \mathbb{H}$;*
2. *$(r \cdot s) \diamond h = r \cdot (s \diamond h)$ for all $r, s \in \mathbb{R}$ and $h \in \mathbb{H}$;*
3. *$r \diamond (h_1 + h_2) = r \diamond h_1 + r \diamond h_2$ for all $r \in \mathbb{R}$ and $h_1, h_2 \in \mathbb{H}$;*
4. *$1 \diamond h = h$ for all $h \in \mathbb{H}$.*

In particular, let m be a positive integer, $\mathbb{R}$ is a commutative ring with identity, $\mathbb{G} = \mathbb{Z}_m$ be a subgroup of the additive group of $\mathbb{R}$ with $[1]_m \in \mathbb{G}$ is also the 1 element of $\mathbb{R}$. Note that $\mathbb{G} = \mathbb{Z}_m$ could also be regarded as a ring.

Theorem 1. *Let ϕ be a surjective (group) homomorphism from the additive group of $\mathbb{R}$ to $\mathbb{G}$ that fix $\mathbb{G} = \mathbb{Z}_m \subseteq \mathbb{R}$. Then for all $r \in \mathbb{R}$ and $\beta \in \mathbb{G}$ we have:*

$$\phi(r \cdot [b]_m) = [b]_m \diamond \phi(r)$$

Proof. By the 3rd axiom of a ring, we have

$$\phi(r \cdot \beta) = \phi(\underbrace{r + r + \cdots + r}_{b})$$

for $\beta = [b]_m \in \mathbb{Z}_m$. Since ϕ is a Abelian group homormophism,

$$\phi(\underbrace{r + r + \cdots + r}_{b}) = \underbrace{\phi(r) + \cdots + \phi(r)}_{b}$$

which is equal to $\beta \cdot \phi(r)$ when we regard $\mathbb{G}$ as ring $\mathbb{Z}_m$. □

There is theorem about the structure of finite generated modules over a principal ideal domain in [19], which could imply the structure of finite Abelian groups.

Theorem 2. (Structure of finite Abelian groups) *For any finite Abelian group $\mathbb{G}$, there exist primes $p_1, \ldots, p_\ell$ and positive integers $\tau_1, \ldots, \tau_\ell$ such that:*

$$\mathbb{G} \cong \mathbb{Z}/(p_1^{\tau_1}) \times \cdots \times \mathbb{Z}/(p_\ell^{\tau_\ell}).$$

2.2 Distributed Point Function

Informally, a t-private n-server DPF [2] allows one to secret-share a point function $f_{\alpha,\beta}$ among n servers such that any t servers learn no information about the function. However, given any input $x \in [N]$, each server can compute an additive share of $f_{\alpha,\beta}(x) \in \mathbb{G}$.

Definition 3. (Distributed point function) *An n-server* DPF $\Pi = (\mathsf{Gen}, \{\mathsf{Eval}_i\}_{i=0}^{n-1})$ *is a tuple of $n+1$ algorithms with the following syntax:*

- $(k_0, \ldots, k_{n-1}) \leftarrow \mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$*: Given a security parameter λ and a point function $f_{\alpha,\beta}$, the (randomized)* key generation algorithm Gen *returns n secret keys $k_0, \ldots, k_{n-1}$.*
- $y_i \leftarrow \mathsf{Eval}_i(k_i, x)$*: Give a secret key k_i and an input $x \in [N]$, the (deterministic)* evaluation algorithm Eval_i *(of server i) returns a group element $y_i \in \mathbb{G}$.*

The protocol Π should satisfy the following requirements:

- **Correctness.** *For any λ, any $f_{\alpha,\beta}$, any $x \in [N]$, and any $(k_0, \ldots, k_{n-1}) \leftarrow \mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$,* $\Pr\left[\sum_{i=0}^{n-1} \mathsf{Eval}_i(k_i, x) = f_{\alpha,\beta}(x)\right] = 1$.
- **Security.** *The security of a t-private DPF requires that every $\leq t$ secret keys leak no information about the point function. Formally, we consider the following security experiment between a challenger and an adversary $\mathcal{A}$ that controls the j-*th *server for $j \in T$ ($T \subseteq \{0, 1, \ldots, n-1\}, |T| \leq t$):*

– *Given the security parameter* λ, $\mathcal{A}$ *generates two point functions* $f^0 = f_{\alpha_0,\beta_0}$ *and* $f^1 = f_{\alpha_1,\beta_1}$, *both having domain* $[N]$ *and range* $\mathbb{G}$.
– *The challenger samples* $b \xleftarrow{\$} \{0,1\}$ *uniformly, generates* n *secret keys* $(k_0, \ldots, k_{n-1}) \leftarrow \mathsf{Gen}(1^\lambda, f^b)$ *for the point function* f^b, *and gives* $k_T = \{k_i : i \in T\}$ *to* $\mathcal{A}$.
– *The adversary* $\mathcal{A}$ *outputs a guess* $b' \leftarrow \mathcal{A}(k_T)$.

Denote by $\mathrm{Adv}(1^\lambda, \mathcal{A}, T) := |\Pr[b = b'] - 1/2|$ *the* advantage *of* $\mathcal{A}$ *in guessing* b *in the experiment. For a circuit size bound* $M = M(\lambda)$ *and an advantage bound* $\epsilon = \epsilon(\lambda)$, *we say that* Π *is* (M, ϵ)-secure *if for all subset* $T \subseteq \{0, \ldots, n-1\}$ *of cardinality* $\leq t$, *and all non-uniform adversaries* $\mathcal{A}$ *of size* $M(\lambda)$, $\mathrm{Adv}(1^\lambda, \mathcal{A}, T) \leq \epsilon(\lambda)$.

A DPF is said to be *statistically* ϵ*-secure* if it is (M, ϵ)-secure for all M, and *perfectly secure* if it is statistically 0-secure. Both kinds of DPFs are called *information-theoretic* DPFs [2]. In this work, we focus on information-theoretic DPFs.

2.3 Private Information Retrieval

A t-private n-server PIR protocol involves two kinds of participants: a *client* and n *servers* $\mathcal{S}_0, \cdots, \mathcal{S}_{n-1}$, where each server has a database $\mathrm{DB} \in \{0,1\}^N$ and the client has an index $\alpha \in [N]$. It allows the client to retrieve DB_α, without revealing α to any t of the servers.

Definition 4 (Private information retrieval). *An* n*-server* PIR $\Gamma = (\mathsf{Que}, \mathsf{Ans}, \mathsf{Rec})$ *is a triple of algorithms with the following syntax:*

- $(\{\mathsf{que}_j\}_{j=0}^{n-1}, \mathsf{aux}) \leftarrow \mathsf{Que}(N, \alpha)$*: This is a randomized* querying algorithm *for the client. Given a retrieval index* $\alpha \in [N]$, *it outputs* n *queries* $\{\mathsf{que}_j\}_{j=0}^{n-1}$, *along with an auxiliary information* aux. *For each* $0 \leq j < n$, *the query* que_j *will be sent to the server* $\mathcal{S}_j$. *The auxiliary information* aux *will be used by the client in the reconstructing algorithm.*
- $\mathsf{ans}_j \leftarrow \mathsf{Ans}(\mathrm{DB}, \mathsf{que}_j)$*: This is a deterministic* answering algorithm *for the server* $\mathcal{S}_j$ $(0 \leq j < n)$. *Given the database* DB *and the query* que_j, *it outputs an answer* ans_j.
- $\mathrm{DB}_\alpha \leftarrow \mathsf{Rec}(\alpha, \{\mathsf{ans}_j\}_{j=0}^{n-1}, \mathsf{aux})$*: This is a deterministic* reconstructing algorithm *for the client. Given the the retrieval index* α, *the answers* $\{\mathsf{ans}_j\}_{j=0}^{n-1}$ *and the auxiliary information* aux, *it outputs* DB_α.

The protocol Γ *should satisfy the following requirements:*

- **Correctness.** *For any* N, *any* $\mathrm{DB} \in \{0,1\}^N$, *any* $\alpha \in [N]$, *and any* $(\{\mathsf{que}_j\}_{j=0}^{n-1}, \mathsf{aux}) \leftarrow \mathsf{Que}(N, \alpha)$, *it holds that* $\mathsf{Rec}(\alpha, \{\mathsf{Ans}(\mathrm{DB}, \mathsf{que}_j)\}_{j=0}^{n-1}, \mathsf{aux}) = \mathrm{DB}_\alpha$.
- t**-Privacy.** *For any* N, *any* $\alpha_1, \alpha_2 \in [N]$, *and any* $T \subseteq \{0, 1, \ldots, n-1\}$ *with* $|T| \leq t$, $\mathsf{Que}_T(N, \alpha_1)$ *and* $\mathsf{Que}_T(N, \alpha_2)$ *are identically distributed, where* Que_T *denotes the concatenation of the* j-th *output of* Que *for all* $j \in T$.

The efficiency of an n-server PIR protocol is measured by its *communication complexity*, which is denoted by $\mathsf{CC}_\Gamma(N)$ and defined as the number of bits communicated

between the client and all servers, maximized over the choices of $\mathrm{DB} \in \{0,1\}^N$ and $\alpha \in [N]$, i.e., $\mathsf{CC}_\Gamma(N) = \max_{\mathrm{DB},\alpha}(\sum_{j=0}^{n-1}(|\mathsf{que}_j| + |\mathsf{ans}_j|))$.

2.4 Secret Sharing and Share Conversion

In Section 3, we will propose a general transformation from PIR to information-theoretic DPF. A stepping stone in this transformation is share conversion, which converts one SSS into another.

Definition 5. (Secret sharing [4, 20]) *An* SSS $\mathcal{L} = (\mathsf{Share}, \mathsf{Recov})$ *for* n *participants allows a dealer to convert a secret* $s \in \mathcal{S}$ *into* n *shares* $(\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}) \leftarrow \mathsf{Share}(s)$*, one to each participant, such that*

- *Any authorized set* $A \subseteq \{0, 1, \ldots, n-1\}$ *of participants can reconstruct the secret* s *by executing the reconstruction algorithm on their shares, i.e.,* $s \leftarrow \mathsf{Recov}(\{\mathbf{c}_j\}_{j\in A})$*;*
- *Any unauthorized set* $B \subseteq \{0, 1, \ldots, n-1\}$ *learns no information about* s*, i.e., for any* $s_1, s_2 \in \mathcal{S}$*,* $\mathsf{Share}_B(s_1)$ *and* $\mathsf{Share}_B(s_2)$ *are identically distributed.*

For ease of exposition, we denote an SSS by $(\mathcal{L}, \mathcal{S})$. An SSS $(\mathcal{L}, \mathcal{S})$ is called a (t, n)-*threshold SSS* if the authorized sets are the subsets of $\{0, 1, \ldots, n-1\}$ of cardinality $\leq t$, and called an *additive* SSS if $s = \mathbf{c}_0 + \cdots + \mathbf{c}_{n-1}$ for all $(\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}) \leftarrow \mathsf{Share}(s)$. We say that $(\mathcal{L}, \mathcal{S})$ has *share space* $\mathcal{C}$ if for any $s \in \mathcal{S}$, the n shares output by $\mathsf{Share}(s)$ all belong to $\mathcal{C}$.

Definition 6. (Share conversion [4]) *Let* $(\mathcal{L}_1, \mathcal{S}_1) = ((\mathsf{Share}_1, \mathsf{Recov}_1), \mathcal{S}_1)$ *and* $(\mathcal{L}_2, \mathcal{S}_2)$ *be two SSSs. Let* $R \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ *be a binary relation such that, for every* $s_1 \in \mathcal{S}_1$ *there exists at least one* $s_2 \in \mathcal{S}_2$ *such that* $(s_1, s_2) \in R$*. We say that* $\mathcal{L}_1$ *is* locally convertible *to* $\mathcal{L}_2$ *w.r.t.* R *if there exist local* share conversion functions $(g_0, \ldots, g_{n-1})$ *with the following property:* For any $s_1 \in \mathcal{S}_1$ and $(\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}) \leftarrow \mathsf{Share}_1(s_1)$, $\big(g_0(\mathbf{c}_0), \ldots, g_{n-1}(\mathbf{c}_{n-1})\big)$ is a valid sharing for some $s_2 \in \mathcal{S}_2$ such that $(s_1, s_2) \in R$.

2.5 Matching Vector Families

Our DPFs are constructed with matching vector (MV) families [12], which also underlie the most efficient PIR schemes [7, 12, 21] to date.

Definition 7. (S-matching family) *Let* $m, h \in \mathbb{Z}^+$ *and let* $S \subseteq \mathbb{Z}_m \setminus \{0\}$*. A pair* (U, V)*, where* $U = \{\mathbf{u}_x\}_{x=1}^N, V = \{\mathbf{v}_x\}_{x=1}^N \subseteq \mathbb{Z}_m^h$*, is said to be an* S-matching family *of size* N *if* $\langle \mathbf{u}_\alpha, \mathbf{v}_\alpha \rangle_m = 0$ *for all* $\alpha \in [N]$*, and* $\langle \mathbf{u}_x, \mathbf{v}_\alpha \rangle_m \in S$ *for all* $x, \alpha \in [N]$ *such that* $x \neq \alpha$*.*

Efremenko [12] defined MV families and gave the first superpolynomial size S-matching families modulo a composite integer m, where $S \subseteq \mathbb{Z}_m \setminus \{0\}$ was the canonical set [13] of m.

Definition 8. (Canonical set) *Let* $m = p_1^{e_1} \ldots p_r^{e_r} > 1$*, where* $p_1, \ldots, p_r$ *are* $r > 1$ *distinct primes and* $e_1, \ldots, e_r \in \mathbb{Z}^+$*. The* canonical set of m*, denoted by* S_m*, is the set of integers* $\sigma \in \mathbb{Z}_m \setminus \{0\}$ *such that* $\sigma \bmod p_i^{e_i} \in \{0, 1\}$ *for all* $i \in [r]$*.*

The S_m-matching families of Efremenko [12] are obtained from the superpolynomial size set systems of Grolmusz [22].

Theorem 3. **([12, 22])** *Let $m = p_1^{e_1} \cdots p_r^{e_r} > 1$, where $p_1, \ldots, p_r$ are $r > 1$ distinct primes and $e_1, \ldots, e_r \in \mathbb{Z}^+$. Then there is a constant $c = c(m)$ such that: for any integer $h > 0$, there is an S_m-matching family (U, V) of size N in $\mathbb{Z}_m^h$ such that $h = O\big(2^{c\sqrt[r]{\log N (\log\log N)^{r-1}}}\big)$.*

For $r = 2$, the constant c in Theorem 3 may be taken as $2 \cdot \max\{p_1, p_2\}$. In the PIR schemes of [7, 12], the S_m-matching family (U, V) of Theorem 3 was used to encode any database $\mathrm{DB} = (\mathrm{DB}_1, \ldots, \mathrm{DB}_N) \in \{0,1\}^N$ as

$$F_{\mathrm{DB}}(\mathbf{z}) = \sum_{j=1}^{N} \mathrm{DB}_j \cdot \mathbf{z}^{\mathbf{u}_j}, \tag{2}$$

a polynomial in $\mathbf{z} = (z_1, \ldots, z_h)$, such that the problem of privately retrieving a database entry DB_i is reduced to the problem of privately recovering a coefficient of F_{DB}. In particular, F_{DB} may be interpreted as a polynomial over a finite field [12] or a finite ring [7].

2.6 Efremenko's PIR

Let q be a prime power such that $q-1$ is a multiple of the integer m from Theorem 3. Then the finite field $\mathbb{F}_q$ contains an element γ of multiplicative order m. In Efremenko [12], the $F_{\mathrm{DB}}(\mathbf{z})$ in (2) was interpreted as a polynomial in $\mathbb{F}_q[\mathbf{z}]$ and any DB_α was recovered by considering the restriction of $F_{\mathrm{DB}}(\mathbf{z})$ on a random multiplicative line in G^h, where $G = \langle\gamma\rangle$. The recovering procedure is based on an S_m-decoding polynomial [12].

Definition 9. **(S-decoding polynomial)** *Let $m \in \mathbb{Z}^+$ and let $S \subseteq \mathbb{Z}_m \setminus \{0\}$. Let q be a prime power such that $m|(q-1)$ and let $\gamma \in \mathbb{F}_q^*$ be of multiplicative order m. A polynomial $P(x) \in \mathbb{F}_q[x]$ is called an* S_m-decoding polynomial *if $P(\gamma^\sigma) = 0$ for all $\sigma \in S$, and $P(\gamma^0) = 1$.*

For any $S \subseteq \mathbb{Z}_m \setminus \{0\}$, a trivial construction may give an S-decoding polynomial

$$P(X) = \prod_{\sigma \in S} (X - \gamma^\sigma) \Big/ \prod_{\sigma \in S} (1 - \gamma^\sigma) \tag{3}$$

with at most $n = |S| + 1$ monomials, e.g., $P(X) = a_0 X^{b_0} + \cdots + a_{n-1} X^{b_{n-1}}$. To retrieve any DB_α, Efremenko [12] requires one to communicate with n servers, choose a random vector $\mathbf{w} \leftarrow \mathbb{Z}_m^h$, send to the j-th server $\mathbf{w} + b_j \mathbf{v}_\alpha$ for all $0 \le j < n$, and finally output

$$\mathrm{DB}_\alpha = \sum_{j=0}^{n-1} a_j \cdot F_{\mathrm{DB}}(\gamma^{\mathbf{w} + b_j \mathbf{v}_\alpha}). \tag{4}$$

The number of monomials in $P(X)$ is equal to the number of required servers, which should be as small as possible. For a given (m, S, q, γ), the S-decoding polynomial in Definition 9 is *not unique*. For example, for $m = 511$ and $S = \{1, 147, 365\}$, Efremenko

[12] showed an S-decoding polynomial with 3 ($< |S| + 1$) monomials. Itol and Suzuki [13] showed a composition theorem for finding S_m-decoding polynomials with fewer monomials.

Theorem 4. (Composition theorem [13]) *Let $m = m_1 m_2$ be the product of two coprime integers m_1 and m_2. If there is an S_{m_i}-decoding polynomial with n_i monomials for $i = 1, 2$, then there is an S_m-decoding polynomial with n monomials such that $n \leq n_1 n_2$.*

Chee et al. [14] showed that if $m = p_1 p_2$ is a Mersenne number, then there is an S_m-decoding polynomial with 3 monomials. Such m is *nice* in the sense that the number of monomials in an S_m-decoding polynomial can be strictly smaller than $|S_m| + 1 = 4$. The nice integers [14] gave the most efficient n-server PIR schemes to date for all $n \geq 27$.

2.7 Generalized Dvir-Gopi PIR

Dvir and Gopi [7] constructed an MV-based 2-server PIR with communication complexity $\exp(O(\sqrt{\log N (\log \log N)}))$. In their PIR, the $F_{\mathrm{DB}}(\mathbf{z})$ in (2) is regarded as a polynomial over the finite ring $\mathbb{Z}_m[\gamma]/(\gamma^m - 1)$. Each server uses not only $F_{\mathrm{DB}}(\mathbf{z})$ but also the following vector-valued function to answer PIR queries:

$$F^{(1)}_{\mathrm{DB}}(\mathbf{z}) = \sum_{j=1}^{N} \mathrm{DB}_j \cdot \mathbf{u}_j \cdot \mathbf{z}^{\mathbf{u}_j}. \tag{5}$$

Boyle et al. [2] generalized [7] such that the reconstruction algorithm computes a linear combination of the servers' answers. They chose $m = 2p^\tau$ for an odd prime p. We give a more general version by choosing $m = qp^\tau$, where p, q are distinct primes. We regard $F_{\mathrm{DB}}(\mathbf{z})$ and $F^{(1)}_{\mathrm{DB}}(\mathbf{z})$ as functions from $\mathbb{R}^h$ to $\mathbb{Z}_{p^\tau}$, where $\mathbb{R} = \mathbb{Z}_{p^\tau}[\gamma]/(\gamma^q - 1)$. To retrieve DB_i, the client interacts with two servers, chooses a random vector $\mathbf{w} \leftarrow \mathbb{Z}_m^h$ and sends to the ℓ-th server $\mathbf{c}_\ell = \mathbf{w} + \ell \cdot \mathbf{v}_i$ for $\ell = 0, 1$, where $\mathbf{v}_i$ is from the set V in an MV family (U, V). The ℓ-th server replies with

$$\mathbf{a}_\ell = (-1)^\ell \gamma^{1-\ell} \left(F_{\mathrm{DB}}(\gamma^{\mathbf{c}_\ell}), F^{(1)}_{\mathrm{DB}}(\gamma^{\mathbf{c}_\ell}) \right) = \sum_{j=1}^{N} \mathrm{DB}_j \cdot (-1)^\ell \gamma^{1-\ell+\langle \mathbf{c}_\ell, \mathbf{u}_j \rangle_m} \cdot (1, \mathbf{u}_j) \tag{6}$$

which is in $\mathbb{R}^{h+1}$. Let $\phi(r) = r_1$ for $r = r_0 + r_1\gamma + \cdots + r_{q-1}\gamma^{q-1} \in \mathbb{R}$. Upon receiving $\mathbf{a}_0$ and $\mathbf{a}_1$ from the two servers, the user recovers DB_i with

$$\mathrm{DB}_i = \phi\left(\langle (\mathbf{a}_0 + \mathbf{a}_1), (1, -\mathbf{v}_i) \rangle_{p^\tau} \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_i \rangle_m} \right) \in \mathbb{Z}_{p^\tau}. \tag{7}$$

3 Our Transformation from Share Conversion to DPF

In this section, we show a transformation from share conversions that satisfy certain properties to DPFs. As many existing PIR protocols [7, 12] imply share conversions

with the properties, our transformation will give a method of constructing DPFs from PIR via share conversion, which will be used in Section 4 to obtain our new DPFs.

We construct perfectly secure DPFs for point functions with domain $[N]$ and range $\mathbb{G}$, where $N \in \mathbb{Z}^+$ and $\mathbb{G}$ is an Abelian group. From Theorem 2 we know that any finite Abelian group $\mathbb{G}$ is isomorphic to a group of form $\mathbb{Z}/(p_1^{\tau_1}) \times \cdots \times \mathbb{Z}/(p_\ell^{\tau_\ell})$. Suppose there are n-server DPFs $(\mathsf{DPF}_1, \ldots, \mathsf{DPF}_\ell)$ with $\mathsf{DPF}_j = (\mathsf{Gen}^j, \mathsf{Eval}_0^j, \ldots, \mathsf{Eval}_{n-1}^j)$ and output group $\mathbb{G}_j = \mathbb{Z}/(p_j^{\tau_j})$. Then it's enough to construct a DPF with output group $\mathbb{G}$ in the following figure.

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: For $\alpha \in [N], \beta = (\beta_1, \ldots, \beta_\ell) \in \mathbb{Z}/(p_1^{\tau_1}) \times \cdots \times \mathbb{Z}/(p_\ell^{\tau_\ell}) \cong \mathbb{G}$,
  - for $j = 1, \ldots, \ell$, $(k_0^j, \ldots, k_{n-1}^j) \leftarrow \mathsf{Gen}_j(1^\lambda, f_{\alpha,\beta_j})$;
  - output $(k_0, \ldots, k_{n-1})$ where $k_i = (k_i^1, \ldots, k_i^\ell)$.
- $\mathsf{Eval}_i(k_i, x)$: For $k_i = (k_i^1, \ldots, k_i^\ell), x = (x_1, \ldots, x_\ell) \in \mathbb{Z}/(p_1^{\tau_1}) \times \cdots \times \mathbb{Z}/(p_\ell^{\tau_\ell}) \cong \mathbb{G}$, output

$$(\mathsf{Eval}_i^1(k_i^1, x_1), \ldots, \mathsf{Eval}_i^\ell(k_i^\ell, x_\ell)) \in \mathbb{Z}/(p_1^{\tau_1}) \times \cdots \times \mathbb{Z}/(p_\ell^{\tau_\ell}) \cong \mathbb{G}.$$

Fig. 1 DPF with output group $\mathbb{G}$

Fig. 1 shows that we only need to consider DPFs with output groups of form $\mathbb{Z}/(p^\tau) = \mathbb{Z}_{p^\tau}$, then we can use them to construct DPFs with any output groups. The construction is essentially a transformation from share conversions with certain nice properties to DPFs.

Let $(\mathcal{L}_1, \mathcal{S}_1) = ((\mathsf{Share}_1, \mathsf{Recov}_1), [N])$ be a (t, n)-threshold SSS with share space $\mathcal{C}_1$. Let $(\mathcal{L}_2, \mathcal{S}_2)$ be an additive SSS with share space $\mathcal{C}_2$, where $\mathcal{C}_2 = \mathcal{S}_2$ is an additive group. Suppose that $R \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ is a binary relation and $(\mathcal{L}_1, \mathcal{S}_1)$ is locally convertible to $(\mathcal{L}_2, \mathcal{S}_2)$. To enable the proposed transformation, we require:

(a) There is a function $\mathsf{Conv} : \{0, 1, \ldots, n-1\} \times \mathcal{S}_1 \times \mathcal{C}_1 \to \mathcal{C}_2$ such that for any $x \in \mathcal{S}_1$, the functions $g_0^x, \ldots, g_{n-1}^x : \mathcal{C}_1 \to \mathcal{C}_2$ defined by

$$g_\ell^x(\mathbf{c}) = \mathsf{Conv}(\ell, x, \mathbf{c}), \qquad \forall 0 \le \ell < n,\ \mathbf{c} \in \mathcal{C}_1, \tag{8}$$

are n local share conversion functions for the binary relation R.

(b) There is a commutative ring $\mathbb{R}$ with identity such that $\mathbb{G} \subseteq \mathbb{R}$ is a subgroup of the additive group of $\mathbb{R}$, $\mathbb{G}$ contains the identity element of $\mathbb{R}$ and there is a surjective homomorphism $\phi : \mathbb{R} \to \mathbb{G}$.

(c) There exist an $\mathbb{R}$-module $\mathbb{H}$, a function $\psi : \mathcal{S}_1 \to \mathbb{H}$, and a bilinear function $\Phi : \mathbb{H} \times \mathcal{C}_2 \to \mathbb{R}$ such that: for any $\alpha \in \mathcal{S}_1$, any $(\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}) \leftarrow \mathsf{Share}_1(\alpha)$, any $x \in \mathcal{S}_1$, and

$$\rho(\alpha, x) := \Phi\left(\psi(\alpha), \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\right), \tag{9}$$

there exists a ring element $\sigma \in \mathbb{R}$ that satisfies

$$\phi(\rho(\alpha, x) \cdot \sigma) = \delta_{\alpha,x}. \tag{10}$$

Share conversion: the SSSs $(\mathcal{L}_1, \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$, the binary relation R, the function Conv, the ring $\mathbb{R}$, the module $\mathbb{H}$, and the functions ϕ, ψ, Φ that satisfy (a), (b) and (c).

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: Given $\alpha \in [N]$ and $\beta \in \mathbb{G}$, compute $(\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{n-1}) \leftarrow \mathsf{Share}_1(\alpha)$; choose $h_0, h_1, \ldots, h_t \leftarrow \mathbb{H}$ uniformly subject to

$$h_0 + h_1 + \cdots + h_t = (\sigma \cdot \beta) \diamond \psi(\alpha);$$

  output $n(t+1)$ keys $\{k_i\}_{i=0}^{n(t+1)-1}$, where $k_i = (h_j, \mathbf{c}_\ell)$ for $i = nj + \ell$ $(0 \le j \le t, 0 \le \ell < n)$.
- $\mathsf{Eval}_i(k_i, x)$: For $k_i = (h_j, \mathbf{c}_\ell)$ and $x \in [N]$, output $\phi\big(\Phi\big(h_j, \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\big)\big)$.

Fig. 2 Perfectly secure DPF Π from share conversion

Given the SSSs $(\mathcal{L}_1, \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$, the binary relation R, the function Conv, the ring $\mathbb{R}$, the module $\mathbb{H}$, and the functions ϕ, ψ, Φ that satisfy (a), (b) and (c), **Fig. 2** shows our construction of perfectly secure DPFs. For a point function $f_{\alpha,\beta}$ with domain $[N]$ and range $\mathbb{G}$, we secret-share α with the (t, n)-threshold SSS $(\mathcal{L}_1, \mathcal{S}_1)$ such that any $\le t$ shares leak no information about $f_{\alpha,\beta}$. For any $x \in [N]$, the outputs $\{\mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\}_{0 \le \ell < n}$ define a function $\rho(\alpha, x)$ in (9) that satisfies (10), where $\sigma \in \mathbb{R}$. We normalize the $\psi(\alpha)$ in (9) by acting the ring element $\sigma \cdot \beta$ in order to have that

$$f_{\alpha,\beta}(x) = \phi\left(\Phi\left((\sigma \cdot \beta) \diamond \psi(\alpha), \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\right)\right). \tag{11}$$

We additively secret-share $(\sigma \cdot \beta) \diamond \psi(\alpha) \in \mathbb{H}$ such that any $\le t$ shares leak no information about $f_{\alpha,\beta}$. In our construction, the $n(t+1)$ servers are organized as a $(t+1) \times n$ array, the (j, ℓ)-th server $(0 \le j \le t, 0 \le \ell < n)$ is given both the jth share of $(\sigma \cdot \beta) \diamond \psi(\alpha)$ and the ℓth share of α such that any $\le t$ servers learn no information about $f_{\alpha,\beta}$. The bilinear property of Φ allows us to distribute the computation of the left-hand side of (11) to the $n(t+1)$ servers and obtain a DPF $(\mathsf{Gen}, \mathsf{Eval}_0, \ldots, \mathsf{Eval}_{n(t+1)-1})$.

Theorem 5. *The construction of* **Fig. 2** *gives a t-private $n(t+1)$-server perfectly secure DPF with output group $\mathbb{G}$ $(= \mathbb{Z}_{p^\tau})$.*

Proof. We need to show that Π is correct and t-private. The correctness requires that for any $\alpha \in [N], \beta \in \mathbb{G}$ and $x \in [N]$, the sum of the $n(t+1)$ servers' outputs is equal

to $f_{\alpha,\beta}(x)$. Since Φ is bilinear, we have that

$$\begin{aligned}\Phi\Big(\sum_{j=0}^{t} h_j, \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\Big) &= \Phi\Big((\sigma\cdot\beta)\diamond\psi(\alpha), \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\Big) \\ &= (\sigma\cdot\beta)\cdot\Phi\Big(\psi(\alpha), \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\Big) = \sigma\cdot\beta\cdot\rho(\alpha, x).\end{aligned} \tag{12}$$

Note that $\beta \in \mathbb{G}$ is the residue class of b modulo p^τ for some integer $0 \le b < p^\tau$. In Section 2.1, we show that $\phi(r\cdot\beta) = \beta\cdot\phi(r)$ for any $r \in \mathbb{R}$ and $\beta \in \mathbb{G}$. By Eq. (10), we have that

$$\phi(\sigma\cdot\beta\cdot\rho(\alpha, x)) = \beta\cdot\phi(\rho(\alpha, x)\cdot\sigma) = \beta\cdot\delta_{\alpha,x} = f_{\alpha,\beta}(x). \tag{13}$$

Due to Eq. (12) and Eq. (13), we have that

$$\begin{aligned}\sum_{i=0}^{n(t+1)-1} \mathsf{Eval}_i(k_i, x) &= \sum_{j=0}^{t}\sum_{\ell=0}^{n-1} \phi\big(\Phi\big(h_j, \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\big)\big) \\ &= \phi\Big(\Phi\big(\sum_{j=0}^{t} h_j, \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell)\big)\Big) = f_{\alpha,\beta}(x).\end{aligned}$$

Regarding privacy, we note that $\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}$ are shares of α under the (t, n)-threshold SSS $(\mathcal{L}_1, \mathcal{S}_1)$, $h_0, \ldots, h_t$ are shares of $(\sigma\cdot\beta)\diamond\psi(\alpha)$ under a t-private additive SSS, and any $\le t$ servers learn $\le t$ of $\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}$ and $\le t$ of $h_0, \ldots, h_t$. It's easy to see that any $\le t$ servers learn no information about $f_{\alpha,\beta}$, i.e., Π is t-private. □

Statistically Secure DPFs. Boyle et al. [2] construct a 3-server statistically secure DPF using a share conversion from $(2, 3)$-CNF sharing to additive secret sharing. By choosing

$$\mathcal{S}_1 = [N],\ \mathbb{G} = \mathbb{Z}_p,\ \mathcal{S}_2 = \mathcal{C}_2 = \mathbb{R} = \mathbb{H} = \mathbb{F}_{p^\tau},\ \Phi(a, b) = a\cdot b, \tag{14}$$

Under this condition, we have

$$\sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell) \begin{cases} \neq 0, & x = \alpha \\ = 0, & x \neq \alpha \end{cases}. \tag{15}$$

Following the techniques in [2], we can generalize their construction to get more general statistically secure DPFs in **Fig. 3**, which is a t-private n-server $2^{-\Omega(\lambda)}$-statistically secure DPF with output group $\mathbb{Z}_p$.

Let Share be the Share_1 algorithm of SSS $(\mathcal{L}_1, \mathcal{S}_1)$, Conv be the algorithm given by Eq. (8) from the share conversion.

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: For $\alpha \in [N]$ and $\beta \in \mathbb{Z}_p$,
  - for $\xi = 1, \ldots, \lambda$ draw $\alpha^*_\xi \leftarrow \{\alpha, N+1\}$ at random and compute $(\mathbf{c}^\xi_0, \ldots, \mathbf{c}^\xi_{n-1}) \leftarrow \mathsf{Share}(\alpha^*_\xi)$;
  - for $\xi = 1, \ldots, \lambda$ set $\mathbf{y} = (y^1, \ldots, y^\lambda)$ as

$$y^\xi = \begin{cases} \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, \alpha, \mathbf{c}^\xi_\ell) & , \ \alpha^*_\xi = \alpha \\ 0 & , \ \alpha^*_\xi = N+1 \end{cases};$$

  - choose $\mathbf{r} \in \mathbb{F}^\lambda_{p^\tau}$ at random under the constraint that $\phi(\langle \mathbf{r}, \mathbf{y} \rangle) = \beta$;
  - output n keys $\{k_i\}_{i=0}^{n-1}$, where $k_i = ((\mathbf{c}^\xi_\ell)^\lambda_{\xi=1}, \mathbf{r})$ for $i = \ell$ $(\ell \in \{0, \ldots, n-1\})$.
- $\mathsf{Eval}_i(k_i, x)$: For $k_i = ((\mathbf{c}^\xi_\ell)^\lambda_{\xi=1}, \mathbf{r})$ and $x \in [N]$;
  - for $\xi = 1, \ldots, \lambda$ set $y^\xi_i = \mathsf{Conv}(\ell, x, \mathbf{c}^\xi_\ell)$, and denote by $\mathbf{y}_i \in \mathbb{F}^\lambda_{p^\tau}$ the vector of all y^ξ_i values concatenated;
  - output $\phi(\langle \mathbf{r}, \mathbf{y}_i \rangle)$.

Fig. 3 Statistically secure MV-based DPF framework

4 DPFs from Our Transformation

In this section, we construct new perfectly secure DPFs by instantiating the transformation from Section 3. We construct perfectly secure DPFs with output group $\mathbb{G} = \mathbb{Z}_{p^\tau}$ for any prime p and any integer $\tau \in \mathbb{Z}^+$. For $p = 2$ and $\tau > 1$, such DPFs are not known to exist before this work. These DPFs allow us to obtain perfectly secure DPFs with any finite Abelian groups as output groups. For any prime p and $\tau = 1$, we provide a DPF that supports colluding servers and an alternative construction of DPFs that have much shorter secret keys.

4.1 DPFs with Output Group $\mathbb{Z}_{p^\tau}$

In this section, we construct a perfectly secure 4-server DPF with output group $\mathbb{G} = \mathbb{Z}_{p^\tau}$, where p may be *any* prime and $\tau \in \mathbb{Z}^+$. Our DPFs are obtained by instantiating the transformation from Section 3. Underlying our construction is our *new* generalization of the Dvir-Gopi PIR [7] with $m = qp^\tau$ (see Section 2.7). Our choice of m only requires that p, q be different primes. In our language, Boyle et al. [2] is a special case of our generalization by fixing $q = 2$. It is this new choice of m that allows us to obtain DPFs with output group $\mathbb{G} = \mathbb{Z}_{2^\tau}$ (let $p = 2$ and q be an odd prime). In **Fig. 1**, we show that the techniques of this section enable the construction of DPFs with *any finite Abelian group* as output group.

To present the new DPFs with $\mathbb{G} = \mathbb{Z}_{p^\tau}$, we directly give the share conversion in our generalization of Dvir-Gopi PIR, and then apply the transformation from Section 3.

Share Conversion. Let $f_{\alpha,\beta}$ be a point function with domain $[N]$ and output group $\mathbb{G} = \mathbb{Z}_{p^\tau}$. We choose a prime $q \neq p$ and let

$$\mathbb{R} = \mathbb{Z}_{p^\tau}[\gamma]/(\gamma^q - 1) \tag{16}$$

be the ring of polynomials modulo $\gamma^q - 1$, with coefficients from $\mathbb{Z}_{p^\tau}$. In our share conversion, the SSSs $(\mathcal{L}_1 = (\mathsf{Share}_1, \mathsf{Recov}_1), \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$ are chosen such that

$$\mathcal{S}_1 = [N], \quad \mathcal{C}_1 = \mathbb{Z}_m^h, \quad \mathcal{S}_2 = \mathcal{C}_2 = \mathbb{R}^{h+1}, \tag{17}$$

where $m = qp^\tau$, $h \in \mathbb{Z}^+$ is an integer such that there is an S_m-matching family $(U, V) \subseteq \mathbb{Z}_m^h$ of size N, and $\mathcal{C}_1, \mathcal{C}_2$ are the share spaces of the two SSSs. For $\alpha \in \mathcal{S}_1$, $\mathsf{Share}_1(\alpha)$ generates two shares $\mathbf{c}_0, \mathbf{c}_1 \in \mathcal{C}_1$ by mapping α to a vector $\mathbf{v}_\alpha \in V$, randomly choosing $\mathbf{w} \leftarrow \mathbb{Z}_m^h$, and finally setting

$$\mathbf{c}_\ell = \mathbf{w} + \ell \cdot \mathbf{v}_\alpha, \ \ell = 0, 1. \tag{18}$$

Given $\mathbf{c}_\ell$ and any $x \in \mathcal{S}_1$, the local conversion function $\mathsf{Conv}(\ell, x, \mathbf{c}_\ell)$ is defined by

$$\mathsf{Conv}(\ell, x, \mathbf{c}_\ell) = (-1)^\ell \gamma^{1-\ell+\langle \mathbf{c}_\ell, \mathbf{u}_x \rangle_m} \cdot (1, \mathbf{u}_x), \tag{19}$$

where $\mathbf{u}_x \in U$ is the x-th element of U. The SSS $(\mathcal{L}_2, \mathcal{S}_2)$ is additive and may recover a value $\mathbf{s}_2(\mathbf{w}, \alpha, x) \in \mathcal{S}_2$ from the converted shares in Eq. (19) via

$$\mathbf{s}_2(\mathbf{w}, \alpha, x) = \sum_{\ell=0}^{1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell). \tag{20}$$

Eq. (20) gives a binary relation $R \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ that will be used in our transformation:

$$R = \{(\alpha, \mathbf{s}_2(\mathbf{w}, \alpha, x)) : \alpha, x \in \mathcal{S}_1, \mathbf{w} \in \mathbb{Z}_m^h\}. \tag{21}$$

From Share Conversion to DPF. Besides the ring $\mathbb{R}$, the SSSs $(\mathcal{L}_1, \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$, the binary relation R, and the local share conversion function Conv that satisfies the requirement of (a) in Section 3, we still need to properly choose a module $\mathbb{H}$ and three functions ϕ, ψ, Φ that satisfy (b) and (c), in order to apply our transformation. Note that $\mathbb{G}$ is a subgroup of the additive group of $\mathbb{R}$ and contains the identity element of $\mathbb{R}$. For any $r \in \mathbb{R}$, there exist q elements $r_0, \ldots, r_{q-1} \in \mathbb{G}$ such that $r = r_0 + r_1\gamma + \cdots + r_{q-1}\gamma^{q-1}$. In particular, the representation of r into the sum is always unique. We choose $\phi : \mathbb{R} \to \mathbb{G}$ such that

$$\phi(r) = r_1, \quad \forall r = r_0 + r_1\gamma + \cdots + r_{q-1}\gamma^{q-1} \in \mathbb{R}. \tag{22}$$

Then it is easy to see that ϕ is a surjective homomorphism and thus satisfies the requirement of (b). For (c), we choose the $\mathbb{R}$-module $\mathbb{H} = \mathbb{R}^{h+1} (= \mathcal{S}_2 = \mathcal{C}_2)$ and $\psi : \mathcal{S}_1 \to \mathbb{H}$ such that

$$\psi(\alpha) = (1, -\mathbf{v}_\alpha), \quad \forall \alpha \in \mathcal{S}_1. \tag{23}$$

Then it is easy to verify that the function $\Phi : \mathbb{H} \times \mathcal{C}_2 \to \mathbb{R}$ define by

$$\Phi(\mathbf{h}, \mathbf{c}) = \langle \mathbf{h}, \mathbf{c} \rangle, \quad \forall \mathbf{h} \in \mathbb{H} = \mathbb{R}^{h+1}, \quad \mathbf{c} \in \mathcal{C}_2 = \mathbb{R}^{h+1}. \tag{24}$$

is bilinear. For any $\alpha \in \mathcal{S}_1$, any $(\mathbf{c}_0, \mathbf{c}_1) \leftarrow \mathsf{Share}_1(\alpha)$, any $x \in \mathcal{S}_1$, Eq. (20), (23) and (24) jointly imply that the $\rho(\alpha, x)$ in Eq. (9) is

$$\rho(\alpha, x) = \langle (1, -\mathbf{v}_\alpha), \mathbf{s}_2(\mathbf{w}, \alpha, x) \rangle. \tag{25}$$

For the above choices of $\mathbf{w}, \alpha$, and x, we set

$$\sigma = \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}, \tag{26}$$

show that Eq. (10) is satisfied (see the proof for Theorem 6), and thus meet the requirement of (c). Applying our transformation from Section 3 with the related building blocks as above, we get the 4-server perfectly secure DPF (see **Fig. 4**).

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: Given $\alpha \in [N]$ and $\beta \in \mathbb{Z}_{p^\tau}$, generate $(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{w}, \mathbf{w} + \mathbf{v}_\alpha)$, choose $\mathbf{h}_0, \mathbf{h}_1 \leftarrow \mathbb{H} = \mathbb{R}^{h+1}$ uniformly subject to

$$\mathbf{h}_0 + \mathbf{h}_1 = \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m} \beta \diamond (1, -\mathbf{v}_\alpha),$$

  output $k_0 = (\mathbf{h}_0, \mathbf{c}_0), k_1 = (\mathbf{h}_0, \mathbf{c}_1), k_2 = (\mathbf{h}_1, \mathbf{c}_0), k_3 = (\mathbf{h}_1, \mathbf{c}_1)$.
- $\mathsf{Eval}_i(k_i, x)$: For every $i \in \{0, 1, 2, 3\}$, $k_i = (\mathbf{h}_j, \mathbf{c}_\ell)$ and $x \in [N]$, output

$$\phi\big(\langle \mathbf{h}_j, (-1)^\ell \gamma^{1-\ell+\langle \mathbf{c}_\ell, \mathbf{u}_x \rangle_m} \cdot (1, \mathbf{u}_x) \rangle\big).$$

Fig. 4 A 4-server perfectly secure DPF with output group $\mathbb{G} = \mathbb{Z}_{p^\tau}$

Theorem 6. *The construction of* **Fig. 4** *gives a perfectly secure 4-server DPF with output group $\mathbb{Z}_{p^\tau}$ (p is any prime, $\tau \in \mathbb{Z}^+$). For point functions with domain $[N]$, the key size of the DPF is $O(\tau \log(p) \cdot 2^{c(p)\sqrt{\log N \log\log N}})$, where $c(2) = 6$, $c(p) = 2p$ for $p \geq 3$.*

Proof. First of all, we show the correctness and security of that construction. Since we have showed that the requirement (a) and (b) in Section 3 is realized, it suffices to show that

$$\phi(\rho(\alpha, x) \cdot \sigma) = \delta_{\alpha,x}.$$

In this construction we have

$$\begin{aligned}\phi(\rho(\alpha,x)\cdot\sigma) &= \phi\left(\langle(1,-\mathbf{v}_\alpha),\sum_{\ell=0}^{1}\mathsf{Conv}(\ell,x,\mathbf{c}_\ell)\rangle\cdot\gamma^{-\langle\mathbf{w},\mathbf{u}_\alpha\rangle_m}\right)\\ &= \phi\left(\langle(1,-\mathbf{v}_\alpha),(\gamma^{1+\langle\mathbf{c}_0,\mathbf{u}_x\rangle_m}-\gamma^{\langle\mathbf{c}_1,\mathbf{u}_x\rangle_m})\cdot(1,\mathbf{u}_x)\rangle\cdot\gamma^{-\langle\mathbf{w},\mathbf{u}_\alpha\rangle_m}\right)\\ &= \phi\left(\gamma^{-\langle\mathbf{w},\mathbf{u}_\alpha\rangle_m}\cdot\gamma^{\langle\mathbf{w},\mathbf{u}_x\rangle_m}\cdot(\gamma-\gamma^{\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m})\cdot(1-\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle \bmod p^\tau)\right).\end{aligned}$$

- For $x=\alpha$, $\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m=0$, $\phi(\rho(\alpha,x)\cdot\sigma)=\phi(\gamma-1)=1$.
- For $x\neq\alpha$, $\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m\in\{1,\sigma_{01},\sigma_{10}\}$, where $\sigma_{01} \bmod p^\tau = 0$, $\sigma_{01} \bmod q = 1$, $\sigma_{10} \bmod p^\tau = 1$, $\sigma_{10} \bmod q = 0$:
  - If $\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m=1$ or $\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m=\sigma_{01}$, then $\gamma-\gamma^{\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m}=0$, hence $\phi(\rho(\alpha,x)\cdot\sigma)=0$;
  - If $\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m=\sigma_{10}$, then $1-\langle\mathbf{v}_\alpha,\mathbf{u}_x\rangle_m \bmod p^\tau=0$, hence $\phi(\rho(\alpha,x)\cdot\sigma)=0$.

Then the requirement (c) satisfies, then the correctness and security of $(\mathsf{Gen},\{\mathsf{Eval}_i\}_{0\le i<4})$ follows from Theorem 5.

Finally, we determine the key size in this construction. Each key k_i is in $\mathbb{H}\times\mathcal{C}_1=\mathbb{Z}_m\times\mathbb{R}^{h+1}$, whose size is $O(\tau\log p\cdot h)$. From Theorem 3 we know the key size $|k_i|=O(\tau\log(p)\cdot 2^{c(p)\sqrt{\log N\log\log N}})$, where $c(2)=6$, $c(p)=2p$ for $p\ge 3$ which completes the proof. □

4.2 DPFs with Output Group $\mathbb{Z}_p$

For $\mathbb{G}=\mathbb{Z}_p$, the DPFs from Section 4.1 have key sizes exponential in p. Boyle et al. [2] have statistically secure DPFs for the same output group. However, both schemes are only 1-private and the key sizes of both schemes are exponential in p as well. In Section 4.2.1, we show that how to obtain a DPF with $\mathbb{G}=\mathbb{Z}_p$ by applying our transformation to a share conversion from the PIR [11] and get a t-private information-theoretic DPF. In Section 4.2.2, we show how to get a DPF with key sizes only *linear* in $\log p$.

4.2.1 t-private DPF

Applying our transformation to Woodruff-Yekhanin PIR of [11], we obtain a t-private DPF with keys of size sublinear in the point function's domain. The construction is as follows.

Share conversion. Let $f_{\alpha,\beta}=(N,\mathbb{G},\alpha,\beta)$ be the point function that we want to share between two servers, where $\mathbb{G}=\mathbb{Z}_p$. We choose a positive integer τ and let

$$\mathbb{R}=\mathbb{F}_{p^\tau}. \tag{27}$$

In our share conversion, the SSSs $(\mathcal{L}_1=(\mathsf{Share}_1,\mathsf{Recov}_1),\mathcal{S}_1)$ and $(\mathcal{L}_2,\mathcal{S}_2)$ are chosen such that

$$\mathcal{S}_1=[N];\qquad \mathcal{C}_1=\mathbb{F}_{p^\tau}^h;\qquad \mathcal{S}_2=\mathcal{C}_2=\mathbb{F}_{p^\tau}^{h+1}; \tag{28}$$

where $\binom{h}{d} \geq N$, $d = \lfloor \frac{2n-1}{t} \rfloor$ and $p^\tau > n$. We first take $E : [N] \to \{0,1\}^h \subseteq \mathbb{F}_{p^\tau}^h$ to be an embedding of the N coordinates into points in $\{0,1\}^h$ of Hamming weight d. The SSS $(\mathcal{L}_1, \mathcal{S}_1)$ secret-shares any $s_1 = \alpha \in \mathcal{S}_1$ between n servers by firstly mapping α to a vector $E(\alpha)$, then randomly choosing $\mathbf{w} \leftarrow \mathbb{F}_{p^\tau}^h$, and finally setting

$$\mathbf{c}_\ell = E(\alpha) + \zeta_\ell \mathbf{w}, \ \forall \ell = 0, 1, \ldots, n-1 \tag{29}$$

where $\zeta_0, \zeta_1, \ldots, \zeta_{n-1} \in \mathbb{F}_{p^\tau}$ are distinct and nonzero. The ℓ-th server computes

$$F(\ell, x, \mathbf{c}_\ell) = \prod_{j : E(x)_j = 1} c_{\ell_j} \tag{30}$$

and its gradients

$$\nabla F(\ell, x, \mathbf{c}_\ell) := \left(\frac{\partial F}{\partial c_{\ell_0}}(\ell, x, \mathbf{c}_\ell), \ldots, \frac{\partial F}{\partial c_{\ell_{n-1}}}(\ell, x, \mathbf{c}_\ell) \right). \tag{31}$$

Then if we let $f(\zeta_\ell) := F(\ell, x, \mathbf{c}_\ell)$ be a degree d polynomial of ζ_ℓ, we have

$$f'(\zeta_\ell) = \langle \mathbf{w}, \nabla F(\ell, x, \mathbf{c}_\ell) \rangle. \tag{32}$$

If $f(\zeta) = a_0 + a_1 \zeta + \cdots + a_d \zeta^d$, we can get

$$a_0 = F(\ell, x, E(\alpha)) = \begin{cases} 0, & x \neq \alpha \\ 1, & x = \alpha \end{cases} \tag{33}$$

by solving the linear equation (sometimes we have $2n - 1 > d$, then we don't need all the $2n$ equations, but this doesn't matter)

$$\begin{bmatrix} f(\zeta_0) \\ f'(\zeta_0) \\ \vdots \\ f(\zeta_{n-1}) \\ f'(\zeta_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & \zeta_0 & \ldots & \zeta_0^d \\ 0 & 1 & \ldots & d\zeta_0^{d-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_{n-1} & \ldots & \zeta_{n-1}^d \\ 0 & 1 & \ldots & d\zeta_{n-1}^{d-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{d-1} \\ a_d \end{bmatrix} \tag{34}$$

Then we can denote the solution of a_0 by

$$a_0 = b_0 f(\zeta_0) + b_0' f'(\zeta_0) + \cdots + b_{n-1} f(\zeta_{n-1}) + b_{n-1}' f'(\zeta_{n-1}). \tag{35}$$

The local conversion function $\mathsf{Conv}(\ell, x, \mathbf{c}_\ell)$ is defined by

$$\mathsf{Conv}(\ell, x, \mathbf{c}_\ell) = \left(b_\ell F(\ell, x, \mathbf{c}_\ell), b_\ell' \nabla F(\ell, x, \mathbf{c}_\ell) \right). \tag{36}$$

The SSS $(\mathcal{L}_2, \mathcal{S}_2)$ recovers $\mathbf{s}_2 \in \mathcal{S}_2$ via computing

$$\mathbf{s}_2(\mathbf{w}, \alpha, x) = \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell) \tag{37}$$

Eq. (37) gives a binary relation $R \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ that will be used in our transformation:

$$R = \{(\alpha, \mathbf{s}_2(\mathbf{w}, \alpha, x)) : \alpha, x \in \mathcal{S}_1, \mathbf{w} \in \mathbb{F}_{p^\tau}^h\}. \tag{38}$$

For any $\alpha \in \mathcal{S}_1$, there exist exactly $N \cdot p^{h\tau}$ elements $s_2 \in \mathcal{S}_2$ such that $(\alpha, s_2) \in R$.

From share conversion to DPF. Besides the ring $\mathbb{R}$, the SSSs $(\mathcal{L}_1, \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$, the binary relation R, and the local share conversion function Conv that satisfies the requirement of (a) in Section 3, we still need to properly choose a module $\mathbb{H}$ and three functions ϕ, ψ, Φ that satisfy (b) and (c), in order to apply our transformation. Note that $\mathbb{G}$ is a subgroup of the additive group of $\mathbb{R}$ and shares the same identity element with $\mathbb{R}$. Let $\phi : \mathbb{F}_{p^\tau} \to \mathbb{Z}_p$ be a homomorphism from the additive group $\mathbb{F}_{p^\tau}$ to the additive group $\mathbb{Z}_p$, which is defined as follows

$$\phi(r) = r_0, \quad \forall r = \sum_{i=0}^{\tau-1} r_i X^i \in \mathbb{R}. \tag{39}$$

Note that there exists an irreducible polynomial $g(X) \in \mathbb{Z}_p[X]$ of degree τ such that $\mathbb{R} = \mathbb{F}_{p^\tau} = \mathbb{Z}_p[X]/\langle g(X) \rangle$ and any element $r \in \mathbb{F}_{p^\tau}$ can be written as $r = \sum_{i=0}^{\tau-1} r_i X^i \in \mathbb{Z}_p[X]$ for some $r_0, \ldots, r_{\tau-1}$. In particular, the representation of r into the sum is always unique.

Then it is easy to see that ϕ is a surjective homomorphism and thus satisfies the requirement of (b). For (c), we choose the $\mathbb{R}$-module $\mathbb{H} = \mathbb{R}^{h+1} (= \mathcal{S}_2 = \mathcal{C}_2)$ and $\psi : \mathcal{S}_1 \to \mathbb{H}$ such that

$$\psi(\alpha) = (1, \mathbf{w}) \tag{40}$$

Then it is easy to verify that the function $\Phi : \mathbb{H} \times \mathcal{C}_2 \to \mathbb{R}$ define by

$$\Phi(\mathbf{h}, \mathbf{c}) = \langle \mathbf{h}, \mathbf{c} \rangle, \quad \forall \mathbf{h} \in \mathbb{H} = \mathbb{R}^{h+1}, \mathbf{c} \in \mathcal{C}_2 = \mathbb{R}^{h+1}. \tag{41}$$

is bilinear. For any $\alpha \in \mathcal{S}_1$, any $(\mathbf{c}_0, \mathbf{c}_1) \leftarrow \mathsf{Share}_1(\alpha)$, any $x \in \mathcal{S}_1$, Eq. (37), (40) and (41) jointly imply that the $\rho(\alpha, x)$ in Eq. (9) is

$$\rho(\alpha, x) = \langle (1, \mathbf{w}), \mathbf{s}_2(\mathbf{w}, \alpha, x) \rangle. \tag{42}$$

For the above choices of $\mathbf{w}, \alpha$, and x, we set $\sigma = 1$, show that Eq. (10) is satisfied (see the proof for Theorem 7), and thus finally meet the requirement of (c). Applying our transformation from Section 3 with the related building blocks as above, we get the expected perfectly secure DPF.

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: Given $\alpha \in [N]$ and $\beta \in \mathbb{Z}_p$, generate $\mathbf{c}_\ell = E(\alpha) + \zeta_\ell \mathbf{w}$, $\forall \ell = 0, 1, \ldots, n-1$, choose $\mathbf{h}_0, \ldots, \mathbf{h}_t \leftarrow \mathbb{H} = \mathbb{R}^{h+1}$ uniformly subject to

$$\mathbf{h}_0 + \cdots + \mathbf{h}_t = \beta \diamond (1, \mathbf{w}),$$

  output $k_{nj+\ell} = (\mathbf{h}_j, \mathbf{c}_\ell)$, where $j \in \{0, \ldots, t\}, \ell \in \{0, 1, \ldots n-1\}$.
- $\mathsf{Eval}_i(k_i, x)$: For every $i \in \{0, 1, 2, 3\}$, $k_i = (\mathbf{h}_j, \mathbf{c}_\ell)$ and $x \in [N]$, output

$$\phi(\langle \mathbf{h}_j, (b_\ell F(\ell, x, \mathbf{c}_\ell), b'_\ell \nabla F(\ell, x, \mathbf{c}_\ell)) \,.$$

Fig. 5 A 4-server perfectly secure DPF with output group $\mathbb{G} = \mathbb{Z}_{p^\tau}$

Theorem 7. *The construction of* **Fig. 5** *is a perfectly secure $n(t+1)$-server DPF with output group $\mathbb{Z}_p$ (p is any prime, $n \in \mathbb{Z}^+$). for point function with with domain $[N]$, the key size of the DPF is $O(\log(p) \cdot N^{1/\lfloor \frac{2n-1}{t} \rfloor})$.*

Proof. First of all, we show the correctness and security of that construction. Since we have showed that the requirement (a) and (b) in Section 3 is realized, it suffices to show that

$$\phi(\rho(\alpha, x) \cdot \sigma) = \delta_{\alpha,x}.$$

In this construction we have

$$\begin{aligned}
\phi(\rho(\alpha, x) \cdot \sigma) &= \phi(\rho(\alpha, x) \\
&= \langle (1, \mathbf{w}), \mathbf{s}_2(\mathbf{w}, \alpha, x) \rangle \\
&= \langle (1, \mathbf{w}), \sum_{\ell=0}^{n-1} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell) \rangle \\
&= \sum_{\ell=0}^{n-1} (b_\ell F(\ell, x, \mathbf{c}_\ell) + \langle \mathbf{w}, b'_\ell \nabla F(\ell, x, \mathbf{c}_\ell) \rangle) \\
&= \sum_{\ell=0}^{n-1} (b_\ell f(\zeta_\ell) + b'_\ell f'(\zeta_\ell)) \\
&= a_0 = \delta_{\alpha,x}.
\end{aligned}$$

Then the requirement (c) satisfies, then the correctness and security of $(\mathsf{Gen}, \{\mathsf{Eval}_i\}_{0 \le i < n(t+1)})$ follows from Theorem 5.

Finally, we determine the key size in this construction. Each key k_i is in $\mathbb{H} \times \mathcal{C}_1 = \mathbb{F}_{p^\tau}^{2h+1}$, whose size is $O(\tau \cdot \log(p) \cdot h)$ with $\binom{h}{d} > n$ and $d = \lfloor \frac{2n-1}{t} \rfloor$. Thus the key size is $O(\log(p) \cdot N^{\lfloor \frac{2n-1}{t} \rfloor})$, which completes the proof. □

4.2.2 DPF with Smaller Key Sizes

In this section, we show that it is possible to have a perfectly secure DPF with output group $\mathbb{G} = \mathbb{Z}_p$ and much shorter keys. We construct an 8-server DPF whose key size is upper bounded by a function only linearly dependent of $\log p$, which reach the optimal

key size relative to p. To obtain the DPF, we firstly build a share conversion from Efremenko's PIR [12] and then apply the transformation from Section 3.

Share Conversion. Let $f_{\alpha,\beta}$ be a point function with domain $[N]$ and output group $\mathbb{G} = \mathbb{Z}_p$. Given the prime p, we choose an integer $m = p_1 p_2$ such that $\gcd(p, m) = 1$, where $p_1, p_2 \leq 5$ are distinct primes. Then there is a prime power $q = p^\tau$ such that $m|(q-1)$. We set

$$\mathbb{R} = \mathbb{F}_q, \tag{43}$$

the finite field of q elements. In our share conversion, the SSSs $(\mathcal{L}_1 = (\mathsf{Share}_1, \mathsf{Recov}_1), \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$ are chosen such that

$$\mathcal{S}_1 = [N]; \quad \mathcal{C}_1 = \mathbb{Z}_m^h; \quad \mathcal{S}_2 = \mathcal{C}_2 = \mathbb{R}, \tag{44}$$

where $h \in \mathbb{Z}^+$ is an integer such that there is an S_m-matching family $(U, V) \subseteq \mathbb{Z}_m^h$ of size N, and $\mathcal{C}_1, \mathcal{C}_2$ are the share spaces of the two SSSs. Let $\gamma \in \mathbb{F}_q^*$ have multiplicative order m. Let $P(X) = a_0 X^{b_0} + a_1 X^{b_1} + a_2 X^{b_2} + a_3 X^{b_3} \in \mathbb{F}_q[X]$ be the trivial S_m-decoding polynomial from Eq. (3). For $\alpha \in \mathcal{S}_1$, $\mathsf{Share}_1(\alpha)$ generates two shares $\mathbf{c}_0, \mathbf{c}_1 \in \mathcal{C}_1$ by mapping α to a vector $\mathbf{v}_\alpha \in V$, randomly choosing $\mathbf{w} \leftarrow \mathbb{Z}_m^h$, and finally setting

$$\mathbf{c}_\ell = \mathbf{w} + b_\ell \mathbf{v}_\alpha, \ \forall \ell \in \{0, 1, 2, 3\} \tag{45}$$

Given $\mathbf{c}_\ell$ and any $x \in \mathcal{S}_1$, the local conversion function $\mathsf{Conv}(\ell, x, \mathbf{c}_\ell)$ is defined by

$$\mathsf{Conv}(\ell, x, \mathbf{c}_\ell) = a_\ell \gamma^{\langle \mathbf{c}_\ell, \mathbf{u}_x \rangle_m} \tag{46}$$

where $\mathbf{u}_x \in U$ is the x-th element of U. Finally, the SSS $(\mathcal{L}_2, \mathcal{S}_2)$ is additive and may recover a value $s_2(\mathbf{w}, \alpha, x)$ via computing

$$s_2(\mathbf{w}, \alpha, x) = \sum_{\ell=0}^{3} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell). \tag{47}$$

Eq. (47) gives a binary relation $R \subseteq \mathcal{S}_1 \times \mathcal{S}_2$ that will be used in our transformation:

$$R = \{(\alpha, s_2(\mathbf{w}, \alpha, x)) : \alpha, x \in \mathcal{S}_1, \mathbf{w} \in \mathbb{Z}_m^h\}. \tag{48}$$

From Share Conversion to DPF. Besides the ring $\mathbb{R}$, the SSSs $(\mathcal{L}_1, \mathcal{S}_1)$ and $(\mathcal{L}_2, \mathcal{S}_2)$, the binary relation R, and the local share conversion function Conv that satisfies the requirement of (a) in Section 3, we still need to properly choose a module $\mathbb{H}$ and three functions ϕ, ψ, Φ that satisfy (b) and (c), in order to apply our transformation. Note that $\mathbb{G}$ is a subgroup of the additive group of $\mathbb{R}$ and contains the identity element of $\mathbb{R}$. As there is an irreducible polynomial $g(X) \in \mathbb{Z}_p[X]$ of degree τ such that

$$\mathbb{R} = \mathbb{F}_q = \mathbb{Z}_p[X]/(g(X)), \tag{49}$$

for any $r \in \mathbb{R}$, there exist τ elements $r_0, \ldots, r_{\tau-1} \in \mathbb{G}$ such that $r = \sum_{i=0}^{\tau-1} r_i X^i \in \mathbb{Z}_p[X]$. In particular, the representation of r into the sum is unique. We choose $\phi : \mathbb{R} \to \mathbb{G}$ such that

$$\phi(r) = r_0, \quad \forall r = r_0 + r_1 X + \cdots + r_{\tau-1} X^{\tau-1} \in \mathbb{R}. \tag{50}$$

Then it is easy to see that ϕ is a surjective homomorphism and thus satisfies the requirement of (b). For (c), we choose the $\mathbb{R}$-module $\mathbb{H} = \mathbb{R}$ $(= \mathcal{S}_2 = \mathcal{C}_2)$ and $\psi : \mathcal{S}_1 \to \mathbb{H}$ such that

$$\psi(\alpha) = 1, \quad \forall \alpha \in \mathcal{S}_1 \tag{51}$$

Then it is easy to verify that the function $\Phi : \mathbb{H} \times \mathcal{C}_2 \to \mathbb{R}$ define by

$$\Phi(h, c) = h \cdot c, \quad \forall h \in \mathbb{H} = \mathbb{R}, \quad c \in \mathcal{C}_2 = \mathbb{R}, \tag{52}$$

is bilinear. For any $\alpha \in \mathcal{S}_1$, any $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \leftarrow \mathsf{Share}_1(\alpha)$, any $x \in \mathcal{S}_1$, Eq. (47), (51), and (52) jointly imply that the $\rho(\alpha, x)$ in Eq. (9) is

$$\rho(\alpha, x) = s_2(\mathbf{w}, \alpha, x). \tag{53}$$

For the above choices of $\mathbf{w}, \alpha$, and x, we set

$$\sigma = \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}, \tag{54}$$

show that Eq. (10) is satisfied (see the proof for Theorem 8), and thus meet the requirement of (c). Applying our transformation from Section 3 with the related building blocks as above, we get the expected perfectly secure DPF (see **Fig. 6**).

- $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta})$: Given $\alpha \in [N]$ and $\beta \in \mathbb{Z}_p$, generate $\mathbf{c}_\ell \leftarrow \mathbf{w} + b_\ell \mathbf{v}_\alpha$ for all $\ell \in \{0, 1, 2, 3\}$, choose $h_0, h_1 \leftarrow \mathbb{H} = \mathbb{F}_q$ subject to

  $$h_0 + h_1 = \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m} \cdot \beta,$$

  output eight keys $\{k_i\}_{i=0}^7$, where $k_i = (h_j, \mathbf{c}_\ell)$ for $i = 4j + \ell$ $(\ell \in \{0, 1, 2, 3\}, j \in \{0, 1\})$.
- $\mathsf{Eval}_i(k_i, x)$: For every $i \in \{0, \ldots, 7\}$, $k_i = (h_j, \mathbf{c}_\ell)$ and $x \in [N]$, output

  $$\phi\big(h_j \cdot a_\ell \gamma^{\langle \mathbf{c}_\ell, \mathbf{u}_x \rangle_m}\big).$$

Fig. 6 An 8-server perfectly secure DPF with output group $\mathbb{G} = \mathbb{Z}_p$

Theorem 8. *The construction of* **Fig. 6** *gives a perfectly secure 8-server DPF with output group $\mathbb{Z}_p$ (p is any prime). For point functions with domain $[N]$, the key size of the DPF is $O(2^{10\sqrt{\log N \log\log N}} + \log p)$.*

Proof. First of all, we show the correctness and security of that construction. Since we have showed that the requirement (a) and (b) in Section 3 is realized, it suffices to show that

$$\phi(\rho(\alpha, x) \cdot \sigma) = \delta_{\alpha,x}.$$

In this construction we have

$$\begin{aligned}
\phi(\rho(\alpha, x) \cdot \sigma) &= \phi\left(\sum_{\ell=0}^{3} \mathsf{Conv}(\ell, x, \mathbf{c}_\ell) \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}\right) \\
&= \phi\left(\sum_{\ell=0}^{3} a_\ell \gamma^{\langle \mathbf{c}_\ell, \mathbf{u}_\alpha \rangle_m} \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}\right) \\
&= \phi\left(\sum_{\ell=0}^{3} a_\ell \gamma^{\langle \mathbf{w}, \mathbf{u}_x \rangle_m} (\gamma^{\langle \mathbf{v}_\alpha, \mathbf{u}_x \rangle_m})^{b_\ell} \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}\right) \\
&= \phi\left(P(\gamma^{\langle \mathbf{v}_\alpha, \mathbf{u}_x \rangle_m}) \cdot \gamma^{\langle \mathbf{w}, \mathbf{u}_x \rangle_m} \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m}\right).
\end{aligned}$$

- For $x = \alpha$, $\gamma^{\langle \mathbf{w}, \mathbf{u}_x \rangle_m} \cdot \gamma^{-\langle \mathbf{w}, \mathbf{u}_\alpha \rangle_m} = 1$ and $P(\gamma^{\langle \mathbf{v}_\alpha, \mathbf{u}_x \rangle_m}) = 1$. Thus $\phi(\rho(\alpha, x) \cdot \sigma) = 1$.
- For $x \neq \alpha$, $P(\gamma^{\langle \mathbf{v}_\alpha, \mathbf{u}_x \rangle_m}) = 0$. Thus $\phi(\rho(\alpha, x) \cdot \sigma) = 0$.

Then the requirement (c) satisfies, then the correctness and security of $(\mathsf{Gen}, \{\mathsf{Eval}_i\}_{0 \le i < 8})$ follows from Theorem 5.

Finally, we determine the key size in this construction. Each key k_i is in $\mathbb{H} \times \mathcal{C}_1 = \mathbb{Z}_m^h \times \mathbb{F}_q$, whose size is $O(\log(m) \cdot h + \log p)$. Note that $m = p_1 p_2$ may be the product of any two distinct primes such that $\gcd(m, p) = 1$. For $p = 2, p = 3$, and $p \ge 5$, we may choose $m = 15, m = 10$ and $m = 6$ respectively such that $\max\{p_1, p_2\}$ is minimized. In particular, we always have that $2 \max\{p_1, p_2\} \le 10$ and thus from Theorem 3 we have that $h = O(2^{10\sqrt{\log N \log\log N}} + \log p)$. Then we know the key size is upper bounded by $|k_i| = O(2^{10\sqrt{\log N \log\log N}} + \log p)$, which completes the proof. □

Generalization to More Servers. Note that the communication complexity of Efremenko's PIR can be reduced if the modulus m has more prime powers as factors (see Theorem 3). Next, we show that DPFs that use more servers but have smaller key sizes can be obtained in a way similar to that of **Fig. 6**.

Theorem 9. *For any integer $r \ge 2$, there exists a perfectly secure 2^{r+1}-server DPF with output group $\mathbb{Z}_p$ (p is any prime). For point functions with domain $[N]$, the key size of the DPF is $O(2^{c(r)} \sqrt[r]{\log N (\log\log N)^{r-1}} + \log p)$, where $c(r)$ is roughly the $(r+1)^{th}$ smallest prime and independent of p.*

Proof. Let $m = p_1 p_2 \dots p_r$, where $p_1 \le \dots \le p_r$ are distinct primes and $p_i \neq p$ for all $i \in [r]$. Let t be the multiplicative order of p modulo m. Then we have that $m | (p^\tau - 1)$. There exist an S_m-decoding polynomial with 2^r monomials. We can similarly construct a perfectly secure 2^{r+1}-DPF. The key size of the DPF is upper bounded by $O(2^{c(r)} \sqrt[r]{\log N (\log\log N)^{r-1}} + \log p)$ by Theorem 3, where $c(r)$ is independent of p. Since p_r can be taken no more than the $(r+1)^{th}$ smallest prime, from Grolmusz [22] (see Section 2.5) $c(r)$ is roughly the $(r+1)^{th}$ smallest prime. □

Reducing the Number of Servers. The DPF in Theorem 8 doubles the number of servers required by the underlying PIR, which is exactly equal to the number of monomials in the S_m-decoding polynomial. Therefore, an S_m-decoding polynomial with fewer monomials will give DPFs that use fewer servers.

Theorem 10. *Let p be a prime. Let m be a product of r distinct primes such that $\gcd(m, p) = 1$. Let t be the multiplicative order of p modulo m. If there is an S_m-decoding polynomial in $\mathbb{F}_{p^\tau}[X]$ that has n monomials, then there is a perfectly secure $2n$-server DPF with output group $\mathbb{Z}_p$. For point functions domain $[N]$, the key size of the DPF is $O(2^{c(r)}\sqrt[r]{\log N(\log\log N)^{r-1}} + \log p)$, where $c(r)$ is roughly equal to the $(r+1)^{th}$ smallest prime, and is independent of p.*

Theorem 10 have many consequences. For example, for $p = 2$, if we choose the S_{511}-decoding polynomial from [12], which has only 3 monomials, then we can obtain a 6-DPF with output group $\mathbb{Z}_2$. In general, we can reduce the number of servers with the composition theorem (Theorem 4). The nice integers [14] allow us to further reduce the number of required servers.

Statistically Secure DPF. Since our construction satisfies Eq. (14), our construction could lead to statistically secure DPFs (see **Fig. 3**).

Theorem 11. *For any integer $r \geq 2$, there exists a $2^{-\Omega(\lambda)}$-statistically secure 2^r-server DPF with output group $\mathbb{Z}_p$ (p is any prime). For point functions with domain $[N]$, the key size of the DPF is $O(\lambda \cdot 2^{c(r)}\sqrt[r]{\log N(\log\log N)^{r-1}} + \lambda \log p)$, where $c(r)$ is roughly the $(r+1)^{th}$ smallest prime and independent of p.*

Theorem 11 gives a statistically secure 4-server DPF with key size $O(\lambda \cdot 2^{10\sqrt{\log N(\log\log N)}} + \lambda \log p)$ only λ times the key size of the 8-server DPF from Theorem 8 but uses fewer servers.

5 Application to PIR with Result Verification

5.1 PIR with Result Verification

Early PIR protocols always assume *honest-but-curious* servers that strictly follow the protocol's specifications. Recently, a lot of efforts [8, 23–28] have been made to deal with *malicious* servers that may collude and provide wrong answers to the client, in order to deceive the client into reconstructing an incorrect value. The protocols that can tolerate malicious servers have particular interest in the modern age of cloud computing because it allows the servers to be implemented by the untrusted cloud services, i.e., outsourcing the servers' computations to the cloud.

Ke and Zhang [8] proposed PIR with result verification (PIR-RV) that can deal with the condition that half or even more servers are malicious without a trusted third party. Like PIR, an n-server PIR-RV protocol involves two kinds of participants: a *client* and n *servers*, where each server has a database $\mathrm{DB} \in \{0,1\}^N$ and the client has an index $\alpha \in [N]$. Compared with PIR, PIR-RV allows the client to verify whether the value of DB_α is correctly reconstructed, when some of the servers may collude and provide wrong answers. The syntax of an n-server PIR-RV $\Gamma = (\mathsf{Que}, \mathsf{Ans}, \mathsf{Rec})$ is identical to that of PIR (see Definition 4), except that Rec is replaced with the following:

- $\{\mathrm{DB}_\alpha, \perp\} \leftarrow \mathsf{Rec}(\alpha, \{\mathsf{ans}_j\}_{j=0}^{n-1}, \mathsf{aux})$*: This is a deterministic* reconstructing algorithm *for the client. Given the the retrieval index* α*, the answers* $\{\mathsf{ans}_j\}_{j=0}^{n-1}$ *and the auxiliary information* aux*, it either outputs* DB_α *or a special symbol* $\perp$ *to indicate that at least one of the answers is incorrect.*

The requirements of correctness and privacy are identical to those in Definition 4. Besides, PIR-RV should satisfy the additional requirement of security. Intuitively, an n-server PIR-RV protocol is (t, ϵ)-*secure* if no collusion of up to t servers can cause the client with input α to output a value $\notin \{\mathrm{DB}_\alpha, \perp\}$ with probability $> \epsilon$, by providing wrong answers.

Definition 10 (Security). *Consider the security experiment in* **Fig. 7**. *An* n*-server PIR-RV protocol* Γ *is* (t, ϵ)-secure *if for any set* $T \subseteq \{0, \ldots, n-1\}$ *with* $|T| \leq t$*, any adversary* $\mathcal{A}$ *that controls the* j-th *servers for all* $j \in T$*, any* N*, any* $\mathrm{DB} \in \{0,1\}^N$ *and any* $\alpha \in [N]$, $\Pr[\mathsf{EXP}^{\mathrm{Ver}}_{\mathcal{A},\Gamma}(N, \mathrm{DB}, \alpha, T) = 1] \leq \epsilon$.

- The challenger generates $(\{\mathsf{que}_j\}_{j=0}^{n-1}, \mathsf{aux}) \leftarrow \mathsf{Que}(N, \alpha)$ and sends $\{\mathsf{que}_j\}_{j\in T}$ to $\mathcal{A}$.
- The adversary $\mathcal{A}$ chooses the answers $\{\mathsf{ans}'_j\}_{j\in T}$ to the challenger.
- The challenger computes $\mathsf{ans}'_j \leftarrow \mathsf{Ans}(\mathrm{DB}, \mathsf{que}_j)$ for all $j \in \{0, \ldots, n-1\} \setminus T$.
- If $\mathsf{Rec}(\alpha, \{\mathsf{ans}_j\}_{j=0}^{n-1}, \mathsf{aux}) \notin \{\mathrm{DB}_\alpha, \perp\}$, outputs 1; otherwise outputs 0.

Fig. 7 The security experiment $\mathsf{EXP}^{\mathrm{Ver}}_{\mathcal{A},\Gamma}(N, \mathrm{DB}, \alpha, T)$.

5.2 Our Construction

In this section, we generalize the DPF in **Fig. 2** to a t-private $n(\zeta+1)$-server DPF Π_ζ for any $\zeta \geq t$ and present our t-private $n(\zeta+1)$-server PIR-RV protocol Γ (see **Fig. 8**).

Since PIR allows the client to privately retrieve DB_α from DB of size N, DPF indicates PIR with the same number of servers by sending every key k_i for $f_{\alpha,1}$ to the server $\mathcal{S}_i$, getting $\sum_{\ell=1}^{N} \mathsf{Eval}_i(k_i, \ell) \cdot \mathrm{DB}_\ell$ in return and reconstructing $\sum_{\ell=1}^{N} f_{\alpha,1}(\ell) \cdot \mathrm{DB}_\ell (= \mathrm{DB}_\alpha)$ by adding up all the answers. A simple idea to construct PIR-RV is to let the client randomly choose $\beta \in \mathbb{G}$ and use $f_{\alpha,\beta}$ instead of $f_{\alpha,1}$. The client will learn $\mathrm{DB}_\alpha = 0$ or 1 if the output is 0 or β, respectively. If the DPF output is not in $\{0, \beta\}$, then some answers must be incorrect. Due to the privacy of β, we can get an $n(t+1)$-server PIR-RV protocol which can tolerate at most t malicious servers, if the DPF Π in **Fig. 2** is used.

To construct PIR-RV that tolerates more malicious servers, we apply the above protocol multiple times with different β, in order to make sure the colluding servers cannot break the security for each time. Suppose there are $\geq n$ honest servers. The protocol will be executed multiple times. Each time it sends the keys about h_0 to a different set of n servers. Then the $n(t+1)$-server PIR-RV can tolerate nt malicious servers.

We can replace the t-private $n(t+1)$-server DPF Π in above PIR-RV with a t-private $n(\zeta+1)$-server DPF Π_ζ for any $\zeta \geq t$. The protocol Π_ζ is identical to Π, except that its key generation Gen divides $(\sigma \cdot \beta) \diamond \psi(\alpha)$ into sum of $(\zeta+1)$ shares instead of

$(t+1)$ shares in Π: $h_0 + \cdots + h_\zeta = (\sigma \cdot \beta) \diamond \psi(\alpha)$. The client can generate $n(\zeta+1)$ keys $\{k_i\}_{i=0}^{n(\zeta+1)-1}$, where $k_i = (h_j, \mathbf{c}_\ell)$ for $i = nj+\ell$ $(0 \le j \le \zeta, 0 \le \ell < n)$. With correctness and privacy similar to that of Π, the protocol Π_ζ also satisfies that no collusion of up to ζ servers can learn any information about β since any ζ servers cannot get all of $\{h_j\}_{j=0}^{\zeta}$, the $n(\zeta+1)$ PIR-RV protocol Γ (**Fig. 8**) constructed with Π_ζ can tolerate $n\zeta$ malicious servers. Let $\mathcal{N} = \{0, 1, \ldots, n(\zeta+1)-1\}$, $\mathcal{B} = \{b | b \subseteq \mathcal{N}, |b| = n\}$ and let $\Pi_\zeta = (\mathsf{Gen}^\zeta, \mathsf{Eval}_0^\zeta, \ldots, \mathsf{Eval}_{n-1}^\zeta)$ be the DPF with output group $\mathbb{G}$. Our PIR-RV protocol Γ is shown in **Fig. 8**.

- $\mathsf{Que}(N, \alpha)$: For each $b \in \mathcal{B}$, randomly choose $\beta_b \in \mathbb{G}$, compute $\left(k_0^b, k_1^b, \ldots, k_{n(\zeta+1)-1}^b\right) \leftarrow \mathsf{Gen}^\zeta(1^\lambda, f_{\alpha,\beta_b})$; denote P_b, Q_b are bijective functions where $P_b : b \to \{0, \ldots, n-1\}$ and $Q_b : (\mathcal{N} \setminus b) \to \{n, n+1, \ldots, n(\zeta+1)-1\}$, for each $j \in \mathcal{N}$, compute
$$\mathsf{que}_{j,b} = \begin{cases} k^b_{P_b(j)}, & j \in b \\ k^b_{Q_b(j)}, & j \notin b \end{cases},$$
output $n(\zeta+1)$ queries $\{\mathsf{que}_j\}_{j=0}^{n(\zeta+1)-1}$ where $\mathsf{que}_j = \{\mathsf{que}_{j,b}\}_{b\in\mathcal{B}}$, together with the auxiliary information $\mathsf{aux} = \{\beta_b\}_{b\in\mathcal{B}}$.
- $\mathsf{Ans}(\mathrm{DB}, \mathsf{que}_j, j)$: Parse $\mathsf{que}_j = \{\mathsf{que}_{j,b}\}_{b\in\mathcal{B}}$. For each $b \in \mathcal{B}$, parse $\mathsf{que}_{j,b} = k^b$, compute
$$\mathsf{ans}_{j,b} = \sum_{\ell=1}^{N} \mathrm{DB}_\ell \cdot \mathsf{Eval}_j^\zeta(k^b, \ell),$$
output $\mathsf{ans}_j = \{\mathsf{ans}_{j,b}\}_{b\in\mathcal{B}}$.
- $\mathsf{Rec}(\alpha, \{\mathsf{ans}_j\}_{j=0}^{n(\zeta+1)-1}, \mathsf{aux})$: Parse $\mathsf{aux} = \{\beta_b\}_{b\in\mathcal{B}}$ and $\mathsf{ans}_j = \{\mathsf{ans}_{j,b}\}_{b\in\mathcal{B}}$ for all $j \in \mathcal{N}$. For each $b \in \mathcal{B}$, compute $R_b = \sum_{j=0}^{n(\zeta+1)-1} \mathsf{ans}_{j,b}$. If there exists $b \in \mathcal{B}$ such that $R_b \notin \{0, \beta_b\}$, output $\perp$; otherwise, for each $b \in \mathcal{B}$, set $\mathsf{res}_b = \delta_{R_b,\beta_b}$. If there exist distinct $b_1, b_2 \in \mathcal{B}$ such that $\mathsf{res}_{b_1} \neq \mathsf{res}_{b_2}$, output $\perp$; otherwise, output $\{\mathsf{res}_b\}_{b\in\mathcal{B}}$.

Fig. 8 $n(\zeta+1)$-Server PIR-RV protocol Γ.

Theorem 12. *The $n(\zeta+1)$-server PIR-RV protocol Γ is t-private and $(n\zeta, \frac{1}{|\mathbb{G}|})$-secure. If the key size of the DPF used in the PIR-RV is K, then the total communication complexity of the PIR-RV is $\mathsf{CC}_\Gamma(N) = O(\zeta^{n+1}K)$*

Proof. The privacy and correctness of the PIR-RV protocol follows directly from the perfect security and correctness of the DPF Π_ζ. It suffices to prove the security and the communication complexity.

If there exists an adversary Adv that can wins the security experiment in **Fig. 7**, for every $b \in \mathcal{B}$, the sum of all outputs of servers should be changed from 0 to β_b or from β_b to 0 for every $b \in \mathcal{B}$. This shows that Adv can only break the PIR-RV by guessing every β_b correctly. We only need to show that there exist a $b \in \mathcal{B}$ such that

Adv knows no information about β_b. The privacy of β_b depends on the privacy of β of Π_ζ.

Each server can only get one share of $(\sigma \cdot \beta_b) \diamond \psi(\alpha)$. Since $(\sigma \cdot \beta_b) \diamond \psi(\alpha)$ is shared additively to $h_{0,b} + \cdots + h_{\zeta,b}$, and the n $h_{0,b}$s are contained in the first n keys. Thus the $h_{0,b}$s generated from function $\mathsf{Gen}(1^\lambda, f_{\alpha,\beta_b})$ are contained in $\{k_i^b\}_{i=0}^{n-1}$. Since the image of $P_b(j)$ for $j \in b$ is in $\{0, \ldots, n-1\}$, these keys containing $h_{0,b}$ are given to the servers with index in b. As there exist a $b \in \mathcal{B}$ that the n servers with index in b are all honest. For this case, Adv know nothing about $h_{0,b}$. Since $(\sigma \cdot \beta_b) \diamond \psi(\alpha)$ is additive shared to $h_{0,b}, \ldots, h_{\zeta,b}$, for every b, the distribution of $(h_{1,b}, \ldots, h_{\zeta,b})$ is uniform in $\mathbb{H}^{n-1}$. This means that any $\beta_0, \beta_1 \in \mathbb{G}$, the distribution of $(h_{1,b}, \ldots, h_{\zeta,b})$ under the case $\beta_b = \beta_0$ and $\beta_b = \beta_1$ are the same. Furthermore, the change of β_b have no effect on $\mathbf{c}_0, \ldots, \mathbf{c}_{n-1}$, which means the distribution of the keys that Adv gets can reveal nothing about β_b. Thus the malicious servers have no information of β_b. For the server, the distribution of β_b is the uniform distribution over $\mathbb{Z}_{p^\tau}$, so the protocol is $(n\zeta, \frac{1}{p^\tau})$-secure.

For each $b \in \mathcal{B}$, the client only send the key to each server and the size of answer is independent of N, so the communication complexity is $O(K)$. Since $|\mathcal{B}| = O(\zeta^n)$, the communication complexity of the PIR-RV protocol Γ is $O(\zeta^n \cdot K)$ to each server, hence the total communication complexity is $\mathsf{CC}_\Gamma(N) = O(\zeta^{n+1} \cdot K)$. □

In particular, if we use the DPF in Section 4.1, and share $(\sigma \cdot \beta) \diamond \psi(\alpha)$ additively to more shares, then we can get a PIR-RV that only need 2 honest servers with subpolynomial communication complexity. We state this as follow.

Theorem 13. *For any $\zeta \in \mathbb{Z}^+$, there exist a 1-private $2(\zeta+1)$-server $(2\zeta, \frac{1}{2^\tau})$-secure PIR-RV protocol with database size N and total communication complexity $\mathsf{CC}_\Gamma(N) = O(\zeta^3\tau \cdot 2^{6\sqrt{\log N \log\log N}})$.*

If we use the DPF in Section 4.2, similarly we can get a PIR-RV that needs 4 honest servers which is quite more efficient. When we take a very large $\mathbb{Z}_p$ as the output group, our PIR-RV protocol could be very close to perfect security with an extremely slow growth in communication complexity.

Theorem 14. *For any $\zeta \in \mathbb{Z}^+$, there exists a 1-private $4(\zeta+1)$-server $(4\zeta, \frac{1}{p})$-secure PIR-RV protocol with database size N and total communication complexity $\mathsf{CC}_\Gamma(N) = O(\zeta^5 \cdot 2^{6\sqrt{\log N \log\log N}} + \zeta^5 \log p)$.*

Ke and Zhang [8] proposed a 2-server $(1, \frac{3}{p-2})$-secure PIR-RV protocol with communication complexity $O(\log p \cdot \sqrt{N})$. Compared with [8], our protocol provides subpolynomial communication complexity and higher malicious server tolerance with at least 4 servers.

6 Conclusions

In this paper, we provide a transformation from share conversion to information-theoretic DPFs. With this transformation, we give a perfectly secure 4-DPF for any output group and a 8-DPF with smaller key size for output group $\mathbb{Z}_p$. We also construct new efficient PIR-RV protocols with the new DPFs. Our DPFs with subpolynomial key size are all t-private for $t = 1$. The question is open for $t > 1$.

References

[1] Gilboa, N., Ishai, Y.: Distributed point functions and their applications. In: Nguyen, P.Q., Oswald, E. (eds.) Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8441, pp. 640–658 (2014). https://doi.org/10.1007/978-3-642-55220-5_35 . https://doi.org/10.1007/978-3-642-55220-5_35

[2] Boyle, E., Gilboa, N., Ishai, Y., Kolobov, V.I.: Information-theoretic distributed point functions. In: Dachman-Soled, D. (ed.) 3rd Conference on Information-Theoretic Cryptography, ITC 2022, July 5-7, 2022, Cambridge, MA, USA. LIPIcs, vol. 230, pp. 1–14 (2022). https://doi.org/10.4230/LIPIcs.ITC.2022.17 . https://doi.org/10.4230/LIPIcs.ITC.2022.17

[3] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995, pp. 41–50 (1995). https://doi.org/10.1109/SFCS.1995.492461 . https://doi.org/10.1109/SFCS.1995.492461

[4] Beimel, A., Ishai, Y., Kushilevitz, E., Orlov, I.: Share conversion and private information retrieval. In: Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012, pp. 258–268 (2012). https://doi.org/10.1109/CCC.2012.23 . https://doi.org/10.1109/CCC.2012.23

[5] Paskin-Cherniavsky, A., Nissenbaum, O.: New bounds and a generalization for share conversion for 3-server PIR. Entropy **24**(4), 497 (2022) https://doi.org/10.3390/e24040497

[6] Paskin-Cherniavsky, A., Schmerler, L.: On share conversions for private information retrieval. Entropy **21**(9), 826 (2019) https://doi.org/10.3390/e21090826

[7] Dvir, Z., Gopi, S.: 2-server PIR with sub-polynomial communication. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015, pp. 577–584 (2015). https://doi.org/10.1145/2746539.2746546 . https://doi.org/10.1145/2746539.2746546

[8] Ke, P., Zhang, L.F.: Two-server private information retrieval with result verification. In: IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022, pp. 408–413 (2022). https://doi.org/10.1109/ISIT50566.2022.9834706 . https://doi.org/10.1109/ISIT50566.2022.9834706

[9] Boneh, D., Boyle, E., Corrigan-Gibbs, H., Gilboa, N., Ishai, Y.: Lightweight techniques for private heavy hitters. In: 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, pp. 762–776

(2021). https://doi.org/10.1109/SP40001.2021.00048 . https://doi.org/10.1109/SP40001.2021.00048

[10] Boyle, E., Gilboa, N., Ishai, Y.: Function secret sharing: Improvements and extensions. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, pp. 1292–1303 (2016). https://doi.org/10.1145/2976749.2978429 . https://doi.org/10.1145/2976749.2978429

[11] Woodruff, D.P., Yekhanin, S.: A geometric approach to information-theoretic private information retrieval. In: 20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA, pp. 275–284 (2005). https://doi.org/10.1109/CCC.2005.2 . https://doi.org/10.1109/CCC.2005.2

[12] Efremenko, K.: 3-query locally decodable codes of subexponential length. In: Mitzenmacher, M. (ed.) Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009, pp. 39–44 (2009). https://doi.org/10.1145/1536414.1536422 . https://doi.org/10.1145/1536414.1536422

[13] Itoh, T., Suzuki, Y.: Improved constructions for query-efficient locally decodable codes of subexponential length. IEICE Transactions on Information and Systems **93-D**(2), 263–270 (2010) https://doi.org/10.1587/transinf.E93.D.263

[14] Chee, Y.M., Feng, T., Ling, S., Wang, H., Zhang, L.F.: Query-efficient locally decodable codes of subexponential length. Computational Complexity **22**(1), 159–189 (2013) https://doi.org/10.1007/s00037-011-0017-1

[15] Corrigan-Gibbs, H., Boneh, D., Mazières, D.: Riposte: An anonymous messaging system handling millions of users. In: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015, pp. 321–338 (2015). https://doi.org/10.1109/SP.2015.27 . https://doi.org/10.1109/SP.2015.27

[16] Newman, Z., Servan-Schreiber, S., Devadas, S.: Spectrum: High-bandwidth anonymous broadcast. In: Phanishayee, A., Sekar, V. (eds.) 19th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2022, Renton, WA, USA, April 4-6, 2022, pp. 229–248 (2022). https://www.usenix.org/conference/nsdi22/presentation/newman

[17] Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R., Elsabagh, M., Kiourtis, N., Schulte, B., Stavrou, A.: Function secret sharing for PSI-CA: with applications to private contact tracing. CoRR **abs/2012.13053** (2020) 2012.13053

[18] Boyle, E., Gilboa, N., Ishai, Y.: Function secret sharing. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual

International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9057, pp. 337–367 (2015). https://doi.org/10.1007/978-3-662-46803-6_12 . https://doi.org/10.1007/978-3-662-46803-6_12

[19] Dummit, D.S., Foote, R.M.: Abstract Algebra vol. 3, (2004)

[20] Beimel, A.: Secret-sharing schemes: A survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6639, pp. 11–46 (2011). https://doi.org/10.1007/978-3-642-20901-7_2 . https://doi.org/10.1007/978-3-642-20901-7_2

[21] Yekhanin, S.: Towards 3-query locally decodable codes of subexponential length. In: Johnson, D.S., Feige, U. (eds.) Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007, pp. 266–274 (2007). https://doi.org/10.1145/1250790.1250830 . https://doi.org/10.1145/1250790.1250830

[22] Grolmusz, V.: Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. Combinatorica **20**(1), 71–86 (2000) https://doi.org/10.1007/s004930070032

[23] Beimel, A., Stahl, Y.: Robust information-theoretic private information retrieval. In: Cimato, S., Galdi, C., Persiano, G. (eds.) Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers. Lecture Notes in Computer Science, vol. 2576, pp. 326–341 (2002). https://doi.org/10.1007/3-540-36413-7_24 . https://doi.org/10.1007/3-540-36413-7_24

[24] Goldberg, I.: Improving the robustness of private information retrieval. In: 2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA, pp. 131–148 (2007). https://doi.org/10.1109/SP.2007.23 . https://doi.org/10.1109/SP.2007.23

[25] Devet, C., Goldberg, I., Heninger, N.: Optimally robust private information retrieval. In: Kohno, T. (ed.) Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, August 8-10, 2012, pp. 269–283 (2012). https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/devet

[26] Zhang, L.F., Safavi-Naini, R.: Verifiable multi-server private information retrieval. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) Applied Cryptography and Network Security - 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8479, pp. 62–79 (2014). https://doi.org/10.1007/978-3-319-07536-5_5 . https://doi.org/10.1007/978-3-319-07536-5_5

[27] Kurosawa, K.: How to correct errors in multi-server PIR. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11922, pp. 564–574 (2019). https://doi.org/10.1007/978-3-030-34621-8_20 . https://doi.org/10.1007/978-3-030-34621-8_20

[28] Zhang, L.F., Wang, H., Wang, L.: Byzantine-robust private information retrieval with low communication and efficient decoding. In: Suga, Y., Sakurai, K., Ding, X., Sako, K. (eds.) ASIA CCS '22: ACM Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May 2022 - 3 June 2022, pp. 1079–1085 (2022). https://doi.org/10.1145/3488932.3497773 . https://doi.org/10.1145/3488932.3497773