

Multi-Party Distributed Point Functions, Revisited

Elaine Shi¹ Tianyao Gu¹ Xuanye Zheng¹
Yue Yang¹ Yiping Liu¹ Yucheng Fu^{2*}

¹Carnegie Mellon University, Pittsburgh, PA, USA

elaineshi@cmu.edu, {tianyaog,xuanyez,yueyang}@andrew.cmu.edu, he.ren.chn@gmail.com

²University of Virginia, Charlottesville, VA, USA

zdp8uu@virginia.edu

Abstract

In this paper, we revisit the design of multi-party distributed point functions (DPFs) and make several new contributions that advance the state of the art. We begin by revisiting security amplification, a fundamental tool underlying many DPF constructions. In particular, the recent landmark work of Goel, Wang, and Wang (CRYPTO'25) critically relies on security amplification and, for a general polynomial number of parties, gives the only known construction based on one-way functions (OWFs) that achieves sublinear dependence on the input domain size. Unfortunately, due to a known gap in the proof of the security amplification theorem of Boyle et al. (CRYPTO'22), we currently still lack a fully established security amplification theorem for DPFs.

We fill this gap by providing a new proof of security amplification for DPFs with tight parameters. Equipped with this security amplification theorem as a key technical tool, we develop several new techniques that asymptotically improve the communication cost of multi-party DPFs in both the honest-majority and corrupt-majority settings. Our main results are summarized below, where N denotes the input domain size, m denotes the number of parties, and t denotes the corruption threshold:

- In the all-but-one-corrupt setting, we describe a new scheme based on OWFs with $\tilde{O}_\lambda \left(N^{\frac{1}{2}+\epsilon} \cdot \sqrt{m} \right)$ share size where $\epsilon > 0$ is an arbitrarily small constant. In comparison, the best previously known OWF-based construction due to Goel et al. incurs $\tilde{O}_\lambda \left(N^{\frac{1}{2}+\epsilon} \cdot m^3 \right)$ share size.
- In the honest-majority setting, assuming $m > (1 + \epsilon)Dt$ for some integer $D \geq 2$ and arbitrarily small constant $\epsilon > 0$, we construct a new OWF-based scheme with share size $\tilde{O}_\lambda(N^{\frac{1+\epsilon}{2D}})$, as well as an information-theoretically secure scheme with share size $\tilde{O}(N^{1/D})$. Both constructions achieve an *exponential* factor improvement in their dependence on m and t compared to the state-of-the-art schemes of Bunn, Kushilevitz, and Ostrovsky.

*The authors are listed in random order.

Contents

1	Introduction	1
1.1	Our Results and Contributions	2
1.2	Additional Related Work	3
2	Preliminaries	4
2.1	Notation and Terminology	4
2.2	Definitions: Multi-Party Distributed Point Function	5
3	Security Amplification of Distributed Point Functions	6
3.1	Strong Approximate Computational Indistinguishability	6
3.2	The Decoy Lemma: Upgrade from Regular to Strong $(1 - \delta)$ -Indistinguishability	7
3.3	Security Amplification for DPFs	8
4	Improved DPF: All-But-One Corruption Setting	8
4.1	Background: DPFs from Random Bipartite Graphs	9
4.2	Improved Random Bipartite Graph Design	10
4.3	Improved Transformation from Random Bipartite Graphs to DPFs	12
5	Improved DPF: Honest-Majority Setting with Computational Security	12
A	Additional Details for Strong Approximate Indistinguishability and DPF Security Amplification	17
A.1	Hardcore Lemma for Computational Indistinguishability	17
A.2	Why We Still Lack a DPF Amplification Theorem	19
A.3	Properties of Strong $(1 - \delta)$ -Computational Indistinguishability	21
A.4	Proof of the Decoy Lemma	23
A.5	From Regular to Strong $(1 - \delta)$ -Security for DPFs	26
A.6	Security Amplification for Strongly $(1 - \delta)$ -Secure DPFs	28
A.7	Security Amplification for Regular $(1 - \delta)$ -Secure DPFs	32
B	Deferred Details: Our All-But-One-Corrupt DPF	32
B.1	Proofs for Our New Random Bipartite Graphs	32
B.2	Improved Transformation from Bipartite Graphs to DPF	34
C	Deferred Details: Honest-Majority DPF with Computational Security	36
C.1	Compiler from All-But-One Corruption to Honest Majority	36
C.1.1	Background: Construction by Bunn et al.	37
C.1.2	Proof of Lemma C.2	38
C.2	Committee Selection	39
C.3	Putting Everything Together	40
D	Improved DPF: Honest-Majority Setting with Perfect Security	40
E	Strong Computational Indistinguishability for the Special Bipartite Graph Design of Goel <i>et al.</i>	43

1 Introduction

A point function $f^{\alpha,\beta}(\cdot)$ is a function that evaluates to some possibly non-zero value β at the point α , and 0 everywhere else. A *distributed point function* (DPF) [GI14] is a secret sharing of a point function $f^{\alpha,\beta}(\cdot)$ among m parties, where each party indexed $i \in [m]$ receives a share denoted $f_i^{\alpha,\beta}(\cdot)$. Further, the following properties are desired:

- *additive reconstruction*: Given any input x , it must be that $\sum_{i \in [m]} f_i^{\alpha,\beta}(x) = f^{\alpha,\beta}(x)$; and
- *t -privacy*: No adversary controlling $t < m$ parties should learn anything about α or β from the combination of the corrupt parties' shares.

As a special case of functional secret sharing [BGI15], distributed point functions (DPFs) have attracted the attention of the cryptography community. They are known to imply interesting applications such as Private Information Retrieval (PIR) [GI14, CGBM15, BGI15, BGI16], private set intersection [TSS⁺20, DIL⁺20, GRS22, GRS23, GGM24], pseudorandom correlation generators [BCGI18, BCG⁺19, BCG⁺20], distributed Oblivious RAM [DS17, BKKO20], privacy-preserving machine learning [RTPB20, YJG⁺24, JGB⁺24], and others [BS24, BGI19, BCG⁺21, BDSS25].

While earlier literature on DPF primarily focused on the two-party setting [GI14, CGBM15, BGI15, BGI16], several recent works [BKO22, GWW25, KPR25] have made notable progress on generalizing DPFs to multiple parties, assuming either all-but-one corruption or honest-majority. The current state of our knowledge is summarized in Table 1 and explained below. Throughout the paper, we let N denote the input domain size, m denote the total number of parties, and t denote the number of corrupted parties.

- *All-but-one corruption*. In the all-but-one corruption setting, the best known results [CGBM15, GWW25] achieve share size with a $\sqrt{N}$ dependence on the domain size N — excluding constructions based on untested assumptions [KPR25]. Specifically, Corrigan-Gibbs et al. [CGBM15] showed how to get share size $O_\lambda(\sqrt{N})$ assuming Decisional Diffie-Hellman (DDH)¹. Goel et al. [GWW25] achieved almost the same $\sqrt{N}$ dependence assuming only one-way functions (OWFs), but at the price of an additional polynomial dependence on the number of parties m — more precisely, their construction requires a per-party share size of $\tilde{O}_\lambda(N^{\frac{1}{2}+\epsilon} \cdot m^3)$ where ϵ denotes arbitrarily small positive constant.
- *Honest majority*. In the honest-majority setting, Bunn et al. [BKO22] showed how to achieve $N^{\frac{1}{2D}}$ dependence on N assuming that $m > Dt$ for some integer $D \geq 2$, but at the price of (super-)exponential dependence on either m or t . In other words, their scheme works only for a constant number of parties (or slightly greater than constant).

In this paper, we revisit multi-party DPFs. We observe that the status quo raises the following challenges.

- *Challenge 1: lack of a security amplification theorem for DPFs*. First, the work of Goel et al. [GWW25] serves a landmark role as the first and only known OWF-based scheme that simultaneously enjoys sublinear dependence on N and polynomial dependence on m . However, we observe that Goel et al. [GWW25] has an incomplete proof. In particular, their result crucially relies on a security amplification theorem for DPFs in the eprint version of Boyle et al. [BGIK22b]. Upon closer examination, the proofs of the theorem suffer from subtle gaps, also observed by the recent work of Ghoshal et al. [GZSP25] (see their Remark E.1). In this sense, to date, we still lack a security amplification theorem for DPFs. Therefore, the first question we ask is whether we can fill in this gap and prove a security amplification theorem for DPFs

¹Throughout, we use $O_\lambda(\cdot)$ to hide a λ or $\text{poly}(\lambda)$ dependence, where λ denotes the security parameter which is related to the strength of the underlying hardness assumption.

Table 1: Comparison with related work. m denotes the number of parties, t is the corruption threshold, N denotes the input domain size, and ϵ is an arbitrarily small positive constant. For simplicity, we do not distinguish the computational and statistical security parameters in this table. We use $\tilde{O}(\cdot)$ to hide logarithmic or polylogarithmic dependencies on N , λ , or m . We use $O_\lambda(\cdot)$ to hide dependencies on λ or $\text{poly}(\lambda)$.

Scheme	Assumption	Share size	Note
Corrupt majority: $t = m - 1$			
[BGI15]	OWF	$O_\lambda(2^{\frac{m}{2}} \cdot N^{\frac{1}{2}})$	
[CGBM15, KPM24]	DDH	$O_\lambda(N^{\frac{1}{2}})$	not additive share
[DHP26]	DDH	$O_\lambda(N^{\frac{1}{3}} \cdot m^t)$	not additive share
[GWW25]	OWF	$\tilde{O}_\lambda(N^{\frac{1}{2}+\epsilon} \cdot m^3)$	incomplete
[KPR25]	non-standard	$\tilde{O}_\lambda(1)$	not additive share
Our work	OWF	$\tilde{O}_\lambda(N^{\frac{1}{2}+\epsilon} \cdot \sqrt{m})$	
Honest majority: $m > 2t$: let $D = \lfloor m/t \rfloor$, $D' = \lfloor (1 - \epsilon)m/t \rfloor$			
[BKO22]	OWF	$O_\lambda\left(N^{\frac{1}{2D}} \cdot \sqrt{2^{mt}} \cdot m^t\right)$	
[BKO22] + [GWW25]	OWF	$O_\lambda\left(N^{\frac{1}{2D'}} \cdot \binom{m}{t}^3\right)$	[GWW25] incomplete
[BKO22]	none	$O(N^{\frac{1}{D}} \cdot m^t)$	
Our work	OWF	$\tilde{O}_\lambda(N^{\frac{1+\epsilon}{2D'}})$	
Our work	none	$\tilde{O}(N^{\frac{1}{D}})$	

that preserves efficiency — if so, we will be able to re-affirm the theorem statements of Goel et al. [GWW25].

- *Challenge 2: further efficiency improvements.* We ask whether we can further improve the efficiency of prior constructions. For the all-but-one corruption setting, we hope to achieve better scaling than $O(m^3)$ with the number of parties. For the honest-majority setting, prior results [BKO22] that achieve improved dependence on N (relative to the all-but-one corruption setting) work only for (approximately) constant number of parties due to (super-)exponential dependencies on m and t . We ask if it is possible to achieve the same dependence on N without suffering from this strong restriction.

1.1 Our Results and Contributions

Security amplification for DPFs, revisited. We revisit security amplification for DPFs, which Goel et al. [GWW25]’s result crucially relies on. Although Boyle et al. [BGIK22b] made a noteworthy attempt to prove such a security amplification theorem, the proof is incomplete, and we are not aware of straightforward ways to fill in the gap — in Appendix A.2, we provide a detailed explanation of the gaps.

We fill in this gap in our knowledge by presenting a new proof for DPF security amplification (Theorem 1.1). Though inspired by Boyle et al. [BGIK22b], our amplification construction notably differs from theirs, which is essential for making the proof go through. This amplification theorem

serves as an important tool for constructing DPFs: not only does it let us recover the result statement of Goel et al. [GWW25], it also plays a central role in achieving all of our new asymptotic improvements shown in Table 1.

Theorem 1.1 (Security amplification for DPFs). *Let $\delta(\lambda) = O\left(\frac{1}{\text{poly log}(N, \lambda)}\right)$ be a small but possibly non-negligible function in λ . Suppose we are given an m -party DPF with share size S , such that any $t < m$ parties' joint views are $(1 - \delta)$ -computationally indistinguishable from a simulated view. Then, we can construct an m -party DPF with share size $S \cdot \text{poly log}(N, \lambda)$, such that any $t < m$ parties' joint views are computationally indistinguishable from a simulated view.*

To prove Theorem 1.1, we introduce a stronger notion of $(1 - \delta)$ -computational indistinguishability as a key stepping stone, and a generic method for upgrading regular $(1 - \delta)$ -computational indistinguishability to the stronger notion. Our strong $(1 - \delta)$ -indistinguishability and upgrade technique may be of independent interest, as it can facilitate security amplification arguments for a broader class of cryptographic primitives.

Asymptotical improvements for multi-party DPFs. Theorem 1.1 opens up pathways for us to further improve the asymptotical efficiency of DPFs in both the all-but-one corruption and honest-majority settings. Our efficiency improvements are summarized in the following theorem statements and also shown in Table 1.

Theorem 1.2 (Multi-party DPF: all-but-one corruption). *Assume the existence of PRGs. For any number of parties m and domain size N , there exists an m -party DPF scheme secure against up to $m - 1$ corruptions for domain $[N]$ and range $\mathbb{Z}_2$ with per-party share size $\tilde{O}(\sqrt{m\lambda_{\text{PRG}}N^{1+\epsilon}})$, where λ_{PRG} denotes the length of the PRG seed, $\epsilon > 0$ is an arbitrarily small constant, and $\tilde{O}$ hides $\text{poly log}(\lambda_{\text{stat}}, N)$ factors for some statistical security parameter λ_{stat} .*

Theorem 1.3 (Multi-party DPF: honest majority). *Assume the existence of PRG. Let $\epsilon > 0$ be an arbitrarily small constant and $m > (1 + \epsilon)Dt$ for some constant integer $D \geq 2$. Then, there exists an m -party DPF tolerating up to t corruptions with per-party share size $\tilde{O}(\sqrt{\lambda_{\text{PRG}} \cdot N^{(1+\epsilon)/D}})$ where λ_{PRG} denotes the length of a PRG seed, and $\tilde{O}$ hides $\text{poly log}(\lambda_{\text{stat}}, N)$ factors for some statistical security parameter λ_{stat} .*

Theorem 1.4 (Multi-party DPF: honest majority, information-theoretic security). *Suppose that $m \geq Dt + 1$ where $D \geq 2$ is a constant integer. Then, there exists an m -party DPF scheme with perfect security, with per-party share size $O(N^{\frac{1}{D}} \cdot \log m)$.*

Compared to prior state-of-the-art results, our new constructions achieve much better scaling with the number of parties. For the all-but-one corruption setting (Theorem 1.2), we reduce the m^3 dependence in Goel et al. [GWW25] to only $\sqrt{m}$. For the honest-majority setting (Theorem 1.3 and Theorem 1.4), we improve over the state-of-the-art results of Bunn et al. [BKO22] by entirely removing the previously (super-)exponential dependence on m and t . For Theorem 1.4, the result is actually somewhat implicit in the PIR scheme of Park et al. [PLM⁺24] — we extract the underlying DPF implied by their construction and explicitly state the result for completeness.

1.2 Additional Related Work

We now review some additional related work.

Information-theoretic multi-party DPFs. Boyle et al. [BGIK22a] constructed information-theoretic DPFs with sub-polynomial share size $O(N^{o(1)})$ for three or four parties, tolerating a

single corruption. While they claim that existing PIR literature [CGKS95, BIK05] has implicitly used m -party DPF from Reed-Muller code, the construction is under-specified and their corruption threshold is limited to $t = 1$. In contrast, our information-theoretic construction targets the honest-majority regime and supports a corruption threshold that grows with the number of parties.

Multi-party DPFs from stronger assumptions. A line of work constructs multi-party DPFs using assumptions beyond Minicrypt, such as DDH [CGBM15, KPM24, DHP26, CK24], DCR [CK24, ARS24], LPN [CK24, CKY25], MQ [CKY25], and other non-standard assumptions [KPR25]. Among them, several schemes [CGBM15, KPM24, DHP26, KPR25] also require encoding the DPF output as a DDH-group elements, which limits their usability. In comparison, our constructions are either information-theoretically secure or based on one-way functions only, and can output additive shares in arbitrary finite fields.

FSS for related function classes. Beyond point functions, FSS has been studied for richer function families such as decision trees, multi-dimensional intervals, and comparison functions [BGI15, BGI19, BCG⁺21]. These works broaden the functionality supported by FSS and enable applications to secure computation with preprocessing, whereas our focus is on the efficiency and security of multi-party DPFs.

2 Preliminaries

2.1 Notation and Terminology

We use $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{F}_q$ to denote the positive integers, the integers, the real numbers, and the finite field of size q , respectively. For a positive integer N , we write $[N]$ to denote the set $\{1, 2, \dots, N\}$. For $a \leq b \in \mathbb{R}$, we use $[a, b]$ and (a, b) to denote the closed and open intervals from a to b .

Negligible functions. A function $f : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ is negligible, denoted $f(\lambda) = \text{negl}(\lambda)$, if for every polynomial $p(\cdot)$, there exists a threshold $\lambda_0 \in \mathbb{N}$ such that for all $\lambda \geq \lambda_0$, we have $f(\lambda) \leq 1/p(\lambda)$.

Probabilistic polynomial time (PPT). An algorithm is probabilistic polynomial time (PPT) if it is randomized and runs in time polynomial in the size of its input.

Computational indistinguishability. Consider ensembles of distributions $\{X(1^\lambda)\}_{\lambda \in \mathbb{N}}$ and $\{Y(1^\lambda)\}_{\lambda \in \mathbb{N}}$. We say that $X(1^\lambda)$ is computationally indistinguishable from $Y(1^\lambda)$, denoted $X(1^\lambda) \approx Y(1^\lambda)$, if for every non-uniform PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\lambda)$ such that

$$\left| \Pr[\mathcal{A}(1^\lambda, X(1^\lambda)) = 1] - \Pr[\mathcal{A}(1^\lambda, Y(1^\lambda)) = 1] \right| \leq \text{negl}(\lambda).$$

$(1 - \delta)$ -computational indistinguishability. Consider ensembles of distributions $\{X(1^\lambda)\}_{\lambda \in \mathbb{N}}$ and $\{Y(1^\lambda)\}_{\lambda \in \mathbb{N}}$. For $\delta(\lambda) \in [0, 1]$, we say that $X(1^\lambda)$ is $(1 - \delta(\lambda))$ -computationally indistinguishable from $Y(1^\lambda)$, denoted $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$, if for every non-uniform PPT adversary $\mathcal{A}$, for all sufficiently large λ ,

$$\left| \Pr[\mathcal{A}(1^\lambda, X(1^\lambda)) = 1] - \Pr[\mathcal{A}(1^\lambda, Y(1^\lambda)) = 1] \right| \leq \delta(\lambda).$$

Probability measures. Let $\mathcal{U}$ be a finite set. A probability measure $\mathcal{M}$ on $\mathcal{U}$ is a function $\mathcal{M} : \mathcal{U} \rightarrow [0, 1]$. We may abbreviate probability measures to measures.

- The density of a measure $\mathcal{M}$ is defined as

$$\mu(\mathcal{M}) := \frac{1}{|\mathcal{U}|} \sum_{U \in \mathcal{U}} \mathcal{M}(U).$$

- If $\mu(\mathcal{M}) > 0$, the distribution $P_{\mathcal{M}}$ induced by $\mathcal{M}$ has probability mass function

$$P_{\mathcal{M}}(U) := \frac{\mathcal{M}(U)}{\sum_{U \in \mathcal{U}} \mathcal{M}(U)}.$$

- Given a measure $\mathcal{M}$ on $\mathcal{U}$, we define a new measure $1 - \mathcal{M}$ on $\mathcal{U}$ as

$$(1 - \mathcal{M})(U) = 1 - \mathcal{M}(U) \quad \forall U \in \mathcal{U}.$$

Pseudorandom generator (PRG). A pseudorandom generator (PRG) is a function $\text{PRG} : \{0, 1\}^{\lambda_{\text{PRG}}} \rightarrow \{0, 1\}^{\ell}$ where the output length ℓ is greater than the input length λ_{PRG} . A PRG is secure if for every PPT adversary $\mathcal{A}$, there exists a negligible function $\text{negl}(\cdot)$ such that

$$\left| \Pr \left[\mathcal{A}(\text{PRG}(s)) = 1 \mid s \xleftarrow{\$} \{0, 1\}^{\lambda_{\text{PRG}}} \right] - \Pr \left[\mathcal{A}(r) = 1 \mid r \xleftarrow{\$} \{0, 1\}^{\ell} \right] \right| \leq \text{negl}(\lambda),$$

where λ is the security parameter.

2.2 Definitions: Multi-Party Distributed Point Function

We first define point functions. Throughout the paper, Dom and Ran may depend on λ , and we assume that $|\text{Dom}|$ is polynomial in λ .

Definition 1 (Point functions). For a finite set Dom and a group Ran , for any $\alpha \in \text{Dom}$ and $\beta \in \text{Ran}$, we define the point function $f^{\alpha, \beta} : \text{Dom} \rightarrow \text{Ran}$ as $f^{\alpha, \beta}(x) = \begin{cases} \beta & x = \alpha \\ 0 & x \neq \alpha \end{cases}$.

We next define multi-party distributed point functions (DPFs).

Definition 2 (Multiparty DPF). A distributed point function (DPF), parametrized by the number of parties $m \in \mathbb{N}$, the corruption threshold $t \in \mathbb{N}$, an input domain Dom and an output range Ran that supports addition, is a pair of PPT algorithms $(\text{Gen}, \text{Eval})$ with the following syntax:

- $\text{Gen}(1^\lambda, \alpha, \beta)$: On input the security parameter 1^λ and function parameters $\alpha \in \text{Dom}$, $\beta \in \text{Ran}$, the share generation algorithm outputs m shares $(k_1, \dots, k_m)$.
- $\text{Eval}(k, x)$: On input a share k and a point $x \in \text{Dom}$, the evaluation algorithm outputs a value $y \in \text{Ran}$, corresponding to that party's output share.

We define the following properties for these two algorithms.

$(1 - \gamma)$ -correctness. For all $x, \alpha \in \text{Dom}$, $\beta \in \text{Ran}$, and $\lambda \in \mathbb{N}$, it holds that

$$\Pr \left[(k_1, \dots, k_m) \leftarrow \text{Gen}(1^\lambda, \alpha, \beta); \sum_{i=1}^m \text{Eval}(k_i, x) = f^{\alpha, \beta}(x) \right] \geq 1 - \gamma(\lambda)$$

where $f^{\alpha, \beta}(\cdot)$ is the point function for $\alpha \in \text{Dom}$ and $\beta \in \text{Ran}$. Observe that the correctness definition implicitly requires that the reconstruction function be additive over Ran .

$(1 - \delta)$ -security. For any subset of at most t corrupted parties $T \subseteq [m]$, any $\alpha_0, \alpha_1 \in \text{Dom}$ and $\beta_0, \beta_1 \in \text{Ran}$, the following two distributions are $(1 - \delta)$ -computationally indistinguishable:

$$\{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_0, \beta_0)\} \approx_{\delta(\lambda)} \{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_1, \beta_1)\}.$$

Perfect security. For any subset of at most t corrupted parties $T \subseteq [m]$, any $\alpha_0, \alpha_1 \in \text{Dom}$ and $\beta_0, \beta_1 \in \text{Ran}$, for any $\lambda \in \mathbb{N}$, the following two distributions are identical:

$$\{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_0, \beta_0)\} = \{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_1, \beta_1)\}.$$

3 Security Amplification of Distributed Point Functions

Why we still lack a DPF amplification theorem. Suppose we have a DPF with some non-negligible δ leakage. More formally, suppose that with the underlying DPF, no probabilistically polynomial-time (PPT) adversary can distinguish the corrupt parties' shares from simulated shares with more than δ probability — a notion henceforth called *regular $(1 - \delta)$ -security*. To prove security amplification, it would be convenient to instead interpret $(1 - \delta)$ -security as:

With at most δ probability, completely leak the secret; and with at least $1 - \delta$ probability, protect the secret from any computationally bounded adversary. (★)

We will later formalize the above intuition as *strong $(1 - \delta)$ -security* from a secret-independent, simulated distribution. If such an interpretation is possible, then we can rely on the Reed-Muller (RM) based construction of Boyle et al. [BGIK22b] to achieve security amplification. Boyle et al. [BGIK22b] states in Lemma 3 of their paper that indeed, *regular $(1 - \delta)$ -security* for DPFs can be interpreted as with *strong $(1 - \delta)$ -security*. In Appendix A.2, we prove that this statement is incorrect.

3.1 Strong Approximate Computational Indistinguishability

To prove a DPF security amplification theorem, we formally define a notion called *strong $(1 - \delta)$ -computational indistinguishability*. Informally, if the probability ensembles $X(1^\lambda)$ and $Y(1^\lambda)$ satisfy strong $(1 - \delta)$ -computational indistinguishability, it means that X is computationally indistinguishable from an intermediate distribution Z , where Z samples from Y with probability $1 - \delta$, and with the remaining δ probability, Z samples from an arbitrary residual distribution. In particular, when $X = \text{Enc}(1^\lambda, s)$ is an encoding of some secret s and $Y = \text{Sim}(1^\lambda)$ is some ideal distribution that does not depend on s , our definition is another way to capture the intuition that Boyle et al. [BGIK22b] wanted, that the secret s is protected with probability $1 - \delta$ and may be completely leaked with probability δ .

Ghoshal et al. [GZSP25] also proposed a similar strong $(1 - \delta)$ -computational indistinguishability notion, but our definition is a slight weakening of theirs by ordering the quantifiers slightly differently. More specifically, the notion in Ghoshal et al. [GZSP25] requires the intermediate distribution Z to be the same for all adversary sizes and advantages, but this is in fact too restrictive for technical reasons. In comparison, our definition allows Z to depend on the adversary size and advantage, as formally stated below:

Definition 3 (Strong approximate computational indistinguishability). We say that the ensemble of distributions $X(1^\lambda)$ is strongly $(1 - \delta(\lambda))$ -computationally indistinguishable from $Y(1^\lambda)$, denoted

by $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$, if for all polynomials $P(\cdot)$ and $Q(\cdot)$, there exists an ensemble $Z(1^\lambda)$ such that for all sufficiently large λ , for all non-uniform adversary $\mathcal{A}$ of size $P(\lambda)$,

$$\text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), Z(1^\lambda)) \leq 1/Q(\lambda),$$

and $Z(1^\lambda)$ can be equivalently rewritten as: with probability $1 - \delta(\lambda)$, output a random sample from $Y(1^\lambda)$; else output a random sample from any residual distribution (which need not be efficiently samplable).

For convenience, we sometimes also refer to strong $(1 - \delta)$ -computational indistinguishability simply as *strong $(1 - \delta)$ -indistinguishability*. For the special case where $Z(1^\lambda)$ has negligible statistical distance from $X(1^\lambda)$, we also say that $X(1^\lambda)$ and $Y(1^\lambda)$ satisfy strong $(1 - \delta)$ -*statistical* indistinguishability.

3.2 The Decoy Lemma: Upgrade from Regular to Strong $(1 - \delta)$ -Indistinguishability

We want to prove a security amplification theorem for a DPF that satisfies regular $(1 - \delta)$ -indistinguishability. To do so, we need to first upgrade regular $(1 - \delta)$ -indistinguishability to strong $(1 - \delta)$ -indistinguishability. Unfortunately, as pointed out in Fact A.9, regular $(1 - \delta)$ -indistinguishability does not directly imply strong $(1 - \delta)$ -indistinguishability in general. Below, we prove a “decoy lemma”, showing how adding decoys suffices for upgrading two distributions that satisfy only regular $(1 - \delta)$ -indistinguishability to two modified distributions that are strongly $(1 - \delta)$ -indistinguishable.

Intuitively, when D and D' are $(1 - \delta)$ -computationally indistinguishable, D may take values that D' does not. Strong computational indistinguishability $D' \approx_\delta D$ requires an intermediate distribution that outputs from D with probability $1 - \delta$, so these D -only values must be added in some way. The decoy lemma addresses this problem by hiding one copy of D' among many copies of D . Even if D' omits some values, the decoy copies of D are likely to cover them, thereby hiding the presence of the single D' sample.

Remark 1. Our decoy lemma is otherwise the same as Lemma E.4 of Ghoshal et al. [GZSP25], who used it to amplify the security of Private Information Retrieval (PIR) schemes — except that we adopt a different order of quantifiers in our definition of strong $(1 - \delta)$ -indistinguishability. Unfortunately, Ghoshal et al. [GZSP25]’s proof of their version is incomplete due to incorrect treatment of the order of quantifiers, and we are not aware of any approach to prove their version correct. Instead, we reprove the decoy lemma using a slightly weaker notion of strong $(1 - \delta)$ -indistinguishability which is nonetheless sufficient for security amplification of DPFs. The weaker notion of strong $(1 - \delta)$ -indistinguishability also suffices for [GZSP25] and hence their results still hold, but we omit the proof here.

Lemma 3.1 (Decoy lemma). *Let $\ell(\cdot)$ be a polynomially bounded function. Suppose that $\{D(1^\lambda, r) : r \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}\}$ and $\{D'(1^\lambda, r) : r \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}\}$ are $(1 - \delta(\lambda))$ -computationally indistinguishable, where $1/\delta(\lambda) \leq \text{poly}(\lambda)$. Then, for $d(\lambda) := (\log \lambda)^2 / \delta(\lambda)^2$ (abbreviated as d), the following ensemble $X'(1^\lambda)$ is strongly $(1 - 2\delta(\lambda))$ -computationally indistinguishable from $X(1^\lambda)$:*

- $X(1^\lambda)$: sample $r_1, \dots, r_d \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}$ and output the **unordered** set $\text{Set}(D(1^\lambda, r_1), D(1^\lambda, r_2), \dots, D(1^\lambda, r_d))$;
- $X'(1^\lambda)$: sample $r_1, \dots, r_d \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}$ and output the **unordered** set $\text{Set}(D(1^\lambda, r_1), \dots, D(1^\lambda, r_{d-1}), D'(1^\lambda, r_d))$.

Proof. Deferred to Appendix A.4. □

3.3 Security Amplification for DPFs

To prove a security amplification theorem for DPFs, we first use the decoy lemma to upgrade a DPF with *regular* $(1 - \delta)$ -security to one that satisfies *strong* $(1 - \delta)$ -security as defined below:

Definition 4 (Strongly $(1 - \delta)$ -secure DPFs). We say that an m -party DPF (with domain Dom and range Ran) satisfies strong $(1 - \delta)$ -security against t corruptions iff the following holds. For any subset of t corrupted parties $T \subseteq [m]$, there exists a (not necessarily efficient) simulator $\text{Sim}(1^\lambda)$ such that for any $\alpha \in \text{Dom}$ and $\beta \in \text{Ran}$, the following distributions are strongly $(1 - \delta)$ -computationally indistinguishable:

$$\{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha, \beta)\} \quad \approx_\delta \quad \{(k_i)_{i \in T} \xleftarrow{\$} \text{Sim}(1^\lambda)\}.$$

We then apply the Reed-Muller-code based upgrade construction of Boyle et al. [BGIK22b] to the upgraded DPF that satisfies Definition 4, resulting in Theorem 3.2. Imprecisely speaking, the intuition is as follows. Evaluating the DPF at some input x can be equivalently viewed as retrieving the α -th entry of a one-hot vector of length N where only the x -th coordinate is non-zero. By applying an RM-encoding on the input x , we can secret share the query point α into multiple queries $\alpha_1, \dots, \alpha_q$, such that evaluating the original DPF can be accomplished by evaluating multiple DPF instances that encode the secrets $\alpha_1, \dots, \alpha_q$, respectively. If each DPF instance satisfies strong $(1 - \delta)$ -security, we can indeed apply the intuition marked as $(\star)$ earlier, and classify the DPF instances into *broken* ones and *intact* ones. By the smoothness property of the RM code, as long as sufficiently many DPF instances remain intact, the adversary learns nothing about α even if every the broken instance i directly leaks its α_i value to the adversary.

Theorem 3.2. *For an arbitrarily small constant $0 < \epsilon < 1$, a domain size $N \in \mathbb{N}$, and security parameter λ , define the parameter $N' := (\max\{N, (\log \lambda)^{3/\epsilon}\})^{1+\epsilon}$. There exists a security threshold $K = (\log(N, \lambda))^{O(1/\epsilon)}$ such that the following holds.*

Suppose that there exists an m -party DPF tolerating up to t corrupt parties for domain $[N']$ and range $\mathbb{Z}_2$ with the following properties:

- $(1 - \text{negl}(\lambda))$ -correctness;
- $(1 - 1/K)$ -security; and
- $S(m, t, [N'], \mathbb{Z}_2)$ per-party share size.

Then we can construct an m -party DPF tolerating up to t corrupt parties for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:

- $(1 - \text{negl}(\lambda))$ -correctness;
- $(1 - \text{negl}(\lambda))$ -security; and
- $(\log(N, \lambda))^{O(1/\epsilon)} \cdot S(m, t, [N'], \mathbb{Z}_2)$ per-party share size.

Proof. Deferred to Appendix A.5, Appendix A.6, and Appendix A.7. □

4 Improved DPF: All-But-One Corruption Setting

Applying Theorem 3.2 to Goel et al. [GWW25] can recover their theorem statement, i.e., a DPF in the all-but-one-corruption setting with $\tilde{O}_\lambda(N^{\frac{1}{2}+\epsilon} \cdot m^3)$ share size. In this section, we asymptotically

improve their result in the all-but-one corruption setting. In comparison with Goel et al. [GWW25], we improve the dependence on m from m^3 to only $\sqrt{m}$. Our improvement arises from the following sources. First, we present a novel combinatorial design of random bipartite graphs that achieves asymptotical improvements over that of Goel et al. [GWW25] (Section 4.2). Second, we improve the transformation from the random bipartite graphs to DPFs through a rebalancing trick introduced by Boyle et al. [BGI15] (Appendix B.2).

4.1 Background: DPFs from Random Bipartite Graphs

Goel et al. [GWW25] constructed a DPF based on special randomized bipartite graphs.

Dividing the domain into segments. We first partition the domain $[N]$ into *segments* of length $\sqrt{N}$. Each point $x \in [N]$ can thus be expressed as a pair $(\text{seg}, \text{offset}) \in [\sqrt{N}]^2$ where $\text{seg} \in [\sqrt{N}]$ denotes the *segment number* and $\text{offset} \in [\sqrt{N}]$ denotes the offset within the segment. Recall that α is the non-zero point of the point function $f^{\alpha, \beta}(\cdot)$. Henceforth, we call the segment that contains α the *target segment*, and call all other segments the non-target segments.

Assignment of PRG seeds to parties per segment. For each segment, we will independently sample a number of PRG seeds, and assign the seeds to parties according to a bipartite graph where the *left vertices denote parties* and the *right vertices denote PRG seeds*. Moreover, each party i is given the same pseudo-random correction word denoted $\text{cw} \in \{0, 1\}^{\sqrt{N}}$, along with a random mask bit $\rho_i^{(\text{seg})} \in \{0, 1\}$ for each segment seg . It is guaranteed that for every non-target segment seg , the $\{\rho_i^{(\text{seg})}\}_{i \in [m]}$ values are secret shares of 0; and for the target segment seg , they form secret shares of 1.

Evaluation. Let $x = (\text{seg}, \text{offset})$ denote the desired input where seg denotes the segment number and offset denotes the offset. For each party $i \in [m]$, let $S_i^{(\text{seg})}$ denote the set of all PRG seeds it has received for the segment seg . The party computes and outputs the bit at index offset of the following evaluation result

$$\rho_i^{(\text{seg})} \cdot \text{cw} + \sum_{\text{sk} \in S_i^{(\text{seg})}} \text{PRG}(\text{sk})$$

Specifically, we choose cw and the bipartite graphs in the following manner to guarantee correctness:

- After determining the PRG keys, we have enough degree of freedom to choose cw to guarantee that the evaluation outcome is correct for the target segment.
- The bipartite graphs are chosen in a way such that for every non-target segment, all parties' PRG evaluation outcomes cancel out.

Desired properties of the bipartite graphs. To assign PRG seeds to parties, we use a pair of distributions on random bipartite graphs henceforth denoted G_0 and G_1 , where each non-target segment uses a freshly sampled instance of G_0 , and the target segment uses a fresh instance of G_1 . For the resulting DPF scheme to satisfy correctness and security, we want the pair of bipartite graphs to satisfy the following properties:

- **Right vertices in G_0 have even degrees with probability $1 - \text{negl}(\lambda)$.** This guarantees that for all non-target segments, all parties PRG evaluation outcomes cancel out under summation.
- **Left vertices in G_1 have exclusive neighbors with probability $1 - \text{negl}(\lambda)$.** This ensures that even when the adversary controls $m - 1$ parties, the correction word cw is uncorrelated with the adversary's view, which is necessary for protecting which segment α belongs to.

- **$(1 - \delta)$ -indistinguishability of subgraphs of G_0 and G_1 .** Suppose that the adversary observes the subgraphs of G_0 and G_1 induced by an arbitrary subset of up to $m - 1$ left vertices and their neighbors. We want the adversary's views of G_0 and G_1 to be indistinguishable with $1 - \delta$ probability where δ is a small but possibly non-negligible function in λ . This ensures that the adversary cannot effectively distinguish which one is the target segment by examining the graph structures.

In summary, Goel et al. [GWW25] proves the following lemma that transforms a bipartite graph design satisfying the aforementioned properties into a DPF scheme with $(1 - \delta)$ -security. The resulting DPF scheme's share size is directly related to the degree of the left vertices in the bipartite graphs.

Lemma 4.1 (Transformation from special bipartite graph design to DPFs [GWW25]). *Suppose there exists a pair of distributions on bipartite graphs (G_0, G_1) satisfying the aforementioned properties. Further, suppose G_0 and G_1 have m number of left vertices, and each left vertex has degree at most d . Further, assume the existence of PRG with seed length λ_{PRG} . Then, there exists an m -party DPF for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:*

- $(1 - \text{negl}(\lambda))$ -correctness;
- $(1 - \delta - \text{negl}(\lambda))$ -security in the presence of up to $m - 1$ corruptions; and
- $O(d\lambda_{\text{PRG}}\sqrt{N})$ per-party share size.

Moreover, Goel et al. [GWW25] constructed such a special bipartite graph design with left degree upper bounded by $\tilde{O}(m^3/\delta^2)$. Combined with Lemma 4.1, this leads to a $(1 - \delta)$ -secure DPF with share size $\tilde{O}(\lambda_{\text{PRG}} \cdot m^3 \cdot \sqrt{N}/\delta^2)$.

4.2 Improved Random Bipartite Graph Design

In this section, we present a more efficient random bipartite graph design. Compared to the construction of [GWW25], we reduce the sizes of G_0 and G_1 by a factor of $\tilde{O}(m^2/\delta)$.

Why Goel et al.'s design is inefficient. Goel et al. [GWW25] designs the distributions G_0 and G_1 as follows. In G_0 , each right vertex samples two left vertices as neighbors. G_1 is otherwise identical as G_0 except that it plants an appropriate number of degree-1 vertices with only one random left-vertex as neighbor. Now suppose that the adversary controls all but one parties. In the adversary's view, each degree-2 right vertex in G_0 with exactly one edge to the honest party will appear as degree-1 right vertex, thereby hiding the presence of the planted right vertices in G_1 . Thus, as long as there is sufficient variance in the number of right vertices in G_0 connected to the honest party, the adversary's views G_0 and G_1 have small statistical distance. However, Goel et al.'s design is inefficient partly because $\tilde{\Omega}(m)$ degree-1 right vertices must be planted so that every left vertex has an exclusive right neighbor. In their construction, this entire $\tilde{\Omega}(m)$ difference must be masked (up to a small statistical difference) with a binomial distribution, requiring the binomial distribution's mean to be as large as m^2 .

Our improved construction. Our construction takes a more structured approach. Instead of randomly planting degree-1 vertices, we deterministically assign an exclusive right neighbor to every left vertex in G_1 . As shown in Figure 1, to hide the m planted vertices in G_1 , for every party i , we sample a random variable X_i , and we create a group $s_{i,j}$ of X_i fresh degree-2 right vertices for every $j \in [m] \setminus \{i\}$. Every vertex in $s_{i,j}$ is connected to both left vertices P_i and P_j .

Suppose that the adversary controls all but party i . Then the vertices in $s_{i,j}$ appear as degree-1 vertices to the adversary, so the m planted vertices appear as if they are one additional vertex in $s_{i,j}$.

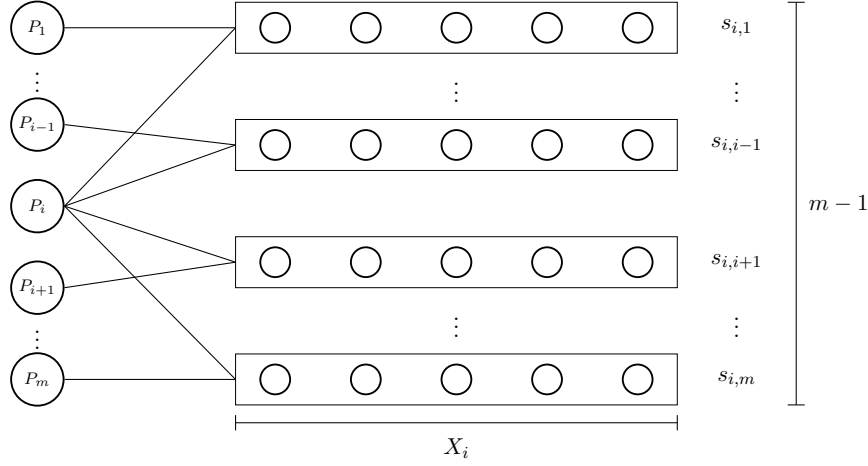


Figure 1: Our new G_0 : showing only a subset of G_0 involving the contributions from $s_{i,1}, \dots, s_{i,m}$. In the entire G_0 , the same is repeated for every $i \in [m]$. We sample $X_i \sim \text{RTGeo}(W, 1 - \delta)$ and create $m - 1$ sets of new vertices $s_{i,j}$ for $j \in [m] \setminus \{i\}$, each of size X_i . The right vertices in $s_{i,j}$ are connected to both P_i and P_j . The graph G_1 is obtained in the same way as G_0 , except that it has m additional degree-1 right vertices, each connected to a different left vertex.

Therefore, distinguishing the subgraphs of G_0 and G_1 with P_i removed reduces to distinguishing X_i from $X_i + 1$. In reality, we do not know which left vertex will be honest, so we repeat this construction symmetrically for every $i \in [m]$. Namely, for every i , we sample an independent random variable X_i and create groups $s_{i,j}$ of size X_i for all $j \neq i$. Then, distinguishing G_0 and G_1 reduces to distinguishing $(X_1, \dots, X_m)$ and $(X_1, \dots, X_h + 1, \dots, X_m)$ where h is the index of the honest vertex.

This also demonstrates the crux of our improvement over Goel et al. [GWW25]. Although both constructions plant m or $\tilde{O}(m)$ degree-1 right vertices in G_1 , our construction only needs to mask the difference of one between exactly one pair of random variables X_h and $X_h + 1$. For example, it suffices to sample X_i from a uniform distribution over $[0, 1/\delta)$ to get $(1 - \delta)$ -indistinguishability of the bipartite graphs. Below, we show that if we instead sample the X_i variables from a more clever, geometric-like distribution (Definition 5), we can directly prove that the subsets of G_0 and G_1 in the adversary's view are *strongly* $(1 - \delta)$ -indistinguishable, which avoids the extra polylogarithmic overhead of applying the decoy lemma in the final security amplification step.

Definition 5 (Reverse truncated geometric distribution). For $W \in \mathbb{N}$ and $p \in (0, 1)$, the discrete probability distribution $\text{RTGeo}(W, p)$ is defined on support $\{0, 1, \dots, W\}$ by the probability mass function

$$\Pr[X = 0] = p^W, \quad \Pr[X = x] = (1 - p)p^{W-x} \quad \forall x = 1, \dots, W.$$

$\text{RTGeo}(W, p)$ is a valid distribution because $\sum_{x \in [W]} \Pr[X = x] = p^W + (1 - p) \sum_{x=1}^W p^{W-x} = p^W + (1 - p) \cdot \frac{1 - p^W}{1 - p} = 1$.

Our new random bipartite graph design. We construct the distributions G_0 and G_1 as follows.

- Graph G_0 :** Start with m left vertices P_i for $i \in [m]$ and no right vertices. For every $i \in [m]$:
1. Sample $X_i \sim \text{RTGeo}(W, 1 - \delta)$, where $W := (\log \lambda)^2 / \delta$.
 2. Add $(m-1)X_i$ new right vertices, and group them into a total of $m-1$ sets $s_{i,j}$ for $j \in [m] \setminus \{i\}$, each with X_i vertices.
 3. For every $j \in [m] \setminus \{i\}$, connect all X_i vertices in $s_{i,j}$ to both P_i and P_j .
- Graph G_1 .** Construct G_0 , and additionally add m new right vertices s'_i for $i \in [m]$, each connected only to the corresponding left vertex P_i .

Theorem 4.2. *Fix an arbitrary $\delta \in (0, 1)$ and security parameter λ , the above distributions G_0 and G_1 satisfy the aforementioned properties and each left vertex has degree at most $O(m(\log \lambda)^2 / \delta)$.*

Proof. Deferred to Appendix B.1. □

4.3 Improved Transformation from Random Bipartite Graphs to DPFs

We improve Goel et al. [GWW25]’s transformation from random bipartite graphs to DPFs. Specifically, through the introduction of a rebalancing trick we can asymptotically improve the parameters of the transformation. We defer the details to Appendix B.2. This new transformation in combination with Theorem 4.2 gives rise to the following theorem:

Theorem 4.3. *For an arbitrarily small constant $0 < \epsilon < 1$, number of parties m , domain size N , and security parameter λ , suppose that we have a PRG $\text{PRG} : \{0, 1\}^{\lambda_{\text{PRG}}} \rightarrow \{0, 1\}^{\text{poly}(\log \lambda, m, \lambda_{\text{PRG}}, N)}$. Then we can construct an m -party, $(m-1)$ -secure DPF for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:*

- 1-correctness;
- $(1 - \text{negl})$ -security; and
- $(\log(N, \lambda))^{O(1/\epsilon)} \sqrt{m \lambda_{\text{PRG}} N^{1+\epsilon}}$ per-party share size.

5 Improved DPF: Honest-Majority Setting with Computational Security

In this section, we consider a setting where the fraction of corrupted parties $t/m < \frac{1}{2}$. In this setting, Bunn et al. [BKO22] showed that assuming OWF, one can get an honest-majority DPF with $O(N^{1/4} \cdot \sqrt{2^{m^t}} \cdot m^t)$ share size. Their result asymptotically improves the dependence on the domain size N than the all-but-one corruption setting, but unfortunately suffers from (super-)exponential dependence on the number of parties m or corruption threshold t . We improve the result of Bunn et al. [BKO22] by achieving the same dependence on N but removing the (super-)exponential dependence on m or t . More specifically, for the almost honest majority setting $m > (1 + \epsilon) \cdot 2t$, the per-share size is bounded by $\text{poly} \log(N, \lambda) \cdot N^{1/4+\epsilon}$, where ϵ is an arbitrarily small positive constant. More generally, we have the following theorem:

Theorem 5.1 (Improved DPF in the honest majority setting). *Assume the existence of OWF. Let $\epsilon \in (0, 1)$ be a suitably small positive constant and let $D \geq 2$ be a constant integer. Then,*

there exists an m -party DPF for input domain size N and a binary output domain, tolerating up to $t \leq (1 - \epsilon) \cdot \frac{1}{D} \cdot m$ corruptions, and achieving per-share size

$$\text{poly log}(N, \lambda) \cdot \sqrt{\lambda_{\text{PRG}}} \cdot N^{(1+\epsilon) \cdot \frac{1}{2D}}.$$

Intuition. To prove Theorem 5.1, we proceed in the following steps.

1. **Apply the compiler of Bunn et al. to our all-but-one-secure scheme.** Henceforth let $D \geq 2$ be an integer. Bunn et al. [BKO22] proved that given an all-but-one secure DPF for a smaller domain of size $n = N^{1/(2D)}$ but a larger number $M = \binom{m}{t}$ parties, we can construct a DPF for domain size N and $m > D \cdot t$ parties, tolerating up to t corruptions, with share size $D \cdot M \cdot S(M, n)$. Notably, the resulting (honest-majority) DPF enjoys a significantly better dependence on the domain size N but exponential dependence on the number of parties m or corruption threshold t . Our first step is to apply the compiler of Bunn et al. [BKO22] to our all-but-one-secure DPF of Section 4. This results in an honest-majority DPF with share size $\tilde{O}_\lambda(D) \cdot \sqrt{\binom{m}{t}} \cdot N^{\frac{1+\epsilon}{2D}}$.
2. **Weakly secure committee selection.** The above still suffers from an exponential dependence on m and t . To remove this dependence, we randomly select a committee among the entire population, to make m and t smaller. Unfortunately, the standard approach of committee selection used in the multi-party computation and consensus literature [BCP15, ACD⁺19] fails in our case for the following reason. Suppose the overall population is 51% honest, then we need to select $\text{poly log}(\lambda)$ committee members for the committee to satisfy honest majority except with negligible probability. However, plugging in a polylogarithmically large m and t , the cost of the resulting scheme remains super-polynomial.

Our idea is to select a much smaller committee, on the order of $\Theta(\log \log \lambda)$ size. This will introduce an extra $1/\text{poly log } \lambda$ security loss into the scheme. Fortunately, this $1/\text{poly log } \lambda$ leakage can be plugged with our security amplification theorem, leading to the final step described below.
3. **Security amplification.** Finally, we apply our new security amplification theorem (Theorem 3.2) to the weakly secure scheme resulting from committee selection, and obtain Theorem 5.1.

Formal proof of Theorem 5.1. We defer the formal proof of Theorem 5.1 to Appendix C. In this formal description, we further simplify the construction and avoid the application of the decoy lemma, by directly proving that applying Bunn et al.’s compiler [BKO22] to a DPF that satisfies *strong* $(1 - \delta)$ -security results in a DPF that satisfies *strong* $(1 - D\delta)$ -security assuming $m > D \cdot t$. Further, committee selection also preserves the strong approximate security of the DPF scheme.

Acknowledgements

We used AI tools for language polishing, including grammar and spelling checks and basic editing. The AI tools include ChatGPT and Codex. This work is in part supported by an ONR grant, a Packard Fellowship, a Cisco gift, and a gift from the late Nikolai Mushegian. We acknowledge helpful technical discussions with Elette Boyle and Aayush Jain.

References

- [ACD⁺19] Ittai Abraham, T-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, page 317–326, New York, NY, USA, 2019. Association for Computing Machinery.
- [ARS24] Damiano Abram, Lawrence Roy, and Peter Scholl. Succinct homomorphic secret sharing. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 301–330. Springer, 2024.
- [BCG⁺19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent ot extension and more. In *Advances in Cryptology - CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III*, page 489–518, Berlin, Heidelberg, 2019. Springer-Verlag.
- [BCG⁺20] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators from ring-lpn. In *Advances in Cryptology - CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II*, page 387–416, Berlin, Heidelberg, 2020. Springer-Verlag.
- [BCG⁺21] Elette Boyle, Nishanth Chandran, Niv Gilboa, Divya Gupta, Yuval Ishai, Nishant Kumar, and Mayank Rathee. Function secret sharing for mixed-mode and fixed-point secure computation. In *Advances in Cryptology - EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part II*, page 871–900, Berlin, Heidelberg, 2021. Springer-Verlag.
- [BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector ole. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS '18*, page 896–912, New York, NY, USA, 2018. Association for Computing Machinery.
- [BCP15] Elette Boyle, Kai-Min Chung, and Rafael Pass. Large-scale secure computation: Multi-party computation for (parallel) RAM programs. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 742–762. Springer, 2015.
- [BDSS25] Elette Boyle, Lalita Devadas, and Sacha Servan-Schreiber. Non-interactive distributed point functions. In *Public-Key Cryptography - PKC 2025: 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12–15, 2025, Proceedings, Part I*, page 3–35, Berlin, Heidelberg, 2025. Springer-Verlag.
- [BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In Elisabeth Oswald and Marc Fischlin, editors, *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, pages 337–367. Springer, 2015.
- [BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16*, page 1292–1303, New York, NY, USA, 2016. Association for Computing Machinery.
 - [BGI19] Elette Boyle, Niv Gilboa, and Yuval Ishai. Secure computation with preprocessing via function secret sharing. In *Theory of Cryptography: 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part I*, page 341–371, Berlin, Heidelberg, 2019. Springer-Verlag.
 - [BGIK22a] Elette Boyle, Niv Gilboa, Yuval Ishai, and Victor I. Kolobov. Information-theoretic distributed point functions. In Dana Dachman-Soled, editor, *3rd Conference on Information-Theoretic Cryptography, ITC 2022, Cambridge, MA, USA, July 5-7, 2022, LIPIcs*, pages 17:1–17:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
 - [BGIK22b] Elette Boyle, Niv Gilboa, Yuval Ishai, and Victor I. Kolobov. Programmable distributed point functions. In *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV*, page 121–151, Berlin, Heidelberg, 2022. Springer-Verlag.
 - [BIK05] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. Comput. Syst. Sci.*, 71(2):213–247, 2005.
 - [BKKO20] Paul Bunn, Jonathan Katz, Eyal Kushilevitz, and Rafail Ostrovsky. Efficient 3-party distributed oram. In *Security and Cryptography for Networks: 12th International Conference, SCN 2020, Amalfi, Italy, September 14–16, 2020, Proceedings*, page 215–232, Berlin, Heidelberg, 2020. Springer-Verlag.
 - [BKO22] Paul Bunn, Eyal Kushilevitz, and Rafail Ostrovsky. Cnf-fss and its applications. In *Public-Key Cryptography – PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part I*, page 283–314, Berlin, Heidelberg, 2022. Springer-Verlag.
 - [BS24] Katharina Boudgoust and Mark Simkin. The power of naps: - compressing or-proofs via collision-resistant hashing. In *Theory of Cryptography - 22nd International Conference, TCC 2024, Milan, Italy, December 2-6, 2024, Proceedings, Part I*, Lecture Notes in Computer Science, pages 35–66. Springer, 2024.
 - [CGBM15] Henry Corrigan-Gibbs, Dan Boneh, and David Mazières. Riposte: An anonymous messaging system handling millions of users. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy, SP '15*, page 321–338, USA, 2015. IEEE Computer Society.
 - [CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 41–50, 1995.

- [CK24] Geoffroy Couteau and Naman Kumar. 10-party sublinear secure computation from standard assumptions. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part IX*, page 39–73, Berlin, Heidelberg, 2024. Springer-Verlag.
- [CKY25] Geoffroy Couteau, Naman Kumar, and Xiaxi Ye. Multiparty homomorphic secret sharing and more from LPN and MQ. In Benny Applebaum and Huijia (Rachel) Lin, editors, *Theory of Cryptography - 23rd International Conference, TCC 2025, Aarhus, Denmark, December 1-5, 2025, Proceedings, Part II*, Lecture Notes in Computer Science, pages 498–530. Springer, 2025.
- [DHPR26] Marc Damie, Florian Hahn, Andreas Peter, and Jan Ramon. Ddh-based schemes for multi-party function secret sharing. In Raimundas Matulevičius, Liina Kamm, and Mubashar Iqbal, editors, *Secure IT Systems*, pages 3–22, Cham, 2026. Springer Nature Switzerland.
- [DIL⁺20] Samuel Dittmer, Yuval Ishai, Steve Lu, Rafail Ostrovsky, Mohamed Elsabagh, Nikolaos Kiourtis, Brian Schulte, and Angelos Stavrou. Function secret sharing for PSI-CA: with applications to private contact tracing. *CoRR*, abs/2012.13053, 2020.
- [DS17] Jack Doerner and Abhi Shelat. Scaling oram for secure computation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS ’17*, page 523–535, New York, NY, USA, 2017. Association for Computing Machinery.
- [GGM24] Gayathri Garimella, Benjamin Goff, and Peihan Miao. Computation efficient structure-aware psi from incremental function secret sharing. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, CRYPTO 2024, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VIII*, page 309–345, Berlin, Heidelberg, 2024. Springer-Verlag.
- [GI14] Niv Gilboa and Yuval Ishai. Distributed point functions and their applications. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EURO-CRYPT 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, pages 640–658. Springer, 2014.
- [GRS22] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I*, page 323–352, Berlin, Heidelberg, 2022. Springer-Verlag.
- [GRS23] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In *Advances in Cryptology – CRYPTO 2023: 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20–24, 2023, Proceedings, Part I*, page 577–610, Berlin, Heidelberg, 2023. Springer-Verlag.
- [GWW25] Aarushi Goel, Mingyuan Wang, and Zhiheng Wang. Multiparty distributed point functions. In *Advances in Cryptology – CRYPTO 2025: 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2025, Proceedings, Part IV*, pages 140–173, Berlin, Heidelberg, 2025. Springer-Verlag.

- [GZSP25] Ashrujit Ghoshal, Mingxun Zhou, Elaine Shi, and Bo Peng. Pseudorandom functions with weak programming privacy and applications to private information retrieval. In *Advances in Cryptology – EUROCRYPT 2025: 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4–8, 2025, Proceedings, Part VII*, page 284–313, Berlin, Heidelberg, 2025. Springer-Verlag.
- [Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [JGB⁺24] Neha Jawalkar, Kanav Gupta, Arkaprava Basu, Nishanth Chandran, Divya Gupta, and Rahul Sharma. Orca: Fss-based secure training and inference with gpus. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19–23, 2024*, pages 597–616. IEEE, 2024.
- [KPM24] Chandan Kumar, Sikhar Patranabis, and Debdeep Mukhopadhyay. Compact key function secret sharing with non-linear decoder. *IACR Communications in Cryptology*, 1(2), 2024.
- [KPR25] Toomas Krips and Pille Pullonen-Raudvere. Multi-party distributed point functions with polylogarithmic key size from invariants of matrices. Cryptology ePrint Archive, Paper 2025/978, 2025.
- [MT10] Ueli Maurer and Stefano Tessaro. A hardcore lemma for computational indistinguishability: Security amplification for arbitrarily weak prgs with optimal stretch. In Daniele Micciancio, editor, *Theory of Cryptography*, pages 237–254, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [PLM⁺24] Andrew Park, Trevor Leong, Francisco Maturana, Wenting Zheng, and K. V. Rashmi. Communication-efficient, Fault Tolerant PIR over Erasure Coded Storage . In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4331–4347, May 2024.
- [RTPB20] Théo Ryffel, Pierre Tholoniati, David Pointcheval, and Francis R. Bach. Ariann: Low-interaction privacy-preserving deep learning via function secret sharing. *Proceedings on Privacy Enhancing Technologies*, 2022, 2020.
- [TSS⁺20] Ni Trieu, Kareem Shehata, Prateek Saxena, Reza Shokri, and Dawn Song. Epione: Lightweight contact tracing with strong privacy. *IEEE Data Eng. Bull.*, 43:95–107, 2020.
- [YJG⁺24] Peng Yang, Zoe Lin Jiang, Shiqi Gao, Hongxiao Wang, Jun Zhou, Yangyiye Jin, Siu-Ming Yiu, and Junbin Fang. Communication-efficient secure neural network via key-reduced distributed comparison function. In *Provable and Practical Security: 18th International Conference, ProvSec 2024, Gold Coast, QLD, Australia, September 25–27, 2024, Proceedings, Part II*, page 143–163, Berlin, Heidelberg, 2024. Springer-Verlag.

A Additional Details for Strong Approximate Indistinguishability and DPF Security Amplification

A.1 Hardcore Lemma for Computational Indistinguishability

Maurer and Tessaro proved the following hardcore lemma for computational indistinguishability [MT10].

Lemma A.1 (Hardcore lemma: concrete version [MT10]). *Let $E : \mathcal{U} \rightarrow \mathcal{X}$ and $F : \mathcal{V} \rightarrow \mathcal{X}$ be functions, and let $\epsilon, \gamma \in (0, 1)$ and $s > 0$ be given. If for all distinguishers $\mathcal{A}$ with size s we have*

$$\left| \Pr_{U \leftarrow \mathcal{U}} [\mathcal{A}(E(U)) = 1] - \Pr_{V \leftarrow \mathcal{V}} [\mathcal{A}(F(V)) = 1] \right| \leq \epsilon,$$

then there exist measures $\mathcal{M}$ on $\mathcal{U}$ and $\mathcal{N}$ on $\mathcal{V}$ (which may depend on s, γ) with $\mu(\mathcal{M}) \geq 1 - \epsilon$ and $\mu(\mathcal{N}) \geq 1 - \epsilon$ such that

$$\left| \Pr_{U' \leftarrow P_{\mathcal{M}}} [\mathcal{A}'(E(U')) = 1] - \Pr_{V' \leftarrow P_{\mathcal{N}}} [\mathcal{A}'(F(V')) = 1] \right| \leq \gamma$$

for all distinguishers $\mathcal{A}'$ with size $s' := \frac{s\gamma^2}{128(\log |\mathcal{U}| + \log |\mathcal{V}| + 1)}$, where $P_{\mathcal{M}}$ denotes the distribution induced by the measure $\mathcal{M}$.

A popular but slightly flawed interpretation of Lemma A.1 is the following: “if two distributions are $(1 - \delta)$ -computationally indistinguishable, then there exists a hardcore measure of density $1 - \delta$ on which they are negligibly computationally indistinguishable”. While this colloquial interpretation captures the intuition of the hardcore lemma, it ignores a few subtleties. Firstly, the hardcore measures for the two distributions can be different. Secondly, the hardcore measure actually depends on the size and advantage of the adversary, so there may not be a fixed pair of hardcore measures for all adversary sizes or advantages. We need to give explicit adversary size and advantage before defining the hardcore measures. As a result, we make the PPT adversary’s size $P(\cdot)$ and advantage $1/Q(\cdot)$ explicit in the statement of the corollary below. In comparison with Lemma A.1 which is stated in terms of *concrete* adversary sizes and advantages, Corollary A.2 gives an *asymptotical* version where the adversary’s size and advantage are functions of a security parameter λ .

Corollary A.2 (Hardcore lemma: asymptotic version). *Let $E(\lambda) : \mathcal{U}(\lambda) \rightarrow \mathcal{X}(\lambda)$ and $F(\lambda) : \mathcal{V}(\lambda) \rightarrow \mathcal{X}(\lambda)$ be functions and let $\epsilon(\lambda) \in (0, 1)$ be given, where $\log |\mathcal{U}(\lambda)|, \log |\mathcal{V}(\lambda)| = \text{poly}(\lambda)$. If the distributions $\{E(\lambda)(U) \mid U \leftarrow \mathcal{U}(\lambda)\}$ and $\{F(\lambda)(V) \mid V \leftarrow \mathcal{V}(\lambda)\}$ are $(1 - \epsilon(\lambda))$ -computationally indistinguishable against all non-uniform PPT adversaries, then for any polynomials $P(\cdot)$ and $Q(\cdot)$, there exist families of measures*

$$\begin{aligned} &\{\mathcal{M}(\lambda, P(\lambda), Q(\lambda)) \text{ on } \mathcal{U}(\lambda) \text{ with density at least } 1 - \epsilon(\lambda)\}_{\lambda \in \mathbb{N}} \\ &\{\mathcal{N}(\lambda, P(\lambda), Q(\lambda)) \text{ on } \mathcal{V}(\lambda) \text{ with density at least } 1 - \epsilon(\lambda)\}_{\lambda \in \mathbb{N}} \end{aligned}$$

such that for all sufficiently large λ , for all non-uniform distinguishers $\mathcal{A}'$ of size $P(\lambda)$,

$$\begin{aligned} &\text{Adv}(\lambda, \mathcal{A}', P_{\mathcal{M}(\lambda, P(\lambda), Q(\lambda))}, P_{\mathcal{N}(\lambda, P(\lambda), Q(\lambda))}) \\ &:= \left| \Pr_{U' \leftarrow P_{\mathcal{M}(\lambda, P(\lambda), Q(\lambda))}} [\mathcal{A}'(E(\lambda)(U')) = 1] - \Pr_{V' \leftarrow P_{\mathcal{N}(\lambda, P(\lambda), Q(\lambda))}} [\mathcal{A}'(F(\lambda)(V')) = 1] \right| \leq 1/Q(\lambda). \end{aligned}$$

Proof. Fix the polynomials $P(\cdot)$ and $Q(\cdot)$. We apply the hardcore lemma (Lemma A.1) with $s'(\lambda) = P(\lambda)$, $\gamma(\lambda) = 1/Q(\lambda)$, and

$$s(\lambda) = 128s'(\lambda)(\log |\mathcal{U}(\lambda)| + \log |\mathcal{V}(\lambda)| + 1)/\gamma(\lambda)^2 = 128P(\lambda) \cdot \text{poly}(\lambda) \cdot Q(\lambda)^2 = \text{poly}(\lambda).$$

Consider the following non-uniform adversary $\mathcal{A}_0$. For every input length λ , $\mathcal{A}_0$ is the adversary of size $s(\lambda)$ with the maximum advantage in distinguishing $E(\lambda)$ and $F(\lambda)$ among all adversaries of

that size. Since $\mathcal{A}_0$ is a non-uniform PPT adversary, $(1 - \epsilon(\lambda))$ -computational indistinguishability implies that there exists a threshold λ_0 such that for all $\lambda \geq \lambda_0$,

$$\text{Adv}(\lambda, \mathcal{A}_0, E(\lambda), F(\lambda)) \leq \epsilon(\lambda).$$

Now fix any $\lambda \geq \lambda_0$. For all non-uniform distinguishers $\mathcal{A}$ of size $s(\lambda)$, we have

$$\text{Adv}(\lambda, \mathcal{A}, E(\lambda), F(\lambda)) \leq \text{Adv}(\lambda, \mathcal{A}_0, E(\lambda), F(\lambda)) \leq \epsilon(\lambda).$$

By the hardcore lemma, there exist $\mathcal{M}$ on $\mathcal{U}(\lambda)$ and $\mathcal{N}$ on $\mathcal{V}(\lambda)$ with density at least $1 - \epsilon(\lambda)$ such that

$$\left| \Pr_{U' \xleftarrow{\$} P_{\mathcal{M}}} [\mathcal{A}'(E(\lambda)(U')) = 1] - \Pr_{V' \xleftarrow{\$} P_{\mathcal{N}}} [\mathcal{A}'(F(\lambda)(V')) = 1] \right| \leq \gamma(\lambda)$$

for all non-uniform distinguishers $\mathcal{A}'$ with size $s'(\lambda) = P(\lambda)$. $\square$

A.2 Why We Still Lack a DPF Amplification Theorem

Boyle et al. [BGIK22b] claimed a security amplification theorem that upgrades a DPF scheme with $(1 - \delta)$ security to one that is fully secure. Moreover, their security amplification theorem has been instrumental in the construction of several DPF schemes [BGIK22b, GWW25]. However, upon closer examination, their security amplification theorem lacks a complete security proof as explained below.

Henceforth let $\text{Enc}(1^\lambda, s)$ denote the encoding of some secret s . Specifically, for a DPF scheme, $\text{Enc}(1^\lambda, \cdot)$ corresponds to the adversary's view whereas the secret s corresponds to the point α where the point function evaluates to non-zero — for simplicity, here we simply assume that $\beta = 1$ and α is the only secret. Boyle et al. [BGIK22b]'s security amplification proof crucially relies on the following informal intuition:

If there exists some ideal distribution $\text{Ideal}(1^\lambda)$ that is independent of the secret s , such that for any $s \in [N]$, $\text{Enc}(1^\lambda, s) \approx_\delta \text{Ideal}(1^\lambda)$, then it can be interpreted as: with probability δ , the secret s is completely leaked and with probability $1 - \delta$, s is protected.

Boyle et al. [BGIK22b] captures the above intuition more formally using the following statement:

Statement A.3 (Lemma 3 of [BGIK22b]). *If there exists some ideal distribution $\text{Ideal}(1^\lambda)$ that is independent of the secret s , such that for any $s \in [N]$, $\text{Enc}(1^\lambda, s) \approx_\delta \text{Ideal}(1^\lambda)$, then there exists a function*

$$\text{Leak}_\delta(s) = \begin{cases} s & \text{w.p. } \delta \\ \perp & \text{w.p. } 1 - \delta \end{cases} \quad (1)$$

and an efficiently computable function Sim such that for every $s \in [N]$,

$$\text{Enc}(1^\lambda, s) \approx \text{Sim}(1^\lambda, \text{Leak}_\delta(s))$$

There are two interpretations of the statement:

Stronger version: $\text{Sim}(1^\lambda, \cdot)$ and $\text{Leak}_\delta(\cdot)$ **do not share randomness**. This is the desired interpretation that matches the aforementioned intuition, and also the version implicitly used in the security amplification proof of Boyle et al. [BGIK22b]. However, this version is not true — Figure 2 depicts a counter-example where

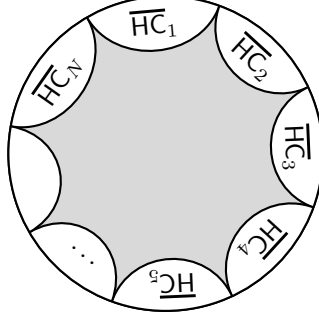


Figure 2: Counter-example: Each $\overline{HC}_s$ has probability mass δ . Suppose that $\text{Ideal}(1^\lambda)$ outputs s upon random coins in $\overline{HC}_s$ and outputs 0 everywhere else. For each $s \in [N]$, $\text{Enc}(1^\lambda, s)$ outputs the same as $\text{Ideal}(1^\lambda)$ except that for coins in $\overline{HC}_s$, it outputs 0. In other words, $\overline{HC}_s$ can be viewed as the complement of the hardcore measure for $\text{Enc}(1^\lambda, s)$. The hardcore measures for different secrets s may not be the same. The gray area represents the intersection of the hardcores.

- Each $\overline{HC}_s$ has probability mass δ .
- $\text{Ideal}(1^\lambda)$ outputs s upon coins in $\overline{HC}_s$ and outputs 0 everywhere else.
- For each $s \in [N]$, $\text{Enc}(1^\lambda, s)$ outputs the same as $\text{Ideal}(1^\lambda)$ except that for coins in $\overline{HC}_s$, it outputs 0. In other words, $\text{Enc}(1^\lambda, s)$ never outputs s .

In this counter-example, it is not hard to see that there does not exist a secret-independent common distribution $\text{Sim}(1^\lambda)$ such that for any $s \in [N]$, (any distribution that is computationally indistinguishable from) $\text{Enc}(1^\lambda, s)$ can be decomposed into sampling from $\text{Sim}(1^\lambda)$ with probability $1 - \delta$ and sampling from a residual, secret-dependent distribution with the remaining probability. In summary, more fundamentally, this stronger version is incorrect because *the hardcore measures for different secrets may not be the same*.

Note that for $(1 - \delta)$ -statistical indistinguishability, Statement A.3 can be fixed naïvely by relaxing the probability δ in Equation (1) to $N \cdot \delta$, i.e., $\text{Leak}_\delta(s)$ outputs $\perp$ on the intersection of all the hardcores (i.e., the gray area in Figure 2). It is also not hard to show that the N -factor loss is necessary for $(1 - \delta)$ -statistical indistinguishability. We stress also that this naïve fix by introducing an N -factor blowup does not seem to directly apply to $(1 - \delta)$ -computational indistinguishability: even if we take the intersection of the hardcores of $\{\text{Enc}(1^\lambda, s)\}_{s \in [N]}$, it is not clear how to find a corresponding “intersected hardcore” in the distribution $\text{Ideal}(1^\lambda)$ to make $\text{Enc}(1^\lambda, s)$ and $\text{Ideal}(1^\lambda)$ computationally indistinguishable on the respective “intersected hardcores”.

Weaker version: $\text{Sim}(1^\lambda, \cdot)$ and $\text{Leak}_\delta(\cdot)$ share randomness. This is the version explicitly stated in Lemma 3 of Boyle et al. [BGIK22b]. Unfortunately, their amplification proof actually implicitly uses the stronger interpretation, partly because the weaker version fails to capture the aforementioned intuition: for the shared randomness among $\text{Leak}_\delta(\cdot)$ and $\text{Sim}(1^\lambda)$ to be a meaningful weakening of the stronger version, the shared random coins must leak the underlying secret s to Sim even when $\text{Leak}_\delta(\cdot) = \perp$. Indeed, in their proof of Lemma 3 [BGIK22b], $\text{Leak}_\delta(s)$ determines whether to output $\perp$ based on whether its random coins land in the hardcore of s or not. Because the hardcores of different secrets may not be the same, learning the random coins of $\text{Leak}_\delta(\cdot)$ when its outcome is $\perp$ leaks information about the underlying secret.

A.3 Properties of Strong $(1 - \delta)$ -Computational Indistinguishability

In this section, we discuss basic properties of strong $(1 - \delta)$ -computational indistinguishability.

First, strong computational indistinguishability is *not a symmetric notion*, unlike computational indistinguishability. That is, $X \approx_\delta Y$ doesn't generally imply that $Y \approx_\delta X$. For example, for non-negligible $\delta < 1/2$, consider the constant distribution $Y = 0$, and $X = \begin{cases} 0 & \text{w.p. } 1 - \delta \\ 1 & \text{w.p. } \delta \end{cases}$. Clearly, $X \approx_\delta Y$, but the other direction does not hold.

Below, we prove some useful properties for strong $(1 - \delta)$ -computational indistinguishability including *transitivity*, *post-processing*, and *composition*.

Fact A.4 (Transitivity of strong $(1 - \delta)$ -indistinguishability). *Suppose that the ensembles $X(1^\lambda), Y(1^\lambda)$, and $W(1^\lambda)$ satisfy $X(1^\lambda) \approx_{\delta_1} Y(1^\lambda)$ and $Y(1^\lambda) \approx_{\delta_2} W(1^\lambda)$. Then $X(1^\lambda) \approx_{\delta_1 + \delta_2} W(1^\lambda)$.*

Proof. Fix polynomials $P(\cdot)$ and $Q(\cdot)$, for which we will show that $X(1^\lambda) \approx_{\delta_1 + \delta_2} W(1^\lambda)$. Consider λ sufficiently large for both $X(1^\lambda) \approx_{\delta_1} Y(1^\lambda)$ and $Y(1^\lambda) \approx_{\delta_2} W(1^\lambda)$.

Let $Z_1(1^\lambda) := \begin{cases} Y(1^\lambda) & \text{w.p. } 1 - \delta_1 \\ R_1(1^\lambda) & \text{w.p. } \delta_1 \end{cases}$ be the intermediate distribution for $X(1^\lambda)$ and $Y(1^\lambda)$ for

the polynomials $P(\lambda)$ and $2Q(\lambda)$. Similarly, let $Z_2(1^\lambda) := \begin{cases} W(1^\lambda) & \text{w.p. } 1 - \delta_2 \\ R_2(1^\lambda) & \text{w.p. } \delta_2 \end{cases}$ be the intermediate distribution for $Y(1^\lambda)$ and $W(1^\lambda)$ for the polynomials $P(\lambda)$ and $2Q(\lambda)$. We claim that

$$T(1^\lambda) := \begin{cases} Z_2(1^\lambda) & \text{w.p. } 1 - \delta_1 \\ R_1(1^\lambda) & \text{w.p. } \delta_1 \end{cases}$$

is the intermediate distribution for $X(1^\lambda) \approx_{\delta_1 + \delta_2} W(1^\lambda)$.

Firstly, for any non-uniform adversary $\mathcal{A}$ of size $P(\lambda)$,

$$\begin{aligned} \text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), T(1^\lambda)) &\leq \text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), Z_1(1^\lambda)) + \text{Adv}(\lambda, \mathcal{A}, Z_1(1^\lambda), T(1^\lambda)) \\ &\leq 1/(2Q(\lambda)) + (1 - \delta_1) \text{Adv}(\lambda, \mathcal{A}, Y(1^\lambda), Z_2(1^\lambda)) \\ &\quad \text{(both } Z_1 \text{ and } T \text{ output from } R_1 \text{ with prob. } \delta_1) \\ &\leq 1/(2Q(\lambda)) + (1 - \delta_1) \cdot 1/(2Q(\lambda)) \\ &\leq 1/Q(\lambda). \end{aligned}$$

Next, note that we can write $T(1^\lambda)$ as

$$T(1^\lambda) = \begin{cases} W(1^\lambda) & \text{w.p. } (1 - \delta_1)(1 - \delta_2) \\ R_2(1^\lambda) & \text{w.p. } (1 - \delta_1)\delta_2 \\ R_1(1^\lambda) & \text{w.p. } \delta_1 \end{cases}.$$

Then we can write $T(1^\lambda)$ as outputting from W with probability $1 - (\delta_1 + \delta_2)$ and from some residual distribution otherwise. Thus, X is strongly $(1 - (\delta_1 + \delta_2))$ -computationally indistinguishable from W . □

Fact A.5 (Post-processing lemma for strong $(1 - \delta)$ -indistinguishability). *Suppose that $X(1^\lambda) \approx_\delta Y(1^\lambda)$ and let $M(\cdot)$ be any PPT machine. Then, $M(1^\lambda, X(1^\lambda)) \approx_\delta M(1^\lambda, Y(1^\lambda))$.*

Proof. Fix polynomials $P(\lambda)$ and $Q(\lambda)$, for which we will show that $M(1^\lambda, X(1^\lambda)) \approx_\delta M(1^\lambda, Y(1^\lambda))$.

Let $Z(1^\lambda) = \begin{cases} Y(1^\lambda) & \text{w.p. } 1 - \delta \\ R(1^\lambda) & \text{w.p. } \delta \end{cases}$ be the intermediate distribution for $X(1^\lambda) \approx_\delta Y(1^\lambda)$ for the polynomials $P(\lambda) + |M| + O(1)$ and $Q(\lambda)$, where $|M|$ is the size of the description of M . We claim that $M(1^\lambda, Z(1^\lambda))$ is the intermediate distribution for $M(1^\lambda, X(1^\lambda)) \approx_\delta M(1^\lambda, Y(1^\lambda))$.

First, for all non-uniform adversaries $\mathcal{A}(1^\lambda, \cdot)$ of size $P(\lambda)$, let $\mathcal{A}'(1^\lambda, x) := \mathcal{A}(1^\lambda, M(1^\lambda, x))$, then we have $|\mathcal{A}'| \leq P(\lambda) + |M| + O(1)$. Hence

$$\text{Adv}(\lambda, \mathcal{A}, M(1^\lambda, X(1^\lambda)), M(1^\lambda, Z(1^\lambda))) = \text{Adv}(\lambda, \mathcal{A}', X(1^\lambda), Z(1^\lambda)) \leq 1/Q(\lambda).$$

Next, note that we can write $M(1^\lambda, Z(1^\lambda))$ as: $M(1^\lambda, Z(1^\lambda)) = \begin{cases} M(1^\lambda, Y(1^\lambda)) & \text{w.p. } 1 - \delta \\ M(1^\lambda, R(1^\lambda)) & \text{w.p. } \delta \end{cases}$,

which can be viewed as outputting from $M(1^\lambda, Y(1^\lambda))$ with probability $1 - \delta$ and from some residual distribution otherwise.

Therefore, $M(1^\lambda, X(1^\lambda))$ is strongly $(1 - \delta)$ -computationally indistinguishable from $M(1^\lambda, Y(1^\lambda))$. $\square$

Fact A.6 (Composition for strong $(1 - \delta)$ -indistinguishability). *Let $\ell := \ell(\lambda)$ be polynomially bounded in λ . Suppose that for each $i \in [\ell]$, $X_i(1^\lambda) \approx_{\delta_i} Y_i(1^\lambda)$. Moreover, suppose that all random variables*

$$X_1(1^\lambda), \dots, X_\ell(1^\lambda), Y_1(1^\lambda), \dots, Y_\ell(1^\lambda)$$

are mutually independent. Then, we have that

$$\{X_1(1^\lambda), \dots, X_\ell(1^\lambda)\} \approx_{\delta_1 + \dots + \delta_\ell} \{Y_1(1^\lambda), \dots, Y_\ell(1^\lambda)\}.$$

Proof. Consider the following hybrid distributions for all $i \in [\ell + 1]$:

$$\text{Hyb}_i := \{Y_1(1^\lambda), \dots, Y_{i-1}(1^\lambda), X_i(1^\lambda), X_{i+1}(1^\lambda), \dots, X_\ell(1^\lambda)\}$$

The only difference between Hyb_i and Hyb_{i+1} is that the i -th component is $X_i(1^\lambda)$ in Hyb_i and $Y_i(1^\lambda)$ in Hyb_{i+1} . Since all components in Hyb_i are independent, we have that $\text{Hyb}_i \approx_{\delta_i} \text{Hyb}_{i+1}$. By Fact A.4, we conclude that

$$\{X_1(1^\lambda), \dots, X_\ell(1^\lambda)\} \approx_{\delta_1 + \dots + \delta_\ell} \{Y_1(1^\lambda), \dots, Y_\ell(1^\lambda)\}$$

$\square$

Fact A.7 (A sufficient condition for strong $(1 - \delta)$ -indistinguishability). *Suppose that for distributions $X(1^\lambda)$ and $Y(1^\lambda)$ on the universe $\mathcal{U}$, for all values $U \in \mathcal{U}$,*

$$\Pr[X = U] \geq (1 - \delta(\lambda)) \Pr[Y = U].$$

Then, X is strongly $(1 - \delta)$ -statistically indistinguishable from Y .

Proof. X can be directly written as: with probability $(1 - \delta(\lambda))$, output from Y ; otherwise, output from some residual distribution. $\square$

Fact A.8. For any $\delta(\lambda) < 1$, suppose that the distribution ensembles $X(1^\lambda)$ and $Y(1^\lambda)$ satisfy $X(1^\lambda) \approx_{\delta(\lambda) + \text{negl}(\lambda)} Y(1^\lambda)$ for some negligible function $\text{negl}(\lambda)$. Then $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$.

Proof. Let $Z(1^\lambda) = \begin{cases} Y(1^\lambda) & \text{w.p. } 1 - \delta(\lambda) - \text{negl}(\lambda) \\ R(1^\lambda) & \text{w.p. } \delta(\lambda) + \text{negl}(\lambda) \end{cases}$ be the intermediate distribution for $X(1^\lambda) \approx_{\delta(\lambda) + \text{negl}(\lambda)} Y(1^\lambda)$. Then consider the following new intermediate distribution:

$$Z'(1^\lambda) := \begin{cases} Y(1^\lambda) & \text{w.p. } 1 - \delta(\lambda) \\ R(1^\lambda) & \text{w.p. } \delta(\lambda) \end{cases}.$$

Because Z and Z' have negligible statistical distance, $X(1^\lambda)$ is computationally indistinguishable from Z' , so Z' is a valid intermediate distribution for $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$. $\square$

Fact A.9 (Strong $(1 - \delta)$ -indistinguishability strictly stronger than $(1 - \delta)$ -indistinguishability). If $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$, then $X(1^\lambda) \approx_{\delta(\lambda) + \text{negl}(\lambda)} Y(1^\lambda)$. However, the converse is not true.

Proof. Suppose that $X(1^\lambda) \approx_{\delta(\lambda)} Y(1^\lambda)$. Fix any polynomials $P(\cdot)$ and $Q(\cdot)$. Let $Z(1^\lambda)$ be the intermediate distribution for $P(\cdot)$ and $Q(\cdot)$. Then for any non-uniform adversary $\mathcal{A}$ of size $P(\lambda)$,

$$\text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), Y(1^\lambda)) \leq \text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), Z(1^\lambda)) + \text{SD}(Z(1^\lambda), Y(1^\lambda)) \leq 1/Q(\lambda) + \delta(\lambda)$$

where $\text{SD}(Z, Y)$ denotes the statistical distance between the two random variables.

Because this holds for any polynomial $Q(\lambda)$, for any non-uniform adversary $\mathcal{A}$ of size $P(\lambda)$, there exists some negligible function $\text{negl}(\lambda)$ such that

$$\text{Adv}(\lambda, \mathcal{A}, X(1^\lambda), Y(1^\lambda)) \leq \delta(\lambda) + \text{negl}(\lambda).$$

Hence we obtain $(1 - \delta(\lambda) - \text{negl}(\lambda))$ -computational indistinguishability for $X(1^\lambda)$ and $Y(1^\lambda)$.

To show that the converse does not hold, consider any non-negligible $\delta(\lambda) < 1/2$ and the following distributions with statistical distance $\delta(\lambda)$:

$$X = \begin{cases} 0 & \text{w.p. } 1 - \delta(\lambda) \\ 1 & \text{w.p. } \delta(\lambda) \end{cases}, \quad Y = 0.$$

Clearly, Y is not strongly $(1 - \delta(\lambda))$ -computationally indistinguishable from X . $\square$

A.4 Proof of the Decoy Lemma

Now we give a proof of Lemma 3.1 that correctly deals with the subtleties in the order of quantifiers that we alluded to. Recall that in the hardcore lemma, the hardcore measure depends on the size and advantage of the adversary, so we need to explicitly define the polynomials $P_1(\cdot)$ and $Q_1(\cdot)$ before we can use the hardcore lemma to get hardcore measures. The proof is otherwise the same as that of Lemma E.4 of Ghoshal et al. [GZSP25].

Proof. Fix arbitrary polynomials $P_1(\cdot)$ and $Q_1(\cdot)$. We show strong indistinguishability with respect to these polynomials.

Since $\{D(1^\lambda, r) : r \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}\}$ and $\{D'(1^\lambda, r) : r \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}\}$ are $(1 - \delta(\lambda))$ -computationally indistinguishable, Corollary A.2 implies the following. For the polynomials $P_1(\lambda)$ and $Q_2(\lambda) := 2Q_1(\lambda)$, there exist families of measures $\mathcal{M}(\lambda, P_1(\lambda), Q_2(\lambda))$ and $\mathcal{N}(\lambda, P_1(\lambda), Q_2(\lambda))$ on $\{0, 1\}^{\ell(\lambda)}$

with density at least $1 - \delta(\lambda)$, and a threshold λ_0 , such that for all $\lambda \geq \lambda_0$, for all non-uniform distinguishers $\mathcal{A}$ of size $P_1(\lambda)$,

$$\text{Adv}(\lambda, \mathcal{A}, P_{\mathcal{M}(\lambda, P_1(\lambda), Q_2(\lambda))}, P_{\mathcal{N}(\lambda, P_1(\lambda), Q_2(\lambda))}) \leq 1/Q_2(\lambda).$$

From now on, fix any $\lambda \geq \lambda_0$. Abbreviate

$$\mathcal{M} := \mathcal{M}(\lambda, P_1(\lambda), Q_2(\lambda)) \quad \text{and} \quad \mathcal{N} := \mathcal{N}(\lambda, P_1(\lambda), Q_2(\lambda)),$$

and let $p := 1 - \delta(\lambda)$. Without loss of generality, assume that $\mathcal{M}$ and $\mathcal{N}$ have density exactly p . Indeed, if $\mu(\mathcal{M}) > p$, we may replace $\mathcal{M}$ by $\mathcal{M}'(x) = \frac{p}{\mu(\mathcal{M})}\mathcal{M}(x)$, which induces the same distribution as $\mathcal{M}$; similarly for $\mathcal{N}$.

Experiment Hyb'. Sample a random position i^* for the special instance D' . For each $i \in [d]$, sample a bit β_i indicating whether the i -th copy should be sampled from the hardcore or not.

1. Sample $i^* \xleftarrow{\$} [d]$ uniformly at random.
2. For $i \in [d]$, sample $\beta_i \xleftarrow{\$} \text{Ber}(p)$.
3. For $i \in [d] \setminus \{i^*\}$, if $\beta_i = 1$, sample $r_i \xleftarrow{\$} P_{\mathcal{M}}$; else sample $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$.
4. For i^* , if $\beta_{i^*} = 1$, sample $r_{i^*} \xleftarrow{\$} P_{\mathcal{N}}$; else sample $r_{i^*} \xleftarrow{\$} P_{1-\mathcal{N}}$.
5. Output $\text{Set}(D(1^\lambda, r_1), \dots, D(1^\lambda, r_{i^*-1}), D'(1^\lambda, r_{i^*}), D(1^\lambda, r_{i^*+1}), \dots, D(1^\lambda, r_d))$.

This is identical to X' .

Experiment Hyb₁. If $\beta_{i^*} = 1$, replace the hardcore sample from D' with a hardcore sample from D .

1. Sample $i^* \xleftarrow{\$} [d]$ uniformly at random.
2. For $i \in [d]$, sample $\beta_i \xleftarrow{\$} \text{Ber}(p)$.
3. For $i \in [d] \setminus \{i^*\}$, if $\beta_i = 1$, sample $r_i \xleftarrow{\$} P_{\mathcal{M}}$; else sample $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$.
4. For i^* , if $\beta_{i^*} = 1$, sample $r_{i^*} \xleftarrow{\$} P_{\mathcal{M}}$, and output

$$\text{Set}(D(1^\lambda, r_1), \dots, D(1^\lambda, r_{i^*-1}), D(1^\lambda, r_{i^*}), D(1^\lambda, r_{i^*+1}), \dots, D(1^\lambda, r_d)).$$

Otherwise, sample $r_{i^*} \xleftarrow{\$} P_{1-\mathcal{N}}$, and output

$$\text{Set}(D(1^\lambda, r_1), \dots, D(1^\lambda, r_{i^*-1}), D'(1^\lambda, r_{i^*}), D(1^\lambda, r_{i^*+1}), \dots, D(1^\lambda, r_d)).$$

Suppose for contradiction that there exists an adversary $\mathcal{A}'$ of size $P_1(\lambda)$ that distinguishes Hyb' and Hyb₁ with advantage greater than $1/Q_2(\lambda)$, i.e.,

$$|\Pr[\mathcal{A}'(\text{Hyb}') = 1] - \Pr[\mathcal{A}'(\text{Hyb}_1) = 1]| > 1/Q_2(\lambda).$$

Note that

$$\begin{aligned} & |\Pr[\mathcal{A}'(\text{Hyb}') = 1] - \Pr[\mathcal{A}'(\text{Hyb}_1) = 1]| \\ &= |p \cdot \Pr[\mathcal{A}'(\text{Hyb}') = 1 \mid \beta_{i^*} = 1] + (1-p) \cdot \Pr[\mathcal{A}'(\text{Hyb}') = 1 \mid \beta_{i^*} = 0] \\ &\quad - p \cdot \Pr[\mathcal{A}'(\text{Hyb}_1) = 1 \mid \beta_{i^*} = 1] - (1-p) \cdot \Pr[\mathcal{A}'(\text{Hyb}_1) = 1 \mid \beta_{i^*} = 0]| \\ &= |p \cdot \Pr[\mathcal{A}'(\text{Hyb}') = 1 \mid \beta_{i^*} = 1] - p \cdot \Pr[\mathcal{A}'(\text{Hyb}_1) = 1 \mid \beta_{i^*} = 1]| \end{aligned}$$

because when $\beta_{i^*} = 0$, the distribution of Hyb' is identical to that of Hyb_1 . Thus,

$$|\Pr[\mathcal{A}'(\text{Hyb}') = 1 \mid \beta_{i^*} = 1] - \Pr[\mathcal{A}'(\text{Hyb}_1) = 1 \mid \beta_{i^*} = 1]| > 1/Q_2(\lambda) \cdot 1/p \geq 1/Q_2(\lambda).$$

Consider the following adversary $\mathcal{A}$ for $\{D(r) \mid r \xleftarrow{\$} P_{\mathcal{M}}\}$ and $\{D'(r) \mid r \xleftarrow{\$} P_{\mathcal{N}}\}$. On input x , it outputs $\mathcal{A}'(\text{Set}(x, D(1^\lambda, r_1), \dots, D(1^\lambda, r_{d-1})))$. When $x \xleftarrow{\$} \{D(r) \mid r \xleftarrow{\$} P_{\mathcal{M}}\}$, the input to $\mathcal{A}'$ is distributed as $\{\text{Hyb}_1 \mid \beta_{i^*} = 1\}$. Similarly, when $x \xleftarrow{\$} \{D'(r) \mid r \xleftarrow{\$} P_{\mathcal{N}}\}$, the input to $\mathcal{A}'$ is distributed as $\{\text{Hyb}' \mid \beta_{i^*} = 1\}$. Hence, $\mathcal{A}$ has advantage greater than $1/Q_2(\lambda)$. Therefore, there exist fixed values for $D(1^\lambda, r_i)$, $i \in [d-1]$, such that hardcoding them into $\mathcal{A}$ preserves advantage greater than $1/Q_2(\lambda)$. This hardcoded adversary has size at most $P_1(\lambda)$, contradicting the fact that for all non-uniform distinguishers $\mathcal{A}$ of size $P_1(\lambda)$, the advantage is bounded by $1/Q_2(\lambda)$.

Thus, for all adversaries $\mathcal{A}'$ of size $P_1(\lambda)$,

$$|\Pr[\mathcal{A}'(\text{Hyb}') = 1] - \Pr[\mathcal{A}'(\text{Hyb}_1) = 1]| \leq 1/Q_2(\lambda).$$

Experiment Hyb_2 .

1. Sample $h \xleftarrow{\$} \text{Bin}(d, p)$ and $\beta \xleftarrow{\$} \text{Ber}(h/d)$.
2. Sample $h - \beta$ copies of D with $r_i \xleftarrow{\$} P_{\mathcal{M}}$, and sample $(d - 1 - (h - \beta))$ copies of D with $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$.
3. If $\beta = 1$, sample one copy of D with $r \xleftarrow{\$} P_{\mathcal{M}}$; else sample one copy of D' with $r \xleftarrow{\$} P_{1-\mathcal{N}}$.
4. Output the unordered set of sampled values.

This is identical to Hyb_1 .

Experiment Hyb'_2 . Let $\mu = pd$.

1. Sample $h \xleftarrow{\$} \text{Bin}(d, p)$. If $h < \mu - \sqrt{\mu} \log \lambda$, abort and output $\perp$.
2. Sample $\beta \xleftarrow{\$} \text{Ber}(h/d)$.
3. Sample $h - \beta$ copies of D with $r_i \xleftarrow{\$} P_{\mathcal{M}}$, and sample $(d - 1 - (h - \beta))$ copies of D with $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$.
4. If $\beta = 1$, sample one copy of D with $r \xleftarrow{\$} P_{\mathcal{M}}$; else sample one copy of D' with $r \xleftarrow{\$} P_{1-\mathcal{N}}$.
5. Output the unordered set of sampled values.

By a Chernoff bound,

$$\Pr[h < \mu - \sqrt{\mu} \log \lambda] \leq 1/\lambda^{\log(\lambda)/2}.$$

Thus, Hyb_2 and Hyb'_2 have negligible statistical distance. Hence, there exists a constant λ_1 such that for all $\lambda > \lambda_1$, $1/\lambda^{\log(\lambda)/2} < 1/(2Q_2(\lambda))$.

Experiment Hyb_3 . Let $\mu = pd$.

1. Sample $h \xleftarrow{\$} \text{Bin}(d, p)$. If $h < \mu - \sqrt{\mu} \log \lambda$, abort and output $\perp$.
2. Sample $\beta \xleftarrow{\$} \text{Ber}(p - \frac{\log \lambda}{\sqrt{d}})$.

3. If $\beta = 1$, sample h copies of D with $r_i \xleftarrow{\$} P_{\mathcal{M}}$, and sample $(d - h)$ copies of D with $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$. Output the sampled values.
4. If $\beta = 0$, output from some suitable residual distribution.

Note that in Hyb'_2 , conditioned on not aborting,

$$\Pr[\beta = 1] \geq \frac{\mu - \sqrt{\mu} \log \lambda}{d} = p - \sqrt{\frac{p}{d}} \log \lambda \geq p - \frac{\log \lambda}{\sqrt{d}} = 1 - 2\delta(\lambda).$$

Thus, Hyb_3 is identically distributed as Hyb'_2 for some appropriate residual distribution.

Experiment Hyb_4 . Let $\mu = pd$.

1. Sample $\beta \xleftarrow{\$} \text{Ber}(1 - 2\delta(\lambda))$.
2. Sample $h \xleftarrow{\$} \text{Bin}(d, p)$.
3. If $\beta = 1$, sample h copies of D with $r_i \xleftarrow{\$} P_{\mathcal{M}}$, and sample $(d - h)$ copies of D with $r_i \xleftarrow{\$} P_{1-\mathcal{M}}$. Output the sampled values.
4. If $\beta = 0$, output from some suitable residual distribution.

By the same Chernoff bound as before, Hyb_4 has negligible statistical distance to Hyb_3 .

Experiment Hyb_5 . Let $\mu = pd$.

1. Sample $\beta \xleftarrow{\$} \text{Ber}(1 - 2\delta(\lambda))$.
2. If $\beta = 1$, output d independent samples from D , i.e., output from $X(1^\lambda)$.
3. If $\beta = 0$, output from some suitable residual distribution.

This is identical to the previous hybrid.

So far, we have shown that for all polynomials $P_1(\lambda)$ and $Q_1(\lambda)$, we can define an ensemble $Z(1^\lambda) := \text{Hyb}_5$ such that for all $\lambda > \max\{\lambda_0, \lambda_1\}$ and all adversaries $\mathcal{A}'$ of size $P_1(\lambda)$,

$$|\Pr[\mathcal{A}'(\text{Hyb}') = 1] - \Pr[\mathcal{A}'(Z(1^\lambda)) = 1]| \leq 1/Q_2(\lambda) + 2 \cdot 1/(2Q_2(\lambda)) = 1/Q_1(\lambda).$$

Thus, $X'(1^\lambda)$ is strongly $(1 - 2\delta(\lambda))$ -computationally indistinguishable from $X(1^\lambda)$. □

A.5 From Regular to Strong $(1 - \delta)$ -Security for DPFs

We use the decoy lemma together with the amplification technique of [BGIK22b] to upgrade weakly secure DPFs into negligibly secure ones, incurring only a polylogarithmic overhead.

To do this, we use the decoy lemma to upgrade *regular* $(1 - \delta)$ -secure multiparty DPFs to *strongly* $(1 - \Theta(\delta))$ -secure ones. The transformation increases the share size by a polylogarithmic multiplicative factor. Intuitively, the following construction creates decoys by sharing the all-zero function a number of times. Additive reconstruction remains correct because adding these zero shares does not change the desired point function.

Upgrade construction. Let $d := (\log \lambda)^2 / \delta^2$. Let $(\text{Gen}, \text{Eval})$ denote the underlying m -party DPF with regular $(1 - \delta)$ -security. We construct the following m -party DPF scheme $(\text{Gen}', \text{Eval}')$ with the same domain and range as the original DPF, but now with *strong* $(1 - 2\delta)$ -security.

$\text{Gen}'(1^\lambda, \alpha, \beta)$

1. Sample a uniformly random index $j^* \xleftarrow{\$} [d]$ for the real share.
2. Run $\text{Gen}(1^\lambda, \alpha, \beta)$ to obtain the real shares $(k_1^{(j^*)}, \dots, k_m^{(j^*)})$.
3. Run $\text{Gen}(1^\lambda, 1, 0)$ independently $d - 1$ times to obtain zero shares $(k_1^{(j)}, \dots, k_m^{(j)})$ for all $j \in [d] \setminus \{j^*\}$.
4. For all $i \in [m]$, output the i -th share $(k_i^{(1)}, \dots, k_i^{(d)})$.

$\text{Eval}'((k_i^{(1)}, \dots, k_i^{(d)}), x)$

1. Run $\text{Eval}(k_i^{(j)}, x)$ for all of $j \in [d]$.
2. Output $\sum_{j \in [d]} \text{Eval}(k_i^{(j)}, x)$.

Lemma A.10. *In the above construction, suppose the underlying DPF denoted $(\text{Gen}, \text{Eval})$ satisfies the following:*

- $(1 - \gamma)$ -correctness;
- $(1 - \delta)$ -security in the presence of up to t corruptions where $\delta \geq 1/\text{poly}(\lambda)$; and
- S per-party share size.

Then, the upgraded DPF denoted $(\text{Gen}', \text{Eval}')$ satisfies

- $(1 - \frac{\log^2 \lambda}{\delta^2} \cdot \gamma)$ -correctness;
- Strong $(1 - 2\delta)$ -security in the presence of up to t corruptions; and
- $\frac{\log^2 \lambda}{\delta^2} \cdot S$ per-party share size.

Proof. We now prove Lemma A.10.

$(1 - d\gamma)$ -correctness. Henceforth let Dom and Ran denote the domain and range of both DPFs. Fix any $x, \alpha \in \text{Dom}$ and $\beta \in \text{Ran}$. By the $(1 - \gamma)$ -correctness of the original scheme, the probability that

$$\sum_{i \in [m]} \text{Eval}(k_i^{(j^*)}, x) = f^{\alpha, \beta}(x)$$

is at least $1 - \gamma$. Similarly, for all $j \in [d] \setminus \{j^*\}$, the probability that

$$\sum_{i \in [m]} \text{Eval}(k_i^{(j)}, x) = f^{0, 0}(x)$$

is also at least $1 - \gamma$. By a union bound, the probability that all of these conditions hold is at least $1 - d\gamma$. Assuming all of these conditions hold, adding the evaluations gives us

$$\begin{aligned} \sum_{i \in [m]} \text{Eval}'((k_i^{(1)}, \dots, k_i^{(d)}), x) &= \sum_{j \in [d]} \sum_{i \in [m]} \text{Eval}(k_i^{(j)}, x) \\ &= (d - 1)f^{0, 0}(x) + f^{\alpha, \beta}(x) \\ &= f^{\alpha, \beta}(x). \end{aligned}$$

Strong $(1 - 2\delta)$ -security.

Fix any subset of t corrupted parties $T \subseteq [m]$ and any $\alpha \in \text{Dom}, \beta \in \text{Ran}$. Define the adversary's view of the underlying DPFs as follows:

$$\begin{aligned} D'(1^\lambda, r) &:= \{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha, \beta)\}, \\ D(1^\lambda, r) &:= \{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, 1, 0)\}, \end{aligned}$$

where $r \xleftarrow{\$} \{0, 1\}^{\ell(\lambda)}$ is the randomness used by Gen . By the $(1 - \delta)$ -security of the original scheme, for PPT adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in distinguishing $D'(1^\lambda, r)$ and $D(1^\lambda, r)$ is less than δ .

Because the adversary's view of the overall scheme is the unordered set of one sample of $D'(1^\lambda, r)$ and $(d - 1)$ samples of $D(1^\lambda, r)$, by the decoy lemma (Lemma 3.1), it is strongly $(1 - 2\delta)$ -computationally indistinguishable from the view of the following simulator:

$\text{Sim}(1^\lambda)$:

1. For $j \in [d]$, sample $(k_1^{(j)}, \dots, k_m^{(j)}) \xleftarrow{\$} \text{Gen}(1^\lambda, 1, 0)$ independently.
2. For all $i \in [m]$, output the i -th share $(k_i^{(1)}, \dots, k_i^{(d)})$.

□

A.6 Security Amplification for Strongly $(1 - \delta)$ -Secure DPFs

Given a strongly $(1 - \delta)$ -secure DPF, we now upgrade it to a fully secure one. The upgrade construction, based on Reed-Muller (RM) codes, is similar to that of Boyle et al. [BGIK22b]. Boyle et al. [BGIK22b] claims that RM-based upgrade works even if the underlying DPF satisfies only regular $(1 - \delta)$ -security; however, their proof is incomplete. We instead prove the RM-based upgrade works when the underlying DPF satisfies *strong* $(1 - \delta)$ -security, and separately we provide a technique for upgrading a DPF from regular to strong $(1 - \delta)$ -security (Appendix A.5).

Intuition. Without loss of generality, we may assume that $\beta = 1$ — we can always extend the DPF's domain size from N to $N + 1$; and if $\beta = 1$, simply let $\alpha = N + 1$, i.e., a point that will never be queried is set to 1.

Boyle et al. [BGIK22b] proposed the following idea: we split the secret α into an appropriate number of RM-code queries $\alpha_1, \dots, \alpha_q$, and a separate DPF instance is applied to each query $j \in [q]$:

$$(k_1^j, k_2^j, \dots, k_m^j) \leftarrow \text{DPF}_j.\text{Gen}(1^\lambda, \alpha_j, 1).$$

Moreover, the underlying DPF_j instances work for a larger domain denoted $\mathbb{F}'$ of size $N' = \Theta(N^{1+\epsilon})$. The smoothness of the RM code guarantees that as long as sufficiently many DPF instances hide their underlying α_j 's, then an adversary observing the α_j values of the broken instances learn nothing about α . During **Eval**, the input x is written as a one-hot vector of length N where only the x -th location is non-zero. This one-hot vector is then RM-encoded into a length $N' := N^{1+\epsilon}$ vector denoted $\vec{P}_x$. Then, for each DPF instance $j \in [q]$, party $i \in [m]$ computes the DPF's evaluation at all points:

$$\text{DPF}_j.\text{EvalAll}(k_i) := \{\text{DPF}_j.\text{Eval}(k_i, y)\}_{y \in \mathbb{F}'}$$

and then computes the following inner-product:

$$s_{i,j} := \vec{P}_x \cdot \text{DPF}_j.\text{EvalAll}(k_i)$$

Since the RM code's reconstruction (which uses Lagrange interpolation) is a linear function with publicly known coefficients, and the DPF's reconstruction is additive, it is not hard to see that the final outcome can be obtained by having each party i first take a linear combination of $s_{i,1}, \dots, s_{i,q}$, and then summing all the results.

Theorem A.11 (Security amplification for strongly $(1 - \delta)$ -secure DPF). *For a real number $0 < \epsilon < 1$, a domain size $N \in \mathbb{N}$, and security parameter λ , define the parameters $N' := N^{1+\epsilon}$ and $K := (4 \log(N, \lambda))^{1+3/\epsilon}$. Suppose the parameters satisfy $\frac{\epsilon \log N}{\log \log N + 2 \log \log \lambda + \log \epsilon + 2} \geq 1$.*

Suppose that there exists an m -party DPF tolerating up to t corrupt parties for domain $[N']$ and range $\mathbb{Z}_2$ with the following properties:

- $(1 - \gamma)$ -correctness;
- Strong $(1 - \frac{1}{K})$ -security; and
- $S(m, t, N')$ per-party share size.

Then, we can construct an m -party DPF tolerating up to t corrupt parties for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:

- $(1 - (4 \log(N, \lambda))^{3+3/\epsilon} \cdot \gamma)$ -correctness;
- $(1 - \text{negl}(\lambda))$ -security; and
- $(4 \log(N, \lambda))^{3+3/\epsilon} \cdot S(m, t, N')$ per-party share size.

Proof. Denote the underlying m -party DPF as $(\text{Gen}, \text{Eval})$, and construct the following DPF as $(\text{Gen}', \text{Eval}')$. In this proof, $\log$ denotes $\log_2$.

We set the following parameters:

- The number of variables of the RM code $r := \frac{\epsilon \log N}{\log \log N + 2 \log \log \lambda + \log \epsilon + 2}$.
- The degree of the RM code $d := rN^{1/r}$.
- The independence parameter for the RM-code queries $\sigma := (\log \lambda)^2$.
- The number of queries for the RM code $q := \sigma d + 1$.
- Let $\mathbb{F}$ be the smallest finite field of characteristic 2 such that $|\mathbb{F}| \geq q + 1$.
- Let $N' := |\mathbb{F}^r| \leq (2q)^r$; we will show that this is at most $N^{1+\epsilon}$.
- Let $e_1, \dots, e_q$ be a fixed set of q distinct *nonzero* points in $\mathbb{F}$.
- Let $E : [N'] \rightarrow \mathbb{F}^r$ be a fixed bijection.
- Fix any additive homomorphism $\phi : \mathbb{F} \rightarrow \mathbb{F}_2$ such that $\phi(1) = 1$, such as the field trace.

$\text{Gen}'(1^\lambda, \alpha, \beta)$

1. If $\beta = 0$, set $\alpha = N + 1$ and $\beta = 1$.
2. Sample a random degree- σ parametric curve $\gamma : \mathbb{F} \rightarrow \mathbb{F}^r$ such that $\gamma(0) = E(\alpha)$; that is, each coordinate of γ is an independent random univariate degree- σ polynomial subject to passing through the corresponding coordinate of $E(\alpha)$ at 0.
3. Compute the RM-code queries $(\alpha_j)_{j \in [q]} \in [N']^q$, where $\alpha_j = E^{-1}(\gamma(e_j))$; that is, α_j is the index in $[N']$ satisfying $E(\alpha_j) = \gamma(e_j)$.
4. For all $j \in [q]$, compute $(k_1^{(j)}, \dots, k_m^{(j)}) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_j, 1)$.
5. Output the i -th share as $(k_i^{(j)})_{j \in [q]}$, i.e., the concatenation of the i -th share of every underlying DPF instance.

$\text{Eval}'((k^{(j)})_{j \in [q]}, x)$

1. Interpolate an r -variable, d -degree polynomial $P \in \mathbb{F}[X_1, \dots, X_r]$ such that

$$P(E(z)) = \begin{cases} 1 & z = x \\ 0 & z \in [N+1], z \neq x \end{cases}.$$

2. For $j \in [q]$, define the Lagrange polynomials $L_j(X) := \prod_{j' \in [q], j' \neq j} \frac{X - e_{j'}}{e_j - e_{j'}}$.
3. Output

$$\phi \left(\sum_{j \in [q]} \sum_{y \in [N']} L_j(0) \cdot P(E(y)) \cdot \text{Eval}(k^{(j)}, y) \right).$$

Efficiency. Note that the number of monomials in a total degree- d , r -variate polynomial is

$$\binom{d+r}{r} > \left(\frac{d}{r} + 1 \right)^r \geq N+1.$$

Thus, the interpolation step in Eval' is possible whenever $E(1), \dots, E(N+1)$ are chosen so that the vector $(P(E(1)), \dots, P(E(N+1)))$ spans all of $\mathbb{F}^{N+1}$ over the choice of the total degree- d , r -variate polynomial P .

By definition, we have the following parameters:

$$\begin{aligned} K &= (4 \log(N, \lambda))^{1+3/\epsilon} \\ r &= \frac{\epsilon \log N}{\log \log N + 2 \log \log \lambda + \log \epsilon + 2} \leq \epsilon \log N \\ d &= rN^{1/r} = r \cdot 2^{(\log N)/r} \\ &= r \cdot (\log N (\log \lambda)^2 \epsilon \cdot 4)^{1/\epsilon} \\ &\leq 4^{1/\epsilon} \cdot (\log(N, \lambda))^{1+3/\epsilon} \\ &\leq K/4 \\ q &= \sigma d + 1 \leq K(\log \lambda)^2/2 \leq (4 \log(N, \lambda))^{3+3/\epsilon} \end{aligned}$$

$$\begin{aligned} N' &= |\mathbb{F}|^r \leq (2q)^r \leq (3\sigma d)^r \\ &\leq \left(3(\log \lambda)^2 r N^{1/r} \right)^r \\ &\leq N \cdot 2^{r(\log r + 2 \log \log \lambda + 2)} \\ &\leq N \cdot 2^{r(\log \epsilon + \log \log N + 2 \log \log \lambda + 2)} \\ &= N \cdot 2^{\epsilon \log N} \\ &= N^{1+\epsilon}. \end{aligned}$$

Because there are q instances of the underlying DPF scheme, we obtain efficiency as desired.

$(1 - q\gamma)$ -correctness. By a union bound, with probability at least $1 - q\gamma$, all instances of the original DPF scheme are correct; that is, for all $j \in [q]$ and $x \in [N]$,

$$\sum_{i \in [m]} \text{Eval}(k_i^{(j)}, x) = f^{\alpha_j, \beta_j}(x).$$

Condition on this event. Since $P(\gamma(X))$ has degree at most $d\sigma < q$, Lagrange interpolation gives

$$P(\gamma(X)) = \sum_{j \in [q]} L_j(X) P(\gamma(e_j)).$$

Then, we have

$$\begin{aligned} & \sum_{i \in [m]} \text{Eval}'((k_i^{(j)})_{j \in [q]}, x) \\ &= \phi \left(\sum_{j \in [q]} \sum_{y \in [N']} L_j(0) \cdot P(E(y)) \sum_{i \in [m]} \text{Eval}(k_i^{(j)}, y) \right) \\ &= \phi \left(\sum_{j \in [q]} \sum_{y \in [N']} L_j(0) \cdot P(E(y)) \cdot f^{\alpha_j, 1}(y) \right) \\ &= \phi \left(\sum_{j \in [q]} L_j(0) \cdot P(\gamma(e_j)) \right) \quad (\text{because } E(\alpha_j) = \gamma(e_j)) \\ &= \phi(P(\gamma(0))) \\ &= \phi(P(E(\alpha))). \end{aligned}$$

Because $P(E(\alpha)) = \begin{cases} 1 & x = \alpha \\ 0 & x \neq \alpha, \alpha \in [N+1] \end{cases}$, we obtain correctness.

$(1 - \text{negl}(\lambda))$ -security.

Fix any subset of t corrupted parties $T \subseteq [m]$. To prove $(1 - \text{negl}(\lambda))$ -security for the final scheme, it suffices to show that for any $\alpha_0, \alpha_1 \in \text{Dom}$ and $\beta_0, \beta_1 \in \text{Ran}$, the adversary's views for $b \in \{0, 1\}$,

$$\{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}'(1^\lambda, \alpha_b, \beta_b)\},$$

are computationally indistinguishable with negligible advantage. It suffices to show that the view is computationally indistinguishable from a distribution independent of α_b and β_b . From here on, we omit the subscript b and write α and β .

By the strong $(1 - 1/K)$ -security of the original scheme, there exists a simulator $\text{Sim}(1^\lambda)$ such that for all $j \in [q]$, the following distributions are strongly $(1 - 1/K)$ -computationally indistinguishable:

$$\begin{aligned} X'(1^\lambda, j) &:= \{(k_i)_{i \in T} \mid (k_1, \dots, k_m) \xleftarrow{\$} \text{Gen}(1^\lambda, \alpha_j, 1)\} \\ \approx_{1/K} X(1^\lambda) &:= \{(k_i)_{i \in T} \xleftarrow{\$} \text{Sim}(1^\lambda)\}. \end{aligned}$$

Fix any polynomial $P_1(\cdot)$ and $Q_1(\cdot)$. By the definition of strong $(1 - 1/K)$ -computational indistinguishability, for the polynomials $P_1(\lambda)$ and $Q_2(\lambda) := q \cdot Q_1(\lambda)$, there exists an ensemble $Z(1^\lambda, j)$ and a threshold λ_j such that for all $\lambda > \lambda_j$, for all non-uniform adversaries $\mathcal{A}'$ of size $P_1(\lambda)$,

$$\text{Adv}(\lambda, \mathcal{A}', X'(1^\lambda, j), Z(1^\lambda, j)) \leq 1/Q_2(\lambda),$$

where $Z(1^\lambda, j)$ can be written as follows: with probability $1 - 1/K$, output a random sample from $X(1^\lambda)$; else, output a random sample from some residual distribution.

By a sequence of q hybrids, for all $\lambda > \max\{\lambda_j\}_{j \in [q]}$ and all adversaries $\mathcal{A}$ of size $P_1(\lambda)$, we have

$$\text{Adv}(\lambda, \mathcal{A}, (X'(1^\lambda, j))_{j \in [q]}, (Z(1^\lambda, j))_{j \in [q]}) \leq 1/Q_1(\lambda).$$

Let **Leak** denote the subset of indices $j \in [q]$ for which $Z(1^\lambda, j)$ does not output a sample from $X(1^\lambda)$. Independently for each $j \in [q]$, we have $\Pr[(j) \in \text{Leak}] = 1/K$. Whenever $Z(1^\lambda, j)$ outputs from $X(1^\lambda)$, the output is independent of α and β ; hence it remains only to analyze the leaked instances.

By a Chernoff bound, because $\mathbb{E}[|\text{Leak}|] = q \cdot 1/K \leq (\log \lambda)^2/2$ and $\sigma = (\log \lambda)^2$,

$$\Pr[|\text{Leak}| > \sigma] = \exp(-\Omega(\log^2 \lambda)) = \text{negl}(\lambda).$$

Now condition on the event that $|\text{Leak}| \leq \sigma$. The queries $\{\alpha_j\}_{j \in [q]}$ of the RM code are evaluations of a random degree- σ polynomial γ with one fixed value at 0, so any σ of these query points are uniformly random and independent. Thus, all α_j values appearing in **Leak** are uniformly random and independent. Therefore, conditioned on the event that $|\text{Leak}| \leq \sigma$, the entire distribution of $(Z(1^\lambda, j))_{j \in [q]}$ is independent of α and β .

Thus, for any adversary $\mathcal{A}$ of size $P_1(\lambda)$, its advantage in distinguishing $(X'(1^\lambda, j))_{j \in [q]}$ from a distribution independent of α and β is at most $1/Q_1(\lambda) + \text{negl}(\lambda)$. Thus, its advantage in distinguishing $b = 0$ from $b = 1$ is at most $2(1/Q_1(\lambda) + \text{negl}(\lambda))$.

Because this holds for arbitrary polynomials $P_1(\cdot)$ and $Q_1(\cdot)$, every PPT adversary has at most negligible advantage. $\square$

A.7 Security Amplification for Regular $(1 - \delta)$ -Secure DPFs

Finally, we combine Lemma A.10 and Theorem A.11 to obtain the final amplification statement (Theorem 3.2), which upgrades weakly secure DPFs to fully secure ones.

Proof of Theorem 3.2. When ϵ is constant, the requirement $r := \frac{\epsilon \log N}{\log \log N + 2 \log \log \lambda + \log \epsilon + 2} \geq 1$ in Theorem A.11 is satisfied whenever $\log N \geq \frac{3}{\epsilon} \log \log \lambda$. If $\log N < \frac{3}{\epsilon} \log \log \lambda$, we can simply raise the domain size to $(\log \lambda)^{3/\epsilon}$. $\square$

B Deferred Details: Our All-But-One-Corrupt DPF

B.1 Proofs for Our New Random Bipartite Graphs

Now we prove Theorem 4.2, which is our new special bipartite graph design. We prove strong $(1 - \delta)$ -security for the graphs, which can be easily upgraded into strong $(1 - \delta)$ -security for the DPF scheme, so that we do not need to invoke the decoy lemma and suffer from the polylogarithmic overhead.

Proof of Theorem 4.2. Right vertices in G_0 have even degree. Every right vertex in G_0 is connected to exactly two left vertices: each set $s_{i,j}$ is connected to P_i and P_j .

Left vertices in G_1 have exclusive neighbors. For every left vertex P_i in G_1 , we add a unique right vertex s'_i exclusively connected to it.

Strong $(1 - \delta)$ -security. Let h be the index of the honest party. Let G'_0 and G'_1 denote the adversary's views of G_0 and G_1 , respectively. We claim that for any $x = 1, \dots, W$, the adversary's

view of G_1 conditioned on $X_h = x - 1$ (denoted $[G'_1 \mid X_h = x - 1]$) is identically distributed as the adversary's view of G_0 conditioned on $X_h = x$ (denoted $[G'_0 \mid X_h = x]$). Indeed, $[G'_1 \mid X_h = x - 1]$ is $[G'_0 \mid X_h = x - 1]$ plus one additional planted right vertex per visible left vertex. On the other hand, $[G'_0 \mid X_h = x]$ is $[G'_0 \mid X_h = x - 1]$ with $m - 1$ additional right vertices, each connected to the honest party m and one visible left vertex. Since the adversary cannot see the honest party's edges, these views are identical.

However, we also have to account for the cases $X_h = 0$ when sampling G_0 and $X_h = W$ when sampling G_1 . When $X_h = 0$ when sampling G_0 , the adversary sees no degree-1 vertex, which cannot happen for G_1 . We deal with this case by conditioning on the event that $X_h \geq 1$ when sampling G_0 , because the probability mass at 0 is negligible. When $X_h = W$ when sampling G_1 , the adversary sees $W + 1$ degree-1 vertices per left vertex, which cannot happen for G_0 . We deal with this case using the asymmetry of the strong computational indistinguishability definition. To show that $G'_1 \approx_\delta G'_0$, we only need to show that G'_1 is indistinguishable from outputting from G_0 with probability $1 - \delta$, so it is fine for G'_1 to take some values that G'_0 cannot with probability δ . Thus, we do not need $\Pr[X_h = W]$ to be negligible, which is different from the $\Pr[X_h = 0]$ case. Finally, we note that the probability mass at $X_h = 0$ is a special value, so we avoid this value entirely by conditioning on $X_h \geq 1$ for G_1 and $X_h \geq 2$ for G_0 .

Let E_1 be the event that when we sample G_1 , we sample $X_h \geq 1$. Let E_0 be the event that when we sample G_0 , we sample $X_h \geq 2$. These events each occur with probability $1 - \text{negl}(\lambda)$ because

$$\Pr[\neg E_1] \leq \Pr[\neg E_0] = (1 - \delta)^W + \delta(1 - \delta)^{W-1} = (1 - \delta)^{W-1} \leq \exp(-\delta(W - 1)) = \text{negl}(\lambda)$$

since we set $W = (\log \lambda)^2 / \delta$.

Note that for any $x \geq 2$, by definition of $\text{RTGeo}(W, 1 - \delta)$, we have $\frac{\Pr[X_h = x - 1]}{\Pr[X_h = x]} = 1 - \delta$. Even when we condition on the events E_1 and E_0 , this inequality also holds approximately because

$$\frac{\Pr[X_h = x - 1 \mid E_1]}{\Pr[X_h = x \mid E_0]} = \frac{\Pr[X_h = x - 1]}{\Pr[X_h = x]} \cdot \frac{\Pr[E_0]}{\Pr[E_1]} \geq (1 - \delta)(1 - \text{negl}(\lambda)) \geq 1 - \delta - \text{negl}(\lambda).$$

Thus, for any view v , we have

$$\begin{aligned} \Pr[G'_1 = v \mid E_1] &\geq \sum_{x=1}^{W-1} \Pr[X_h = x \mid E_1] \Pr[G'_1 = v \mid X_h = x] && \text{(dropping the } x = W \text{ term)} \\ &= \sum_{x=1}^{W-1} \Pr[X_h = x \mid E_1] \Pr[G'_0 = v \mid X_h = x + 1] && \text{(identical distribution)} \\ &= \sum_{x=2}^W \Pr[X_h = x - 1 \mid E_1] \Pr[G'_0 = v \mid X_h = x] \\ &\geq (1 - \delta - \text{negl}(\lambda)) \sum_{x=2}^W \Pr[X_h = x \mid E_0] \Pr[G'_0 = v \mid X_h = x] \\ &= (1 - \delta - \text{negl}(\lambda)) \Pr[G'_0 = v \mid E_0]. \end{aligned}$$

Thus, $[G'_1 \mid E_1] \approx_{\delta + \text{negl}(\lambda)} [G'_0 \mid E_0]$ by Fact A.7. By Fact A.8, this implies $[G'_1 \mid E_1] \approx_\delta [G'_0 \mid E_0]$. Because $G'_1 \approx_0 [G'_1 \mid E_1]$ and $[G'_0 \mid E_0] \approx_0 G'_0$, by transitivity, we have $G'_1 \approx_\delta G'_0$. $\square$

Remark 2. In Appendix E, we show that the special bipartite graphs of [GWW25] also satisfy strong $(1 - \delta)$ -security with similar efficiency as the original version with regular security. Therefore, even though they cited [BGIK22b]’s security amplification lemma that has an incomplete proof, their DPF construction is in fact secure even without using the decoy lemma.

B.2 Improved Transformation from Bipartite Graphs to DPF

In this section, we present a simple rebalancing trick to reduce the share size dependence on the number of parties when we transform special bipartite graphs into DPFs, compared to the construction of [GWW25]. This rebalancing trick has been implicitly used in the construction of Boyle et al. [BGI15], but was overlooked by [GWW25]. We also show how to use *strong* $(1 - \delta)$ -security of the bipartite graphs to prove *strong* $(1 - \delta)$ -security for resulting DPF. This way, we can get strong $(1 - \delta)$ -secure DPFs *without* using the decoy lemma, and thus avoid the polylogarithmic overhead from the decoys.

Intuition of the rebalancing trick [BGI15]. When we instantiate the multi-party DPF, there are two main contributors to the share size: the total size of the PRG seeds and the size of the correction words. By dividing the domain into a $\sqrt{N}$ by $\sqrt{N}$ grid, the construction of [GWW25] achieves a balance in terms of N . However, it does not consider the overhead due to the size of the PRG seeds λ_{PRG} and the number of PRG seeds per segment, which is a function of the number of parties. Because these overheads only apply to the PRG seeds and not the correction words, we can divide the domain into a grid with slightly fewer segments to reduce the share size dependence on the number of parties.

Lemma B.1. *Suppose we have special bipartite graphs for m parties and left-degree d with strong $(1 - \delta)$ -security. For domain size N , suppose we have a PRG $\text{PRG} : \{0, 1\}^{\lambda_{\text{PRG}}} \rightarrow \{0, 1\}^{\sqrt{d\lambda_{\text{PRG}}N}}$.*

Then, there exists an m -party, $(m - 1)$ -secure DPF for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:

- $(1 - \text{negl}(\lambda))$ -correctness;
- Strong $(1 - \delta)$ -security; and
- $O(\sqrt{d\lambda_{\text{PRG}}N})$ per-party share size.

Proof. The following DPF scheme is based on Theorem 1 of [GWW25]. The only modification we make is to reinterpret the N -length output vector of $f^{\alpha, \beta}$ as a $(\sqrt{N/(d\lambda_{\text{PRG}})}) \times (\sqrt{d\lambda_{\text{PRG}}N})$ matrix, rather than a $\sqrt{N} \times \sqrt{N}$ matrix.

$\text{Gen}(1^\lambda, \alpha, \beta)$

1. Interpret the target point α as a pair (α_1, α_2) , where $\alpha_1 \in [\sqrt{N/(d\lambda_{\text{PRG}})}]$ specifies the segment and $\alpha_2 \in [\sqrt{d\lambda_{\text{PRG}}N}]$ specifies the position inside that segment.
2. For every segment $\text{seg} \in [\sqrt{N/(d\lambda_{\text{PRG}})}]$ and every party $i \in [m]$, initialize an empty seed set $S_i^{(\text{seg})} \leftarrow \emptyset$.
3. For each segment $\text{seg} \in [\sqrt{N/(d\lambda_{\text{PRG}})}]$, generate and assign PRG seeds as follows.
 - (a) If $\text{seg} \neq \alpha_1$, freshly sample a graph G_0 .
 - (b) If $\text{seg} = \alpha_1$, freshly sample a graph G_1 .
 - (c) For every right vertex v in the sampled graph, sample an independent PRG seed $r_v^{(\text{seg})} \leftarrow \{0, 1\}^{\lambda_{\text{PRG}}}$.

- (d) For every party $i \in [m]$, if the right vertex v is connected to the i -th left vertex, add the seed $r_v^{(\text{seg})}$ to that party's seed set for segment seg , namely update $S_i^{(\text{seg})} \leftarrow S_i^{(\text{seg})} \cup \{r_v^{(\text{seg})}\}$.
4. For every segment $\text{seg} \in [\sqrt{N/(d\lambda_{\text{PRG}})}]$, sample random bits $\rho_1^{(\text{seg})}, \dots, \rho_m^{(\text{seg})}$ such that $\bigoplus_{i=1}^m \rho_i^{(\text{seg})} = 1$ if $\text{seg} = \alpha_1$, and $\bigoplus_{i=1}^m \rho_i^{(\text{seg})} = 0$ otherwise. These bits determine which parties apply the correction word.
5. Define the correction word $\text{cw} \in \{0, 1\}^{\sqrt{d\lambda_{\text{PRG}}N}}$ as follows.
 - (a) Initialize cw as the one-hot vector that is β on coordinate α_2 and is zero elsewhere.
 - (b) For each party $i \in [m]$, add all of its PRG evaluations for the segment α_1 to cw , that is, $\text{cw} \leftarrow \text{cw} \oplus \bigoplus_{r \in S_i^{(\alpha_1)}} \text{PRG}(r)$ for all $i \in [m]$.
6. Finally, output the i -share as

$$k_i = (\{S_i^{(\text{seg})}, \rho_i^{(\text{seg})}\}_{\text{seg} \in [\sqrt{N/(d\lambda_{\text{PRG}})}]}, \text{cw}).$$

Eval(k_i, x)

1. Interpret the query point x as a pair (x_1, x_2) , where $x_1 \in [\sqrt{N/(d\lambda_{\text{PRG}})}]$ and $x_2 \in [\sqrt{d\lambda_{\text{PRG}}N}]$.
2. Parse the key as $k_i = (\{S_i^{(\text{seg})}, \rho_i^{(\text{seg})}\}_{\text{seg} \in [\sqrt{N/(d\lambda_{\text{PRG}})}]}, \text{cw})$.
3. Compute the party's share for the x_1 -th segment, defined as

$$y_i^{(x_1)} = (\rho_i^{(x_1)} \cdot \text{cw}) \oplus \bigoplus_{r \in S_i^{(x_1)}} \text{PRG}(r).$$

4. Output the x_2 -th bit of $y_i^{(x_1)}$.

Efficiency. A share consists of a correction word in $\{0, 1\}^{\sqrt{d\lambda_{\text{PRG}}N}}$, $\sqrt{N/(d\lambda_{\text{PRG}})}$ bits $\rho_i^{(\text{seg})}$, and $d \cdot \sqrt{N/(d\lambda_{\text{PRG}})}$ PRG seeds (each with length λ_{PRG}). Thus, the total share size is $O(\sqrt{d\lambda_{\text{PRG}}N})$.

(1 - negl(λ))-correctness. By a union bound over all $\sqrt{N/(d\lambda_{\text{PRG}})}$ instances of G_0 and G_1 , with $1 - \text{negl}(\lambda)$ probability, for all non-target segments $\text{seg} \neq \alpha_1$, all PRG seeds appear an even number of times. Because the bits $\rho_i^{(\text{seg})}$ sum to 0 for $\text{seg} \neq \alpha_1$, the correction word is also XOR-ed an even number of times, so the sum of all shares for non-target segments is zero.

For the target segment α_1 , because the correction word is XOR-ed an odd number of times, all shares sum to the one-hot vector that is β at α_2 .

Strong (1 - δ)-security. By PRG security, the adversary's view is computationally indistinguishable from the view in which the pseudorandom outputs are replaced with random strings. We condition on the success of the required properties of the special bipartite graph, namely all right vertices of G_0 have even degree and all left vertices of G_1 have exclusive neighbors. This incurs a $\text{negl}(\lambda)$ statistical loss.

Fix an honest party, and let G'_0 and G'_1 denote the adversary's views of G_0 and G_1 , respectively. In G_1 , the honest party receives a PRG seed that the adversary does not see, so the correction word appears uniformly random to the adversary.

Thus, it suffices to consider the security of how the PRG seeds are assigned to parties. The adversary's view of the assignment of the PRG seeds is given by $(G'_0, \dots, G'_1, \dots, G'_0)$, where the

position of G'_1 is secret. Strong $(1-\delta)$ -security of the graphs gives $G'_1 \approx_\delta G'_0$, i.e., G'_1 is computationally indistinguishable from some distribution Z that outputs from G'_0 with probability $1-\delta$. Hence $(G'_0, \dots, G'_1, \dots, G'_0)$ is computationally indistinguishable from $(G'_0, \dots, Z, \dots, G'_0)$, which outputs from $(G'_0, \dots, G'_0, \dots, G'_0)$ with probability $1-\delta$. Thus, $(G'_0, \dots, G'_1, \dots, G'_0) \approx_\delta (G'_0, \dots, G'_0)$. Letting $\text{Sim}(1^\lambda)$ be $(G'_0, \dots, G'_0)$ proves strong $(1-\delta)$ -security for the DPF scheme. $\square$

Finally, we obtain Theorem 4.3 using the new special bipartite graphs (Theorem 4.2) to construct a DPF (Lemma B.1), and then applying the amplification theorem (Theorem 3.2). Note that we did not use the decoy lemma, which achieves a polylogarithmic saving in share size.

C Deferred Details: Honest-Majority DPF with Computational Security

In this section, we prove the following theorem, of which Theorem 5.1 is a special case.

Theorem C.1 (Honest-majority DPF based on OWF). *Assume the existence of OWF. Suppose we are given the following parameters which may be functions of λ : $\epsilon \in (0, 1)$ and some integer $D \geq 2$. Suppose also that $m > (1+\epsilon)Dt$. Then, there exists an m -party DPF tolerating up to t corruptions for any input domain $[N]$ and output range $\{0, 1\}$ satisfying the following properties where $\text{negl}(\cdot)$ is a suitable negligible function, λ_{PRG} denotes the seed length of the PRG, and λ denotes a separate statistical security parameter:*

- 1-correctness;
- $(1 - \text{negl}(\lambda) - \text{negl}(\lambda_{\text{PRG}}))$ -security; and
- $(\log(N, \lambda))^{\Theta(\frac{1}{\epsilon})} \cdot D^{\Theta(\log \log(N, \lambda)/\epsilon^3)} \cdot \sqrt{\lambda_{\text{PRG}}} \cdot N^{(1+\epsilon) \cdot \frac{1}{2D}}$ per-share size².

In Theorem C.1, if we choose $\epsilon \in (0, 1)$ to be a suitably small positive constant, and choose $D \geq 2$ to be a constant, we immediately get Theorem 5.1.

C.1 Compiler from All-But-One Corruption to Honest Majority

We now modify the result of Bunn et al. [BKO22] into a strong indistinguishability variant. More specifically, we prove the following lemma, which constructs an honest-majority DPF from an all-but-one-corrupt DPF. The resulting DPF has better dependence on the domain size N but significantly worse dependence on m and t .

Lemma C.2 (Strong computational indistinguishability variant of [BKO22]). *Let $D \geq 2$ be an integer, and suppose that m, t are positive integers satisfying $m \geq Dt + 1$. All above parameters may be functions of λ . Let $n = \lceil N^{1/D} \rceil$ and $M = \binom{m}{t}$, and suppose that M is polynomial bounded in λ . Suppose that there exists an M -party DPF for the small input domain $= [n]$ and any output range that is a ring with identity, satisfying the following properties:*

- $(1 - \gamma)$ -correctness;
- Strong $(1 - \delta)$ -security in the presence of up to $M - 1$ corruptions; and
- S per-share size.

Then, there is an m -party DPF for the larger domain $[N]$ and the same output range, satisfying the following properties:

²Theorem C.1 makes the implicit assumption that the parameters are chosen to guarantee that this expression is polynomially bounded.

- $(1 - D\gamma)$ -correctness;
- Strong $(1 - D\delta)$ -security in the presence of up to t corruptions; and
- $D \cdot M \cdot S$ per-share size.

C.1.1 Background: Construction by Bunn et al.

We first review the construction of Bunn et al. [BKO22].

Decompose a point function into a product of point functions with smaller domains.

Bunn et al. [BKO22]’s compiler works as follows. Suppose that $m > D \cdot t$. We encode the input x as a length- D vector $x := (x_1, \dots, x_D)$. Given a point function $f^{\alpha, \beta}(x)$ with input domain $[N]$, and decompose the point function into the product of D point functions $(f_1, \dots, f_D)$ where $(\alpha_1, \dots, \alpha_D)$ is the encoding of α , and $\prod_{d \in [D]} \beta_d = \beta$:

$$f^{\alpha, \beta} := \prod_{d \in [D]} f_d^{\alpha_d, \beta_d}(x_d) \quad (2)$$

Therefore, each underlying point function $f_d^{\alpha_d, \beta_d}(\cdot)$ has a smaller domain size $n := N^{1/D}$.

Instantiate each underlying point function with an M -party DPF. We will apply an M -party DPF denoted DPF_d for implementing each underlying point function $f_d^{\alpha_d, \beta_d}(\cdot)$. For each underlying DPF_d , its keys for all M parties are henceforth denoted $k_1^{(d)}, \dots, k_M^{(d)}$. It helps to imagine that there are M *virtual parties*, and each virtual party $j \in [M]$ gets the keys

$$K_j := \{k_j^{(d)}\}_{d \in [D]}.$$

Virtual party to actual party assignment. The key question is how to distribute the M virtual parties’ keys to the m actual parties. Bunn et al. [BKO22] proposed the following idea:

- Let $S_1, \dots, S_M$ be all subsets of $[m]$ of size t .
- Each actual party $i \in [m]$ obtains the keys K_j for virtual party $j \in [M]$ iff $i \notin S_j$.

This assignment satisfies the following two desirable properties:

P1 (*related to feasibility of evaluation.*) For any size D combination $(j_1, \dots, j_D) \in [M]^D$, there exists an actual party that possesses the keys of all virtual parties indexed by $j_1, \dots, j_D$.

P2 (*related to security.*) For any set of up to t corrupted (actual) parties, there exists a virtual party $j^* \in [M]$ whose keys K_{j^*} do not belong to any corrupt party.

Evaluation. For the evaluation, we rewrite Equation (2) as the following:

$$f^{\alpha, \beta}(x) = \prod_{d \in [D]} \left(\sum_{j \in [M]} \text{DPF}_d.\text{Eval}(k_j^{(d)}, x_d) \right)$$

When we expand the above, we get M^D monomials indexed by $(j_1, \dots, j_D) \in [M]^D$, and each monomial will be evaluated by a *unique* representative and accumulated to the sum. Observe that as long as the representative possess the keys of all virtual parties indexed $j_1, \dots, j_D$, then it is able to evaluate the corresponding monomial. The feasibility of distributed evaluation therefore follows from property **P1**.

We define the following notations:

- Let $S_1, S_2, \dots, S_M$ be all t -size subsets of $[m]$.
- Let $\tau : [N] \rightarrow [n]^D$ be any injective function that maps each element of $[N]$ to a unique tuple in $[n]^D$, written as $\tau(\alpha) = (\alpha_1, \alpha_2, \dots, \alpha_D)$.

Gen($1^\lambda, \alpha, \beta$) let $(\alpha_1, \alpha_2, \dots, \alpha_D) = \tau(\alpha)$, $\beta_1 = \beta$, and $\beta_2 = \beta_3 = \dots = \beta_D = 1$.

1. For all $d \in [D]$, compute $(k_1^{(d)}, \dots, k_M^{(d)}) \xleftarrow{\$} \text{DPF}_d.\text{Gen}(1^\lambda, \alpha_d, \beta_d)$.
2. Output the i -th share $k_i := \left(i, \{k_j^{(d)}\}_{i \notin S_j, d \in [D]}\right)$ for all $i \in [m]$.

Eval(k_i, x) parse $k_i := \left(i, \{k_j^{(d)}\}_{i \notin S_j, d \in [D]}\right)$ and let $(x_1, x_2, \dots, x_D) = \tau(x)$.

1. Let $a_i \leftarrow 0$, and for all $(j_1, j_2, \dots, j_D) \in [M]^D$,
 - (a) Compute the set $S := [m] - \bigcup_{d=1}^D S_{j_d}$.
 - (b) If i is the minimum element in S , set $a_i \leftarrow a_i + \prod_{d=1}^D \text{DPF}_d.\text{Eval}(k_{j_d}^{(d)}, x_d)$.
2. Output a_i .

Figure 3: The construction of Bunn et al. [BKO22].

Security. It is not hard to see that as long as the underlying DPFs are secure under all-but-one corruption and property **P2** holds, then the resulting DPF is secure.

Formal description. A formal presentation of Bunn et al. [BKO22]’s scheme can be found in Figure 3.

C.1.2 Proof of Lemma C.2

Bunn et al. [BKO22]’s proof assumes that the underlying DPFs are fully secure under all-but-one corruption. In our Lemma C.2, we instead have that the underlying DPFs satisfy strong $(1 - \delta)$ -security, and we want to prove that the resulting DPF satisfies strong $(1 - D\delta)$ -security. To do so, we proceed in the following hybrids.

- **Real.** Output the distribution of the adversary’s view in the real world.
- **Real’.** Fix the set of corrupt (actual) parties controlled by the adversary. Let $S \subset [M]$ be the set of virtual parties such that for any $j^* \in S$, the keys $K_{j^*} = \{k_{j^*}^{(d)}\}_{d \in [D]}$ do not belong to the adversary — due to property **P2** which is proven by Bunn et al. [BKO22], $|S| \geq 1$. Output the following distribution as the adversary’s view: for each $d \in [D]$, output the real distribution of keys belonging to the set of virtual parties $[M] \setminus S$ for the underlying DPF_d .
- **Ideal.** Almost the same as **Real’** except that for each $d \in [D]$, we now instead output the output of the simulator of DPF_d for the set of corrupt virtual parties $[M] \setminus S$.

It is not hard to see that **Real’** is identically distributed as **Real** since it is just an equivalent rewrite of **Real**. Further, due to the composition lemma (Fact A.6), we have that **Real’** $\approx_{D, \delta}$ **Ideal**. Observe also that **Ideal** implicitly defines a simulator that simulates the adversary’s view without knowing its real view. Therefore, the resulting scheme satisfies strong $(1 - D\delta)$ -security.

Finally, $(1 - D\gamma)$ -correctness follows from the correctness proof of Bunn et al. [BKO22] plus a simple union bound, and the share size calculation is the same as Bunn et al. [BKO22]. This concludes the proof of Lemma C.2.

C.2 Committee Selection

Committee selection allows us to significantly reduce the dependence of the share size on the number of parties, at the cost of only a small loss in security and corruption tolerance. The idea is as follows. We randomly sample a committee of size m from the M parties and distribute the keys only among the committee members. As long as the fraction of corrupted parties in the overall population is bounded by $(1 - \epsilon) \cdot c$, the probability that the committee contains more than a c fraction of corrupted members is exponentially small in $\epsilon^2 cm$. The lemma is stated as follows.

Lemma C.3 (Committee Selection). *Let $c, \epsilon, \gamma, \delta \in (0, 1)$ be functions of λ . Given some arbitrary domain and range, suppose there exists an m -party DPF satisfying the following properties:*

- $(1 - \gamma)$ -correctness;
- Strong $(1 - \delta)$ -security in the presence of up to $c \cdot m$ corruptions;
- S per-share size.

Then, for any $M \geq m$, there exists an M -party DPF for the same domain and range, satisfying the following properties.

- $(1 - \gamma)$ -correctness;
- Strong $(1 - \delta')$ -security in the presence of up to $(1 - \epsilon) \cdot cM$ corruptions, where $\delta' = \delta + \exp(-\Omega(\epsilon^2 \cdot c \cdot m))$;
- S per-share size.

Construction. Lemma C.3 follows from randomly sampling a committee of size m as shown in the construction below. We will show that as long as in the overall population, the corrupt fraction is bounded by $(1 - \epsilon) \cdot c$, then except with probability $\exp(-\Omega(\epsilon^2 \cdot c \cdot m))$, the committee's corrupt fraction is bounded by c ; and this failure probability contributes additively to the leakage of the resulting DPF.

Let $\text{DPF} = (\text{Gen}, \text{Eval})$ be the underlying DPF.

$\text{Gen}'(1^\lambda, \alpha, \beta)$

1. Sample a committee $C \subseteq [M]$ of size m (using sampling without replacement), and fix an arbitrary ordering of C .
2. For $i \notin C$, let $k_i := \perp$, and for all committee members, generate their keys by running the underlying DPF's generator:

$$\{k_i\}_{i \in C} \leftarrow \text{Gen}(1^\lambda, \alpha, \beta).$$

3. Output $(k_1, \dots, k_M)$.

$\text{Eval}'(k_i, x)$ If $k_i = \perp$, output 0; else output $\text{Eval}(k_i, x)$.

Proof. Correctness, key size, and reconstruction properties follow directly from the construction. Therefore, it suffices to prove strong computational indistinguishability.

Due to the Chernoff bound for sampling without replacement [Hoe63], the probability that the committee C has more than cm corrupt members is upper bounded by

$$\Pr[\text{Committee } C \text{ has more than } cm \text{ corrupt}] \leq \exp(-\Omega(\epsilon^2 cm)) \quad (3)$$

To prove strong indistinguishability, consider the following sequence of hybrids.

- **Real.** Let **Real** be the real view of the adversary.
- **Hyb.** Let **Hyb** be the following distribution: randomly sample a committee of size m subject to the constraint that the number of corrupt committee members be bounded by $c \cdot m$. Next, output the adversary's view as follows:
 - For the corrupt committee members, output the real-world adversary's view of the underlying DPF.
 - For every corrupt non-committee member, output $\perp$ as the view.
- **Ideal.** Let **Ideal** be the following distribution: randomly sample a committee of size m subject to the constraint that the number of corrupt committee members be bounded by $c \cdot m$. Next, output the adversary's view as follows:
 - For the corrupt committee members, call the underlying DPF's simulator and output the joint view.
 - For every corrupt non-committee member, output $\perp$ as its view.

By definition and Equation (3), we trivially have that **Real** $\approx_{\exp(-\Omega(\epsilon^2 \cdot c \cdot m))}$ **Hyb**. Further, since the underlying DPF satisfies $(1 - \delta)$ -computational indistinguishability, by the post-processing lemma (Fact A.5), we have that **Hyb** $\approx_{\delta}$ **Ideal**. By transitivity (Fact A.4), we have that **Real** $\approx_{\delta'}$ **Ideal** where $\delta' := \delta + \exp(-\Omega(\epsilon^2 \cdot c \cdot m))$. Finally, observe that **Ideal** implicitly defines a simulator which need not know the adversary's actual view to output the simulated distribution. In conclusion, the resulting DPF scheme satisfies $(1 - \delta')$ -strong indistinguishability. $\square$

C.3 Putting Everything Together

Figure 4 shows how we put everything together and obtain Theorem C.1.

D Improved DPF: Honest-Majority Setting with Perfect Security

We now present a simple information-theoretically secure scheme based on Shamir secret sharing. Our construction is a small tweak of the DPF of [PLM⁺24, Section 4.3] without Hermite interpolation. The main difference is that we require DPF reconstruction to be additive, so we need to multiply the shares by the Lagrange weights during **Eval** instead of running Hermite interpolation during reconstruction.

Theorem D.1. *Suppose we are given parameters $N, m \in \mathbb{N}$ and an integer $D \in [2, m - 1]$. Then there exists an m -party DPF tolerating up to $\lfloor (m - 1)/D \rfloor$ corruptions for domain $[N]$ and range $\{0, 1\}$, with the following properties:*

- 1-correctness;
- perfect security; and
- $O(DN^{1/D} \log m)$ per-share size.

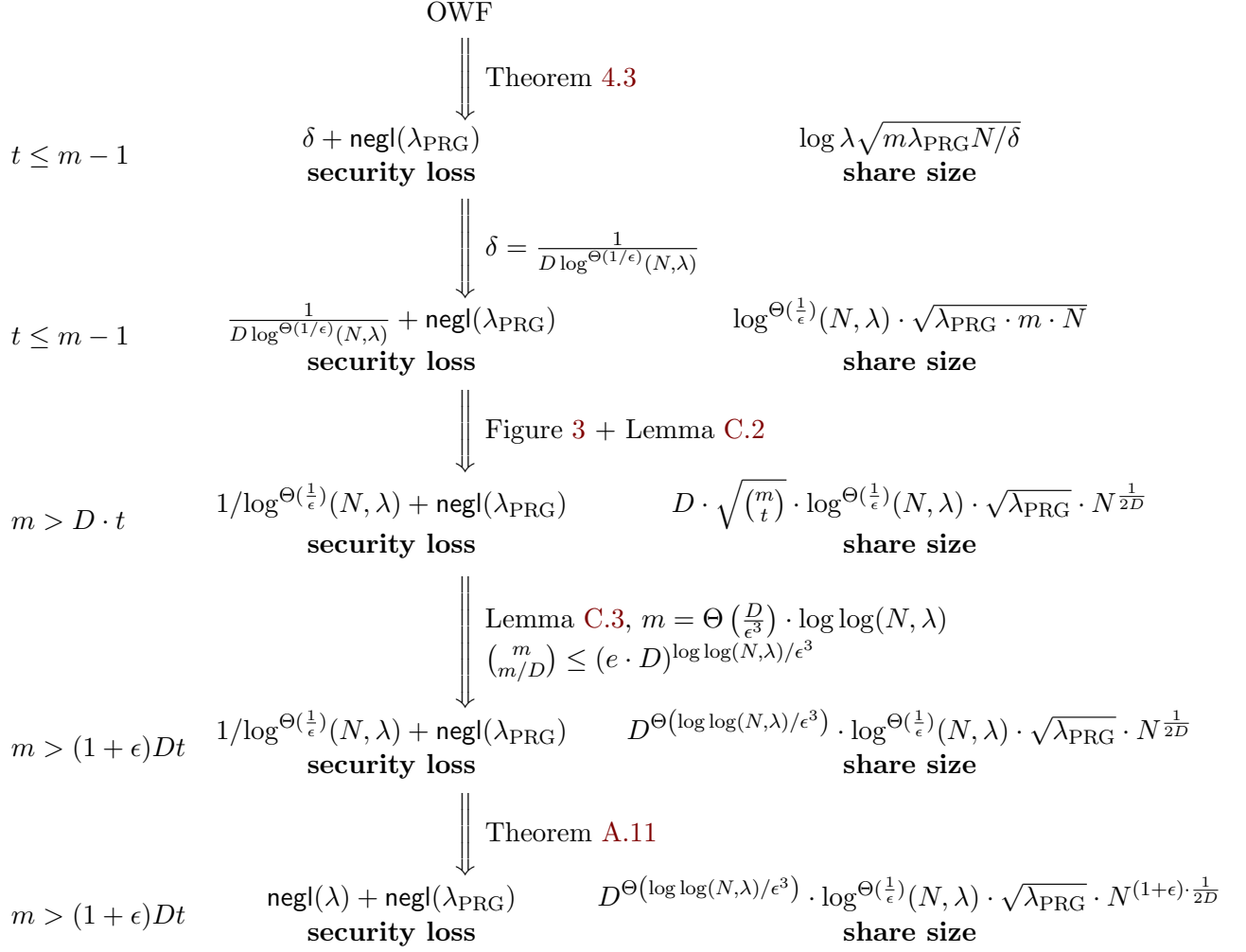


Figure 4: Our construction in the honest-majority setting.

Intuition. The construction views the domain as a D -dimensional grid $[n]^D$, where $n = N^{1/D}$. A target point α is written as $\alpha = (\alpha_1, \dots, \alpha_D)$. Instead of sharing the full length- N point function, we share D length- n one-hot vectors: for each $d \in [D]$, let $\mathbf{u}^{(d)} = (u_1^{(d)}, \dots, u_n^{(d)}) \in \mathbb{F}^n$ be the one-hot vector for the position $\alpha_d \in [n]$. Then, for any query point $x = (x_1, \dots, x_D)$, we can express the desired point function as $f^{\alpha, \beta}(x) = \prod_{d \in [D]} u_{x_d}^{(d)}$, which is 1 exactly when all coordinates of x match α , and is 0 otherwise.

To generate the shares, we Shamir-share every position $j \in [n]$ of the D short vectors $\mathbf{u}^{(d)}$ using degree- t polynomials $U_j^{(d)}(X)$, so each party receives Dn field elements in total. To evaluate at x , a party looks up its shares of $u_{x_1}^{(1)}, \dots, u_{x_D}^{(D)}$ and multiplies them locally. These local products are evaluations of the product of the Shamir polynomials, namely $\prod_{d \in [D]} U_{x_d}^{(d)}(X)$, whose constant term is exactly $f^{\alpha, \beta}(x)$. However, multiplying D degree- t polynomials gives degree at most Dt , so the product can be reconstructed from all m parties only if $Dt \leq m - 1$. This motivates setting $t = \lfloor (m - 1)/D \rfloor$. The initial Shamir shares hide the target point against any t corrupted parties, while the multiplied shares allow reconstruction from the evaluations of all m parties. Finally, by

the linear reconstruction formula for Shamir sharing, each party multiplies its local product by the appropriate public Lagrange coefficient. The resulting values are additive shares of $f^{\alpha,\beta}(x)$.

Consider the following parameter setting:

- The threshold of the number of corrupt parties $t := \lfloor (m-1)/D \rfloor$.
- Let $n := \lceil N^{1/D} \rceil$.
- Let $\mathbb{F}$ be the smallest finite field of characteristic 2 with $|\mathbb{F}| > m$.
- Let $e_1, \dots, e_m$ be fixed distinct nonzero points in $\mathbb{F}$.
- Let $E : [N] \rightarrow [n]^D$ be any fixed injection.
- Fix any additive homomorphism $\phi : \mathbb{F} \rightarrow \mathbb{Z}_2$ such that $\phi(1) = 1$, such as the field trace.

Gen($1^\lambda, \alpha, \beta$) write $E(\alpha) = (\alpha_1, \dots, \alpha_D) \in [n]^D$.

1. For every $d \in [D]$, define a one-hot vector $\mathbf{u}^{(d)} = (u_1^{(d)}, \dots, u_n^{(d)}) \in \mathbb{F}^n$ in the following way. For $d = 1$, we define $\mathbf{u}^{(1)}$ to be the one-hot vector that is β at position α_1 and is 0 elsewhere. For $d = 2, \dots, D$, we define $\mathbf{u}^{(d)}$ to be the one-hot vector that is 1 at position α_d and is 0 elsewhere.
2. For every $d \in [D]$ and position $j \in [n]$, run a Shamir secret sharing scheme for the scalar $u_j^{(d)}$. That is, independently sample a uniformly random degree- t polynomial $U_j^{(d)}(X) \in \mathbb{F}[X]$ subject to $U_j^{(d)}(0) = u_j^{(d)}$.
3. For all $i \in [m]$, output the i -th share $k_i := \left(i, (U_j^{(d)}(e_i))_{d \in [D], j \in [n]} \right)$, consisting of the index i and the i -th share of every Shamir secret sharing scheme.

Eval(k_i, x) write $E(x) = (x_1, \dots, x_D) \in [n]^D$, and parse k_i as $\left(i, (U_j^{(d)}(e_i))_{d \in [D], j \in [n]} \right)$.

1. Define the Lagrange polynomial $L_i(X) := \prod_{i' \in [m], i' \neq i} \frac{X - e_{i'}}{e_i - e_{i'}}$.
2. Output $\phi \left(L_i(0) \prod_{d \in [D]} U_{x_d}^{(d)}(e_i) \right)$.

Efficiency. Every share consists of $Dn + 1$ elements of $\mathbb{F}$, where $n = \lceil N^{1/D} \rceil$ and $|\mathbb{F}| = O(m)$. Thus, the share size is $O(DN^{1/D} \log m)$.

1-correctness. The polynomial $\prod_{d \in [D]} U_{x_d}^{(d)}(X)$ has degree at most $tD \leq m - 1$. By Lagrange interpolation,

$$\prod_{d \in [D]} U_{x_d}^{(d)}(X) = \sum_{i \in [m]} L_i(X) \prod_{d \in [D]} U_{x_d}^{(d)}(e_i).$$

Thus, the sum of the shares is

$$\phi \left(\sum_{i \in [m]} L_i(0) \prod_{d \in [D]} U_{x_d}^{(d)}(e_i) \right) = \phi \left(\prod_{d \in [D]} U_{x_d}^{(d)}(0) \right) = \phi \left(\prod_{d \in [D]} u_{x_d}^{(d)} \right) = \begin{cases} \beta & x = \alpha \\ 0 & x \neq \alpha \end{cases}.$$

Perfect security. For every $j \in [n]$ and $d \in [D]$, the polynomial $U_j^{(d)}$ has degree t with a fixed value at 0, so its values on any t evaluation points are uniformly random.

E Strong Computational Indistinguishability for the Special Bipartite Graph Design of Goel *et al.*

In fact, the special bipartite graphs in [GWW25] also satisfy strong computational indistinguishability. The parameter we obtain is a $\log(\lambda)$ -factor worse than the statistical distance bound in the paper, but it is still better than the polylogarithmic blowup caused by the black-box usage of the decoy lemma.

Lemma E.1. *For B the number of right vertices in G_0 and b the number of additional right vertices in G_1 , suppose that $B > \log^2 \lambda$, $B > b^2$ and $b = \omega(m \log(\lambda))$. Then the construction in Theorem 3 of [GWW25] is a special bipartite graph design for m parties and left-degree $O(\frac{B+b}{m})$, with strong $\left(1 - O\left(\frac{bm \log \lambda}{\sqrt{B}}\right)\right)$ -security.*

Proof. Recall that G_0 is sampled by independently connecting each of the B right vertices with two random left vertices, sampled with replacement, and G_1 is sampled by additionally creating b right vertices, each connected with one random left vertex.

Left-degree. By a Chernoff bound and a union bound, with $1 - \text{negl}(\lambda)$ probability, every left vertex has at most $O((B + b)/m)$ neighbors.

Right vertices in G_0 have even degree. This is always satisfied by design.

Left vertices in G_1 have exclusive neighbors. Because there are $b = \omega(m \log(\lambda))$ degree-1 vertices in G_1 , by a Chernoff bound, with $1 - \text{negl}(\lambda)$ probability, every left vertex in G_1 gets at least one exclusive neighbor.

Strong $\left(1 - O\left(\frac{bm \log \lambda}{\sqrt{B}}\right)\right)$ -security.

By the proof of Theorem 3 in the original paper [GWW25], if we fix the number of balls tossed twice and the number of balls tossed once, then the adversary's view of G_1 and G_0 are identically distributed. Fix the number of balls tossed twice to be D . Then the number of balls tossed once in the view of G_0 is $S \sim \text{Bin}(B - D, \frac{2(m-1)}{2m-1})$, and the number of balls tossed once in the view of G_1 is $S' := S_1 + T$ where $S_1 \sim \text{Bin}(B - D, \frac{2(m-1)}{2m-1})$ and $T \sim \text{Bin}(b, \frac{m-1}{m})$. Thus, by Fact A.7, it suffices to show that conditioning on some $(1 - \text{negl}(\lambda))$ -probability event E ,

$$\frac{\Pr[S' = s \mid E]}{\Pr[S = s \mid E]} \geq 1 - O\left(\frac{bm \log \lambda}{\sqrt{B}}\right)$$

for all D and s .

Because $D \sim \text{Bin}(B, (\frac{m-1}{m}))$, by a Chernoff bound, $|D - \mathbb{E}[D]| < \sqrt{\mathbb{E}[D]} \log \lambda$ with $1 - \text{negl}(\lambda)$ probability. We fix any D in this range. Then $|S - \mathbb{E}[S]| \leq \frac{1}{2} \sqrt{\mathbb{E}[S]} \log \lambda$ and $|S_1 - \mathbb{E}[S_1]| \leq \frac{1}{2} \sqrt{\mathbb{E}[S_1]} \log \lambda$ with $1 - \text{negl}(\lambda)$ probability. From now on, we condition on all these events happening (which has $1 - \text{negl}(\lambda)$ probability) and we drop the conditioning notation for brevity.

We first observe that S' is a Poisson binomial distribution, so it is unimodal, i.e. $\Pr[S' = s]$ increases as s increases until the mode, which is either the floor or ceiling of $\mathbb{E}[S']$, and then decreases. Similarly, S is also unimodal with mode $\mathbb{E}[S] < \mathbb{E}[S']$. Then we claim that $\frac{\Pr[S' = s]}{\Pr[S = s]}$ is minimized when $s \leq \mathbb{E}[S]$, since for $\mathbb{E}[S] < s < \mathbb{E}[S']$, $\Pr[S' = s]$ increasing and $\Pr[S = s]$ decreasing, and for $s \geq \mathbb{E}[S']$, we have $\frac{\Pr[S' = s]}{\Pr[S = s]} \geq 1$. Thus, we only need to bound $\frac{\Pr[S' = s]}{\Pr[S = s]}$ for $\mathbb{E}[S] - \frac{1}{2} \sqrt{\mathbb{E}[S]} \log \lambda \leq s \leq \mathbb{E}[S]$.

Next, we claim that for any $\mathbb{E}[S] - \sqrt{\mathbb{E}[S]} \log \lambda \leq s \leq \mathbb{E}[S]$,

$$\frac{\Pr[S = s - 1]}{\Pr[S = s]} = 1 - O\left(\frac{m \log \lambda}{\sqrt{B}}\right).$$

To see this, let $a := B - D$ and $\rho := \frac{2(m-1)}{2m-1}$ for simplicity, i.e. $S \sim \text{Bin}(a, \rho)$,

$$\begin{aligned} \frac{\Pr[S = s - 1]}{\Pr[S = s]} &= \frac{s(1 - \rho)}{(a - s + 1)\rho} \\ &\geq \frac{(a\rho - \sqrt{a\rho} \log \lambda)(1 - \rho)}{(a - (a\rho - \sqrt{a\rho} \log \lambda) + 1)\rho} && \text{(this function is increasing in } s) \\ &= 1 - \frac{\log \lambda - \rho/\sqrt{a\rho}}{\sqrt{a\rho}(1 - \rho) + \rho \log \lambda + \rho/\sqrt{a\rho}} \\ &= 1 - O\left(\frac{\log \lambda}{\sqrt{a}(1 - \rho)}\right) && \text{(because } \rho > 1/2, \sqrt{a} \gg \log \lambda) \\ &= 1 - O\left(\frac{m \log \lambda}{\sqrt{B}}\right). && \text{(because } D \ll B) \end{aligned}$$

Then we have

$$\begin{aligned} \Pr[S' = s] &= \sum_{k=0}^b \Pr[S = s - k] \Pr[T = k] \\ &= \sum_{k=0}^b \left(1 - O\left(\frac{m \log \lambda}{\sqrt{B}}\right)\right)^k \Pr[S = s] \Pr[T = k] && \text{(because } b < \frac{1}{2} \sqrt{\mathbb{E}[S]} \log \lambda) \\ &= \sum_{k=0}^b \left(1 - O\left(\frac{km \log \lambda}{\sqrt{B}}\right)\right) \Pr[S = s] \Pr[T = k] \\ &= \left(1 - O\left(\frac{tm \log \lambda}{\sqrt{B}}\right)\right) \Pr[S = s]. \end{aligned}$$

□

Corollary E.2. *For any real number $0 < \delta < 1$ and security parameter λ , we can construct special bipartite graphs for m parties and left-degree $O((\log(\lambda))^6 \cdot m^3/\delta^2)$ with strong $(1 - \delta)$ -security.*

Proof. We set the following parameters in Lemma E.1:

$$b = \Omega(m \cdot \log^2 \lambda), \quad B = \Omega((bm \log \lambda/\delta)^2) = (\log(\lambda))^6 \cdot m^4/\delta^2.$$

□

Combining Lemma B.1, Corollary E.2, and the amplification theorem (Theorem 3.2), we obtain the following fully secure DPF.

Corollary E.3. *For an arbitrarily small constant $0 < \epsilon < 1$, number of parties m , domain size N , and security parameter λ , suppose that we have a PRG $\text{PRG} : \{0, 1\}^{\lambda_{\text{PRG}}} \rightarrow \{0, 1\}^{\text{poly}(\log \lambda, m, \lambda_{\text{PRG}}, N)}$. Then we can construct an m -party, $(m - 1)$ -secure DPF for domain $[N]$ and range $\mathbb{Z}_2$ with the following properties:*

- 1-correctness;
- $(1 - \text{negl})$ -security; and
- $(\log(N, \lambda))^{O(1/\epsilon)} \sqrt{m^3 \lambda_{\text{PRG}} N^{1+\epsilon}}$ per-party share size.