Shuffling is Universal: Statistical Additive Randomized Encodings for All Functions

Nir Bitansky[1], Saroja Erabelli[1], Rachit Garg[1], and Yuval Ishai[2]

[1]New York University
[2]Technion and AWS
nbitansky@gmail.com, saroja.erabelli@gmail.com,
rg5134@cims.nyu.edu, yuval.ishai@gmail.com

Abstract

The shuffle model is a widely used abstraction for non-interactive anonymous communication. It allows n parties holding private inputs $x_1, \ldots, x_n$ to simultaneously send messages to an evaluator, so that the messages are received in a random order. The evaluator can then compute a joint function $f(x_1, \ldots, x_n)$, ideally while learning nothing else about the private inputs. The model has become increasingly popular both in cryptography, as an alternative to non-interactive secure computation in trusted setup models, and even more so in differential privacy, as an intermediate between the high-privacy, little-utility *local model* and the little-privacy, high-utility *central curator model*.

The main open question in this context is which functions f can be computed in the shuffle model with *statistical security*. While general feasibility results were obtained using public-key cryptography, the question of statistical security has remained elusive. The common conjecture has been that even relatively simple functions cannot be computed with statistical security in the shuffle model.

We refute this conjecture, showing that *all* functions can be computed in the shuffle model with statistical security. In particular, any differentially private mechanism in the central curator model can also be realized in the shuffle model with essentially the same utility, and while the evaluator learns nothing beyond the central model result.

This feasibility result is obtained by constructing a statistically secure *additive randomized encoding* (ARE) for any function. An ARE randomly maps individual inputs to group elements whose sum only reveals the function output. Similarly to other types of randomized encoding of functions, our statistical ARE is efficient for functions in NC^1 or NL. Alternatively, we get computationally secure ARE for all polynomial-time functions using a one-way function. More generally, we can convert any (information-theoretic or computational) "garbling scheme" to an ARE with a constant-factor size overhead.

1 Introduction

For almost four decades, secure multi-party computation (MPC) [Yao86, GMW19, BGW88, CCD88] has been one of the most thoroughly studied areas in cryptography. As such, the feasibility of MPC in various models is by now well understood. This is especially true in the information-theoretic setting where no computational restrictions are considered.

One notable exception is *non-interactive MPC in the shuffle model* [IKOS06], whose information-theoretic feasibility is still open. In this model, n parties holding private inputs $x_1, \ldots, x_n$ can independently send (multiple) anonymous messages to a server. The server should then recover the result $f(x_1, \ldots, x_n)$ of applying the prescribed function f, but should learn essentially nothing beyond the output. In particular, no a-priori setup or correlation between the parties is required as in other models for non-interactive MPC [FKN94, BGI+14, HIJ+17, AAP19]. Our focus is on the *semi-honest* model with *no insiders*; namely, the parties follow the protocol and do not collude with the server (see discussion on the stronger *robust* model in the related work section below).

The shuffle model has been considered in several cryptographic contexts, like private information retrieval and secure aggregation [IKOS06, GMPV20, BBGN20, IKLM24, GIK+24], but perhaps gained the most interest in the context of *differential privacy* (DP). In DP, the shuffle model is seen as an intermediate between *the local model* where no trust is assumed and *the central curator model* where the curator aggregating the data needs to be fully trusted by all parties. Several works have demonstrated that shuffle privacy can often offer better utility than local privacy. In fact, while the common belief seems to be that the model of shuffle privacy is weaker than that of central privacy, a general separation is not known. Separations have only been shown under additional structural constraints (or for a stronger robust notion of privacy) [BCJM21, BEM+17, BBGN19, CSU+19, BHNS20, Che21, GDDS21, KH+21, SCM22].

Additive Randomized Encodings. An intimately related notion to shuffle MPC is that of *additive randomized encodings* (ARE) [HIKR23]. An ARE for a k-party function $f(x_1, \ldots, x_k)$, allows each party i to locally compute a randomized encoding $\widehat{x}_i$ of their input x_i over an Abelian group $\mathbb{G}$. The encodings are then added together in $\mathbb{G}$ to yield a sum encoding $\widehat{w} = \widehat{x}_1 + \cdots + \widehat{x}_k$ of the output $w = f(x_1, \ldots, x_k)$.

A server, given the sum encoding $\widehat{w}$ can decode the output w, but learns no additional information; namely, the sum encoding can be simulated from the output w. ARE follow the general paradigm of randomized encodings of functions (RE) [Yao86, IK00, AIK06], and their multi-party version (MPRE) [ABT21], meant to reduce secure computation of potentially complex functions f to simpler ones. The relation to shuffle MPC and DP stems from the fact that shuffling is sufficient for secure addition [IKOS06].[1] Hence any function f that admits an ARE can be securely computed in the shuffle model.

Having introduced ARE in [HIKR23], the authors also gave several feasibility results, depending on the ARE security (simulation accuracy): *perfect*, *statistical*, or *computational*. In the computational setting, they showed that any efficient function has a computationally-secure ARE under a Diffie-Hellman type assumption in bilinear groups. The assumption was then reduced to any public-key encryption scheme in [BEG25]. In sharp contrast, for perfect security, ARE exist only for degenerate functions, essentially OR/XOR plus local preprocessing/postprocessing [HIKR23, Hiw25].

At the same time, the question of statistical ARE has remained elusive. In [HIKR23] the authors strongly conjecture that statistical ARE do not exist (even for simple functions as equality over small domains). Indeed, their conjecture is implied by the conjecture that shuffle privacy is weaker than central privacy, and accordingly was viewed as a possible stepping stone toward separating the two models. However, as in the privacy case, separation attempts have only ruled out variants of ARE, such as ℓ_2-ARE [HIKR23, Hiw25].

[1] In fact, a converse statement also holds. See Appendix A in [Che21].

1.1 This Work

We show that statistical ARE actually do exist for all functions.

Theorem 1.1. *For any multi-party $f : D^k \to D$ over a finite domain D^k and any error bound $\varepsilon > 0$, there exists an ARE for f with statistical correctness and security errors at most ε.*

We turn to study the minimal *size* of an ARE for f, defined as the total bit-length of the encoded inputs.[2] By applying a variant of amplification techniques from the literature [HIKN08, IKOS09], we establish a close relation between the ARE size of f and its *garbling* size, captured by the notion of *decomposable randomized encoding* (DRE).[3] A DRE for $f : \{0,1\}^k \to D$ is a randomized function of the form $g(x, r) = (g_1(x_1, r), \ldots, g_k(x_k, r))$ whose output is a randomized encoding of $f(x)$.

Theorem 1.2 (From DRE to ARE, informal)**.** *Let $f : (\{0,1\}^n)^k \to \{0,1\}^m$ be a k-party function family, for polynomially bounded $k = k(\tau), n = n(\tau), m = m(\tau)$. Assume that f, viewed as function over nk-bit strings, has a DRE with statistical correctness and security errors at most $\varepsilon = \varepsilon(\tau)$, where each output g_i is of size $c = c(\tau)$. Then the k-party f (over n-bit inputs) has an ARE of size $O(cnk)$ (i.e., $O(c)$ per input bit) and statistical correctness and security errors at most $\varepsilon + 2^{-cn}$.*

The same holds if the DRE is computationally secure, in which case the ARE is computationally secure.

Using DRE constructions from the literature [Yao86, FKN94, IK97, BKN18], we obtain the following:

- Any $f \in NL/\mathrm{poly}$ has a $\mathrm{poly}(\tau)$-efficient ARE with statistical correctness and security errors $2^{-\tau}$.
- Assuming one-way functions, any $f \in P/\mathrm{poly}$, has a $\mathrm{poly}(\tau)$-efficient ARE with statistical correctness error $2^{-\tau}$ and computational security. More concretely, if τ is the seed length of a pseudorandom generator, then any circuit of size S admits a computationally secure ARE of size $O(\tau \cdot S)$.
- For any finite f, there is an ARE for m copies of f that is of size $O(m)$ and statistical correctness and security errors 2^{-m}. More generally, any $f \in NC^0$ of output size m has such an ARE.
- Any function f has an ARE of size sublinear in its truth table, $2^{nk/2} \cdot \mathrm{poly}(\tau)$, and statistical correctness and security errors $2^{-\tau}$.

Theorem 1.2 is essentially optimal in the sense that an ARE of size c for a function $f : \{0,1\}^k \to \{0,1\}^m$ directly implies a DRE for f of size c (see Corollary 2 in Section 5.4). In particular, extending our efficient information-theoretic ARE for NL/poly to capture all polynomial-time functions would require solving a long-standing open problem in information-theoretic cryptography [FKN94]. We also note that the P/poly construction only makes black-box use of one-way functions, which should be contrasted with the computational ARE from public-key encryption [BEG25] that are non-black-box.

We turn to discuss several useful implications of the above results.

[2] To properly account for the communication cost of ARE in applications, if $\widehat{x}_i$ is always in a subgroup $\mathbb{G}_i$, its size is $\log|\mathbb{G}_i|$ (see Definition 5.2). When the number of parties is constant, one can think of the size as simply $\log|\mathbb{G}|$.

[3] A DRE is sometimes referred to in the literature as a multiparty *private simultaneous messages* (PSM) protocol [FKN94, IK97] or as a *garbling scheme* [Yao86, BHR12].

Implication 1: Shuffle MPC and Shuffle Privacy. The main implication of the statistical ARE construction, when combined with shuffle addition [IKOS06], is information-theoretic non-interactive shuffle MPC for all functions. This in particular yields a (non-interactive) transformation from any differentially private algorithm in the central curator model to one in the shuffle model, with essentially the same utility and privacy.

Implication 2: MPC in the Client-Server Model. Suppose that n *clients*, holding private inputs $x_1, \ldots, x_n$, employ m *servers* to securely compute a function f of their private inputs. We aim to obtain a non-interactive protocol in which each client sends a single message to each server who locally processes its received messages and sends a single message to an *evaluator*, who after some local processing outputs the result $f(x_1, \ldots, x_n)$. Here we assume that the evaluator may be corrupted and ask what is the tradeoff between the fraction of (semi-honestly) corrupted clients, corrupted servers, and the level of security. Previous information-theoretic results focus on the setting in which a minority of servers are corrupted and aim to maximize the fraction of allowed client corruptions [BIW10, AIKP22].

Using ARE, we obtain a new feasibility for the extreme case where any strict subset of the servers can be corrupted, but no clients are corrupted. To obtain this, the clients each use the ARE to locally encode their input and additively secret-share their encodings among the servers. The servers locally add all the shares, and send the sum to the evaluator.

Prior to our work, such client-server protocols were possible in the computational setting, making non-black-box use of oblivious transfer (OT) [BL18, GS18, GIS18] or public-key encryption [BEG25]. (The OT-based computational protocols also allow insiders, namely the evaluator may collude with input clients. However, in the information-theoretic setting this is impossible and in fact necessitates OT.)

Implication 3: Visual Secure Computation. In *visual secret sharing* [NS95], an arbitrary image can be split into two images that look totally random, but combining them using a bit-wise OR of their pixels (e.g. using transparencies) yields the original picture.

Our ARE construction yields a conceptually similar notion of *visual secure computation*. Specifically, our ARE construction implies a reduction from securely evaluating an arbitrary two-party function $f(x, y)$ to securely evaluating a sequence of bitwise two-party ORs (see details in the technical overview below). For instance, imagine two distant parties A, B in outer space holding private inputs x, y would like to securely and non-interactively communicate a joint function $f(x, y)$ to a third distant party C. Then having C hold a light sensor, A, B can each shoot a laser beam at C's sensor depending on the encoded bit, the sensor detects the existence of light but not its source (or number of sources). Performing this for every bit of the encoding, C can then decode $f(x, y)$ and nothing else.

1.2 Technical Overview

In a nutshell, our construction follows an approach taken in [BEG25] toward the construction of ARE:

1. Identify a relaxed notion of Leaky ARE (LARE) that allows some *leakage* on private inputs, and which is relatively simple to construct.

2. Lift the LARE construction to a full-fledged (non-leaky) ARE.

In [BEG25], the authors identify such a notion (called *one-sided ARE*), which they obtain information theoretically. Then they use *public-key encryption* to lift it to a full-fledged (computational) ARE.

In this work, we identify a stronger notion of LARE, which we are still able to construct information theoretically. We then lift our stronger notion of LARE to full-fledged ARE, but this time, *without relying on any computational assumptions*. Both steps are rather simple, and rely on elementary tools.

We next explain the two steps in more detail. Throughout, we focus on the two-party setting. Indeed, as we later discuss, any two-party ARE can then be made multi-party using DRE [HIKR23].

Our Notion of Leaky ARE. In a LARE the security requirement is relaxed so that in addition to the function's output $f(x,y)$, the (sum) encoding $\widehat{x}+\widehat{y}$ may also leak some additional information $\ell(x,y)$. Formally, $\widehat{x}+\widehat{y}$ can be statistically simulated given $f(x,y)$ and $\ell(x,y)$. Our notion of LARE allows a rather specific leakage function:

$$\ell(x,y)=\begin{cases} x \text{ if } f(x,y)=1 \\ \perp \text{ if } f(x,y)=0 \end{cases} ,$$

namely, when the output is 1, the encoding may leak the input x (but nothing on y), and when the output is 0, nothing leaks.

We first explain why this notion is useful; that is, how it can be lifted to full-fledged ARE. Then we explain how to construct LARE.

From LARE to ARE. Let $f : [d] \times [d] \to \{0,1\}$ be a Boolean function where both inputs x, y are taken from some finite domain, which is w.l.o.g represented by integers $[d] = \{1,\ldots,d\}$. We show how to construct an ARE for f given a LARE for the function $g : \mathbb{F}_2^d \times (\mathbb{F}_2^d \times \mathbb{F}_2) \to \mathbb{F}_2$

$$(\boldsymbol{a},(\boldsymbol{b},r)) \overset{g}{\longmapsto} \langle \boldsymbol{a},\boldsymbol{b}\rangle + r \ .$$

The first party, given input $x \in [d]$, represents x as a standard unit vector $\boldsymbol{e}_x \in \mathbb{F}_2^d$. The second party considers the function $f(\cdot,y)$ that has its input y hardwired, and represents it as a vector $\boldsymbol{f}_y \in \mathbb{F}_2^d$. In particular, the inner product of the two vectors yields the function output $\left\langle \boldsymbol{e}_x, \boldsymbol{f}_y \right\rangle = f(x,y)$. The advantage of this linear representation of the function is that its amenable to local secret sharing.

In what follows, let τ be a parameter (that will govern statistical security). The first party samples τ random shares $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_\tau \in \mathbb{F}_2^d$ such that $\boldsymbol{e}_x = \boldsymbol{x}_1 + \cdots + \boldsymbol{x}_\tau$. The second party samples τ shares $r_1,\ldots,r_\tau \in \mathbb{F}_2$ of the bit $0 = r_1 + \cdots + r_\tau$. They then run in parallel τ LAREs for the function $g(\boldsymbol{x}_i,(\boldsymbol{f}_y,r_i)) = \left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + r_i$ to compute:

$$\left\langle \boldsymbol{x}_1, \boldsymbol{f}_y \right\rangle + r_1, \ldots, \left\langle \boldsymbol{x}_\tau, \boldsymbol{f}_y \right\rangle + r_\tau \ .$$

Decoding is then straightforward - by adding all the results the decoder obtains

$$\left\langle \sum_i^d \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + \sum_i^d r_i = \left\langle \boldsymbol{e}_x, \boldsymbol{f}_y \right\rangle + 0 = f(x,y) \ .$$

To see why this is secure, observe that except with probability $2^{-\Omega(\tau)}$ over the choice of random $r_1,\ldots,r_{\tau-1}$, some of the outputs $\left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + r_i$ equal 0, in which case nothing leaks (but the fact that the output is 0). At the same time, whenever the output is 1, only $\boldsymbol{x}_i$ leaks, and since this happens strictly less than τ times, these shares are uniformly random and independent of the actual input x.

LARE by Reduction to ARE for OR. The construction of LARE for general functions (in particular, the function g used above) starts from the basic fact that the two-party OR function has a simple (non-leaky) ARE over any group $(\mathbb{G}, +)$ [HIKR23]. Specifically, each party encodes 0 as the identity element $0_{\mathbb{G}}$ and encodes 1 as a random element $r \in \mathbb{G}$. The decoder maps $0_{\mathbb{G}}$ to 0 and any other element $g \in \mathbb{G}^{\times}$ to 1. It is easy to see that this yields a perfectly-secure ARE for OR, with correctness error $1/|\mathbb{G}|$.

Then we construct a LARE for any finite function $f(x, y)$ by a reduction to bitwise OR of binary strings. The construction is inspired by the construction of ARE for general functions from ARE for Equality in [HIKR23]. Specifically, the first party considers the vector $\mathbf{1} - \boldsymbol{e}_x$, which is 1 on all coordinates $i \neq x$ and 0 on x. The second party considers the vector $\mathbf{1} - \boldsymbol{f}_y$, which is 1 on all coordinates i such that $f(i, y) = 0$ and 0 on all coordinates such that $f(i, y) = 1$. Observe that if $f(x, y) = 0$ then the bitwise OR of $\mathbf{1} - \boldsymbol{e}_x$ and $\mathbf{1} - \boldsymbol{f}_y$ is the all ones vector $\mathbf{1}$, whereas if $f(x, y) = 1$, this bitwise OR is the vector $\mathbf{1} - \boldsymbol{e}_x$, which only leaks x.

Efficient ARE Constructions. As shown in [HIKR23], to obtain efficient constructions of ARE, it suffices to construct ARE for the finite bit OT function $(b_0, b_1), c \longmapsto b_c$. Then one can use DRE to get an ARE for any function.[4] Specifically, recall that in a DRE, given a random secret key r, each input bit x_i is individually mapped into an encoding $\widehat{x}_i$. The evaluator given the encoded inputs bits $\widehat{x}_1, \ldots, \widehat{x}_k$ can decode the value of the function $f(x_1, \ldots, x_k)$, and learns nothing beyond. To turn such a DRE to an ARE given ARE for OT, we can have one party sample the randomness r and execute (string) OT ARE with every other party i, to allow the ARE evaluator to obtain $\widehat{x}_i$.

Using the above approach we can directly get an efficient ARE for all functions by choosing an (efficient) group $\mathbb{G}$ of size $\approx 2^{\tau}$ and using the ARE described above. This already yields some version of Theorem 1.2, but not an optimal one. In particular, running t instances of this ARE OT would require communication $t \cdot \mathrm{poly}(\tau)$.

To get a *constant rate* version of the above we use amplification techniques from [HIKN08, IKOS09]. Whereas the polynomial-rate version of the theorem relies on quite elementary tools, the mentioned amplification techniques require more advanced machinery such as algebraic-geometric codes (or more generally appropriate multiplication codes). See Section 5.4 for more details.

1.3 More Related Work

The abstract study of randomized encodings was initiated by Ishai and Kushilevitz [IK00], and has since led to a substantial body of work (see surveys [Ish13, App17]). AREs can be understood as a restricted form of multi-party randomized encodings (MPREs) [ABT21], where the emphasis is on simplifying the global encoding function that aggregates local encodings. While general MPREs can tolerate corruptions of some parties and still maintain security, AREs only provide security guarantees against an external observer (e.g., an untrusted server), and not against internal corruption.

To address this gap, Halevi et al. [HIKR23] introduced the notion of *robust* ARE which extends the basic ARE model to handle corrupted parties (aka *insiders*). However, unlike MPRE where only the output is leaked, in robust ARE a set C of corrupted parties may (inherently) learn the *residual function* $f(x_H, \cdot)$ where the honest parties' H inputs are fixed. As shown in [HIKR23], such robust AREs are in fact equivalent to obfuscation. They provide a heuristic construction of simulation-based robust AREs assuming ideal

[4] In [HIKR23] this is actually described with garbled circuits [Yao86] as a special case of DRE, typically referring to the computational setting.

obfuscation. Follow-up work [BF24] presents a construction under indistinguishability obfuscation and DDH/LWE.

Hiwatashi [Hiw25] investigates limitations of information-theoretic AREs. Extending techniques from [HIKR23], they characterize which Boolean functions admit AREs where security and correctness are measured in the ℓ_2-norm (rather than ℓ_1).

Our notion of Leaky ARE (LARE) is reminiscent of *semi-private randomized encodings* (SPRE) [AIK18]. SPREs are (single-party) REs that hide the input x when $f(x) = 0$ and leak x when $f(x) = 1$. Formally, the two notions are incomparable: LARE are two-party REs, which leak x but hide y when $f(x, y) = 1$ (and leak nothing when $f(x, y) = 0$). In [AR16] a similar notion called *One-Sided Private Simultaneous Messages* (1PSM) is considered and shown to imply PSM (with related communication complexity). This notion can be seen as a two-party RE, with important differences from LARE: first and foremost, as a variant of PSM it requires correlated secret randomness, and second when $f(x, y) = 1$, both x, y are leaked (rather than just x).

As for MPC in the shuffle model, whereas the default setting (including the one in this paper) is *non-interactive*, Beimel et al. showed that with interaction (even two rounds) the shuffle model regains the full power of MPC protocols [BHNS20]. This follows from the fact that shuffling gives rise to point-to-point secure channels [IKOS06], over which one could use MPC protocols (e.g. [ABT21]). Furthermore, we observe that our construction achieves setupless non-interactive MPC assuming access to a trusted secure aggregation (SecAgg) functionality, rather than a shuffling functionality as in [KLS21, CCKS22].

1.4 Open Questions

Our work leaves several interesting questions for future research.

- **Perfect correctness.** Does every f admit a *perfectly correct* and statistically secure ARE? For computationally secure ARE, perfect correctness is possible [HIKR23, BEG25]. For ARE with statistical security, we are only able to achieve a relaxed "Las Vegas" notion of correctness, where the evaluator is never wrong but may declare failure with a small probability (see Appendix A). In November 2026, we have added in Appendix B a solution to this problem based on interaction with ChatGPT.

- **Robust ARE.** Can the information-theoretic feasibility of ARE be extended to the *robust* notion of ARE from [HIKR23] in the case of *finite* functions? (Recall that an efficient construction for general functions implies obfuscation.) A positive answer, even for finite 3-party functions, would settle the main open question about multi-party randomized encodings [ABT21]. A positive answer for all finite functions would settle the main open question about best-possible information-theoretic MPC [HIKR18].

- **ARE vs. PSM.** The tight relation between ARE and DRE shows that, in the context of private simultaneous messages (PSM) protocols [FKN94] with one-bit inputs, common secret randomness is not much more powerful than an addition oracle. Is the same true for longer inputs? For example, does any 2-party PSM for $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ imply an ARE for f with similar cost?

2 Preliminaries

For discrete probability distributions X and Y over support set A, their *statistical distance* (or total variation distance) is $\Delta(X,Y) = \frac{1}{2}\sum_{a\in A}|\Pr[X=a]-\Pr[Y=a]|$. Throughout, we assume for simplicity all groups $\mathbb{G}$ are *efficient*, namely, they have representation of size $c\log|\mathbb{G}|$ and operations in time $|\mathbb{G}|^c$ (for some universal constant c). We denote by $X \approx_\delta Y$ the fact that $\Delta(X,Y) \le \delta$. For an Abelian group $(\mathbb{G},+)$ and $x \in \mathbb{G}$ we say that $x_1,\ldots,x_k$ are (secret) shares of x if they are random in $\mathbb{G}$ conditioned on their sum being x (for strings $x \in \{0,1\}^n$ we consider by default the group $(\mathbb{Z}_2^n,\oplus)$). For a positive integer n, we use $[n]$ to denote the set $\{1,\ldots,n\}$. For a set S, $x \leftarrow S$ denotes uniformly sampling from S. We next review the main cryptographic primitives we use in this work.

2.1 Additive Randomized Encodings

Definition 2.1 (Additive Randomized Encoding [HIKR23])**.** Let $f : (\{0,1\}^n)^k \to \{0,1\}^m$ be a k-party function. An ARE scheme for f over an Abelian group $(\mathbb{G},+)$ consists of functions $\Pi = (\mathsf{Enc},\mathsf{Dec})$ with the following syntax:

- $\widehat{x_i} \leftarrow \mathsf{Enc}(x_i,i)$ is a randomized encoding function that given an input $x_i \in \{0,1\}^n$ and an index $i \in [k]$ outputs an encoding $\widehat{x_i} \in \mathbb{G}$.
- $w \leftarrow \mathsf{Dec}(\widehat{w})$ is a deterministic decoding algorithm that given $\widehat{w} \in \mathbb{G}$ outputs a value w.

We require the following properties:

- Correctness: Π is ε-correct if for all $x_1\ldots,x_k \in \{0,1\}^n$:

$$\Pr\left[\mathsf{Dec}\left(\sum_{i=1}^{k}\mathsf{Enc}(x_i,i)\right) = f(x_1,\ldots,x_k)\right] \ge 1-\varepsilon \ .$$

- Security: Π is δ-secure if there exists a simulator function Sim such that for any $x_1,\ldots,x_k \in \{0,1\}^n$:

$$\sum_{i=1}^{k}\mathsf{Enc}(x_i,i) \approx_\delta \mathsf{Sim}(f(x_1,\ldots,x_k)) \ .$$

ARE Families. Depending on the context, we may view an ARE either as a finite object (as defined above) or as an infinite family of such objects, parameterized by τ. In the latter case, we assume that all computational costs, including group operations, are polynomially bounded in τ, and also use τ as a statistical or computational security parameter.

3 Leaky ARE

In this section we define and construct a relaxed notion of two-party ARE called *leaky ARE*. This notion allows completely leaking the input of one of the parties when the corresponding function output is 1.

Definition 3.1 (Leaky ARE)**.** A Leaky ARE (LARE) scheme $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ for $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ has the same syntax and correctness requirement as a (plain) two-party ARE. The security requirement is replaced with the following leaky security requirement:

- Leaky Security: Π is δ-leaky-secure if there exists a simulator function Sim such that for any $x, y \in \{0,1\}^n$:

$$\mathsf{Enc}(x,1) + \mathsf{Enc}(y,2) \approx_\delta \mathsf{Sim}(f(x,y), \ell(x,y))$$

$$\text{where } \ell(x,y) = \begin{cases} x \text{ if } f(x,y) = 1 \\ \perp \text{ if } f(x,y) = 0 \end{cases}$$

Remark 3.2 (Relation to One-Sided ARE)**.** The above notion of leaky ARE is a strengthening of the notion of one-sided ARE defined in [BEG25] where $\ell(x,y) \equiv x$ (regardless of $f(x,y)$).

3.1 LARE for any Finite Function

In this section, we construct a LARE for any finite function. Throughout this section, we treat the complexity of the function as constant, and address efficiency, correctness, and security in terms of the underlying group size.

Theorem 3.3. *For any (efficient) group $\mathbb{G}$, any Boolean function $f : [d] \times [d'] \to \{0,1\}$ has a $O(|\mathbb{G}|^{-1})$-correct, 0-leaky-secure ARE over $\mathbb{G}^{O(1)}$. Encoding/decoding efficiency is* $\mathrm{polylog}(|\mathbb{G}|)$.

Proof. The construction is presented in Figure 1.

The efficiency of encoding/decoding follows readily. We prove that the scheme satisfies the two claimed correctness and security properties.

Correctness:
Recall that the sum encodings $\widehat{w}$ are computed additively as $\widehat{x} + \widehat{y}$.

- When $f(x,y) = 1$, $\widehat{w}_x = 0$ and decoding outputs 1.
- When $f(x,y) = 0$, for all $i \in [d]$, $\widehat{w}_j$ is uniformly random in $|\mathbb{G}|$, and hence all are non-zero, except with probability at most $d/|\mathbb{G}|$, and decoding outputs 0.

Leaky Security:

- When $f(x,y) = 1$, $\mathsf{Sim}(1,x)$ randomly samples $\widehat{w}_i \leftarrow \mathbb{G}$ for all $i \neq x$ and sets $\widehat{w}_x = 0$. This perfectly emulates the encoding $\sum_{i:f(i,y)=0} (r_i + s_i) \cdot \boldsymbol{e}_i + \sum_{i \neq x : f(i,y)=1} r_i \cdot \boldsymbol{e}_i$.
- When $f(x,y) = 0$, $\mathsf{Sim}(0,\perp)$ randomly samples $\widehat{w}_j \leftarrow \mathbb{G}$ for all $j \in [d]$. This perfectly emulates the encoding $s_x \boldsymbol{e}_x + \sum_{i \neq x : f(i,y)=0} (r_i + s_i) \cdot \boldsymbol{e}_i + \sum_{i:f(i,y)=1} r_i \cdot \boldsymbol{e}_i$.

□

LARE_f

$\underline{\mathrm{Enc}(x, 1)}$:

- Sample $r_1, \ldots, r_d \leftarrow \mathbb{G}$.
- Set $\widehat{x} = \sum_{i \in [d] \setminus \{x\}} r_i \cdot \boldsymbol{e}_i \in \mathbb{G}^d$, where $\boldsymbol{e}_i \in \{0,1\}^d$ is the i-th standard basis vector and $r_i \boldsymbol{e}_i$ is interpreted as a vector in $\mathbb{G}^d$ in the natural way.

$\underline{\mathrm{Enc}(y, 2)}$:

- Sample $s_1, \ldots, s_d \leftarrow \mathbb{G}$.
- Set $\widehat{y} = \sum_{i : f(i,y)=0} s_i \cdot \boldsymbol{e}_i$.

$\underline{\mathrm{Dec}(\widehat{w})}$:

- Parse the (sum) encoding $\widehat{w}$ as a vector $(\widehat{w}_1, \ldots, \widehat{w}_d)$.
- If for all i, $\widehat{w}_i \neq 0$, output 0, else output 1.

Figure 1: LARE for f

Remark 3.4 (ARE for OR)**.** The construction can be interpreted as a reduction from ARE for the OR to a LARE for general functions. Here the party holding $x \in [d]$ translates its input to $\widehat{x} = \sum_{i \in [d] \setminus \{x\}} \boldsymbol{e}_i \in \{0,1\}^d$ and the party holding y translates its input to $\widehat{y} = \sum_{i : f(i,y)=0} \boldsymbol{e}_i \in \{0,1\}^d$, the parties then apply bitwise OR to the two strings using the OR for ARE from [HIKR23], which encodes $b \in \{0,1\} \mapsto b \cdot r \in \mathbb{G}$ for a random group element $r \in \mathbb{G}$.

Remark 3.5 (Correctness)**.** The above construction has a $O(|\mathbb{G}|^{-1})$ correctness error, all coming from the correctness error of the OR for ARE. This will lead to (and will be the only source of) an error in our final construction of ARE. In Appendix A, we show how to obtain *Las Vegas correctness*; namely, with probability $1 - O(|\mathbb{G}|^{-1})$ decoding is correct, and otherwise outputs $\perp$. Subsequently, the work of Ishai, Kushilevitz, and Yerushalmi [IKY26] explored the limits of perfect correctness for ARE. In November 2026, we added Appendix B showing a perfectly correct ARE based on interaction with ChatGPT Astra.

4 ARE from LARE

In this section, we show how to transform LARE to ARE for general functions. We start from the case of finite functions (Section 4.1) and then address the case of general functions and asymptotic efficiency (Section 5).

4.1 Finite Functions

Throughout this subsection, we treat the complexity of the function as constant, and address efficiency, correctness, and security in terms of the underlying group size and an additional parameter τ.

Theorem 4.1. *For any (efficient) group $\mathbb{G}$ and parameter $\tau \in \mathbb{N}$, any Boolean function $f : [d] \times [d] \to \{0,1\}$ has a $O(\tau/|\mathbb{G}|)$-correct, $2^{-\tau}$-secure ARE over $\mathbb{G}^{O(\tau)}$. Encoding/decoding efficiency is $\tau \cdot \mathrm{polylog}(|\mathbb{G}|)$.*

In Appendix B we reformulate a perfectly correct version of the theorem.

The theorem (and its perfectly correct version in Appendix B) are derived from the following lemma together with Theorem 3.3.

Lemma 4.2. *Let $\tau \in \mathbb{N}$ be a parameter and let $f : [d] \times [d] \to \{0,1\}$ be a Boolean function. Then any ε-correct, δ-leaky-secure LARE, for inner product in $\mathbb{F}_2^{d+1}$, over a group $\mathbb{G}$, can be converted into a $\varepsilon\tau$-correct $(\delta\tau + 2^{-\tau+1})$-secure ARE for f, over $\mathbb{G}^\tau$. The ARE for f has the same encoding/decoding efficiency as τ applications of the underlying LARE.*

Proof of Lemma 4.2.

Construction 4.3 (Statistical ARE for General Functions)**.**

- Consider the function $g : \mathbb{F}_2^d \times (\mathbb{F}_2^d \times \mathbb{F}_2) \to \mathbb{F}_2$:

$$g(\boldsymbol{a}, (\boldsymbol{b}, r)) = \langle \boldsymbol{a}, \boldsymbol{b} \rangle + r \quad .$$

- Let LARE_g be an ε-correct, δ-leaky-secure LARE scheme (definition 3.1) for g.[5]

Proposition 4.4. *The scheme ARE_f is $\tau\varepsilon$-correct.*

Proof. From the ε-correctness of LARE_g, except with probability at most $\tau\varepsilon$, it holds that for every $i \in [\tau]$

$$\mathsf{LARE}_g.\mathsf{Dec}(\widehat{w}_i) = g(\boldsymbol{x}_i, (\boldsymbol{f}_y, r_i)) = \left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + r_i \quad .$$

In this case,

$$\mathsf{ARE}_f.\mathsf{Dec}(\widehat{w}) = \sum_{i \in [\tau]} (\left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + r_i) = \left\langle \sum_{i \in [\tau]} \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + \sum_{i \in [\tau]} r_i = \left\langle \boldsymbol{e}_x, \boldsymbol{f}_y \right\rangle + 0 = \boldsymbol{f}_y[x] = f(x,y) \quad .$$

□

Proposition 4.5. *The scheme ARE_f is $(\delta\tau + 2^{-\tau+1})$- secure.*

Proof. We start by defining the simulator $\mathsf{ARE}_f.\mathsf{Sim}(f(x,y))$:

[5] Note that a LARE for g can be emulated from LARE for inner product over $\mathbb{F}_2^{d+1}$ (as stated in Lemma 4.2), but using g will be easier to parse.

ARE_f

$\underline{\mathsf{Enc}(x, 1)}$:

- Let $\boldsymbol{e}_x \in \{0,1\}^d$ be the standard unit vector corresponding to x.
- Sample additive secret shares $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\tau \in \{0,1\}^d$ of $\boldsymbol{e}_x$.
- Sample $\widehat{x}_i = \mathsf{LARE}_g.\mathsf{Enc}(\boldsymbol{x}_i, 1)$ for every $i \in [\tau]$.
- Output $(\widehat{x}_1, \ldots \widehat{x}_\tau)$.

$\underline{\mathsf{Enc}(y, 2)}$:

- Let $\boldsymbol{f}_y \in \{0,1\}^d$ be the vector corresponding to $f(\cdot, y)$; namely $\boldsymbol{f}_y[x] = f(x, y)$.
- Sample additive secret shares $r_1, \ldots, r_\tau$ of 0.
- Sample $\widehat{z}_i = \mathsf{LARE}_g.\mathsf{Enc}(z_i = (\boldsymbol{f}_y, r_i), 2)$ for every $i \in [\tau]$.
- Output $(\widehat{z}_1, \ldots, \widehat{z}_\tau)$.

$\underline{\mathsf{Dec}(1^\lambda, \widehat{w}) :}$

- Parse the (sum) encoding $\widehat{w}$ as a vector $(\widehat{w}_1, \ldots, \widehat{w}_\tau)$.
- Output $\sum_{i=1}^{\tau} \mathsf{LARE}_g.\mathsf{Dec}(\widehat{w}_i) \mod 2$.

Figure 2: ARE for f

- Sample additive secret shares $v_1, \ldots, v_\tau$ of $f(x, y)$.
- If $\forall i \in [\tau], v_i = 1$, output $\perp$.
- For i such that $v_i = 0$, sample $\widehat{w}_i \leftarrow \mathsf{LARE}_g.\mathsf{Sim}(0, \perp)$.
- For i such that $v_i = 1$, sample $\boldsymbol{x}_i \in \{0,1\}^d$ uniformly at random and sample $\widehat{w}_i \leftarrow \mathsf{LARE}_g.\mathsf{Sim}(1, \boldsymbol{x}_i)$.
- Output $\widehat{w}_1, \ldots, \widehat{w}_\tau$.

To bound the statistical distance between the real and simulated (sum) encodings, we consider several hybrids.

Hybrid 0 **(real encodings).** Here the parties compute their encodings $\widehat{w}_i = \widehat{x}_i + \widehat{z}_i$ according to Construction 4.3.

Hybrid 1. Here instead of using the encoding functions, we simulate the encodings. Specifically, if $g(\boldsymbol{x}_i, (\boldsymbol{f}_y, r_i)) = 1$, we sample $\widehat{w}_i \leftarrow \mathsf{LARE.Sim}(1, \boldsymbol{x}_i)$, and if $g(\boldsymbol{x}_i, (\boldsymbol{f}_y, r_i)) = 0$, $\widehat{w}_i \leftarrow \mathsf{LARE.Sim}(0, \perp)$.

By the δ-leaky-security of LARE_g, the resulting encodings are $\tau\delta$-statistically-close to those in the previous hybrid.

Hybrid 2. Here, in case it holds that for all $i \in [\tau]$, $g(\boldsymbol{x}_i, (\boldsymbol{f}_y, r_i)) = \left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle + r_i = 1$, we output $\perp$.

Since $r_1, \ldots, r_{\tau-1}$ are uniformly random and independent of the values $\left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle$, this event occurs with probability at most $2^{-\tau+1}$, and hence the encodings in this hybrid are $2^{-\tau+1}$-statistically-close to those in the previous hybrid.

Hybrid 3 (simulated encodings). Here the encodings are distributed according to the above described simulator $\mathsf{ARE}_f.\mathsf{Sim}$.

The encodings in this hybrid are identically distributed to those in the previous hybrid. Indeed, the values $\left\langle \boldsymbol{x}_1, \boldsymbol{f}_y \right\rangle + r_1, \ldots, \left\langle \boldsymbol{x}_\tau, \boldsymbol{f}_y \right\rangle + r_\tau$ are uniformly random conditioned on their sum being $\sum_{i \in \tau} \left\langle \boldsymbol{x}_i, \boldsymbol{f}_y \right\rangle = f(x, y)$ just like the simulated secret shares $v_1, \ldots, v_\tau$. Furthermore, whenever $\left\langle \boldsymbol{x}_i, \boldsymbol{f}_i \right\rangle + r_i = 0$ for some i, the resulting encodings are independent of $\boldsymbol{x}_i$, and depend (at most) on $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_\tau$, which are uniformly random and independent as in the simulation. Finally if $\left\langle \boldsymbol{x}_i, \boldsymbol{f}_i \right\rangle + r_i = 1$ for all i, both distributions are $\perp$. □

□

5 ARE from DRE

In this section, we draw a tight connection between ARE and decomposable randomized encodings (DRE). First, we recall the definition of DREs below.

Definition 5.1 (Decomposable Randomized Encodings)**.** Let $f : (\{0,1\}^n)^k \to \{0,1\}^m$ be a k-party function. A DRE scheme for f consists of functions $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ with the following syntax:

- $\widehat{x}_i \leftarrow \mathsf{Enc}(r, x_i, i)$ is a randomized encoding function that given randomness $r \in \{0,1\}^\lambda$, an input $x_i \in \{0,1\}^n$, and an index $i \in [k]$ outputs an encoding $\widehat{x}_i \in \mathbb{G}$.
- $w \leftarrow \mathsf{Dec}(\widehat{w})$ is a deterministic decoding algorithm that given $\widehat{w} \in \mathbb{G}$ outputs a value w.

We require the following properties:

- Correctness: Π is ε-correct if for all $x_1 \ldots, x_k \in \{0,1\}^n$ and for all $r \in \{0,1\}^\lambda$:

$$\Pr\left[\mathsf{Dec}\left((\mathsf{Enc}(r, x_i, i))_{i=1}^k\right) = f(x_1, \ldots, x_k)\right] \geq 1 - \varepsilon \ .$$

- Security: Π is δ-secure if there exists a simulator function Sim such that for any $x_1, \ldots, x_k \in \{0,1\}^n$:

$$\left((\mathsf{Enc}(r, x_i, i))_{i=1}^k\right) \approx_\delta \mathsf{Sim}(f(x_1, \ldots, x_k)) \ .$$

We further say that security is *computational/perfect* if instead of statistical indistinguishability we have computational/perfect indistinguishability.

The *size* of the DRE is the bit-length of the encoding output.

Next, we discuss how our ARE for finite two-party functions can be leveraged to obtain ARE for general multi-party functions f with comparable communication complexity to the corresponding DRE for f. We start by discussing and defining the notion of *size* for multi-party ARE.

5.1 ARE Size

A standard cost measure of randomized encodings is their *size*, typically defined as the bit-length of the encoded output. In the context of ARE, this is just $\log|\mathbb{G}|$, the bit-length of a single group element. While this measure is good enough for the two-party case (which already captures the main contributions of this work), the multi-party case calls for a more refined measure. In particular, note that the $\log|\mathbb{G}|$ measure can be smaller than the communication complexity of f. To better capture the cost of ARE applications and the relation between ARE and the related notion of DRE, we define ARE size as $\sum_{i=1}^{k}|\widehat{x}_i|$, where the size of each encoded input takes into account the size of an associated subgroup.

Definition 5.2 (ARE Size)**.** We say that a k-party ARE over a group $\mathbb{G}$ has *size* (at most) c if:

1. $\mathbb{G} = \mathbb{G}_1 \times \cdots \times \mathbb{G}_t$ for some (explicit and efficient) groups $\mathbb{G}_i$.
2. For any party $i \in [k]$, there is a set $S_i \subseteq [t]$ such that the party's encodings $\widehat{x}_i$ are supported on $\mathbb{G}_{S_i} \times \{0\}_{\overline{S_i}}$. Namely, each party encodes into a subset of coordinates (when $\mathbb{G}$ is viewed as a product).
3. $\sum_{i\in[k]} \log|\mathbb{G}_{S_i}| := \sum_{i\in[k],j\in S_i} \log|\mathbb{G}_j| \leq c$.

We note that the above fine-grained notion of size will mainly be of interest when considering k-party ARE for growing k. In the two-party case for instance, we will focus on the (representation) size of the ARE group $\log|\mathbb{G}|$ which is an upper bound on the size of the ARE up to a factor of 2. In fact, our main construction of k-party ARE for $k > 2$ simulates $k-1$ parallel instances of 2-party ARE, where the above notion of size adds up their sizes.

The above notion of size aims to better capture the actual communication cost of ARE in applications. For example, when using ARE to obtain an MPC protocol in the client-server model, each client only needs to secret-share its input over the corresponding subgroup. Consequently, the ARE size is the total size of all messages received by each server. A qualitatively similar relation between ARE size and communication complexity holds also for the shuffle model. See [GMPV20, BBGN20, GIK+24] for the best known bounds on the communication cost of secure addition in the shuffle model.

ARE vs. DRE size. Recall that a DRE for a Boolean function $f : \{0,1\}^k \to D$ is defined similarly to an ARE, except that the encoded output is of the form $g(x,r) = (g_1(x_1,r),\ldots,g_k(x_k,r))$. The size of the DRE is the bit-length of the output.[6] The above notion of ARE size ensures that if f has an ARE of size c, then it also has a DRE of size c. We show that the converse also holds up to a constant factor; see Section 5.4 for discussion.

[6] See [BHI+20] for known upper and lower bounds on DRE complexity.

5.2 From DRE and OT-ARE to ARE

Our starting point is the work of [HIKR23], showing that efficient multi-party ARE for general functions is implied by any ARE for the finite bit oblivious transfer (OT) function:

$$(b_0, b_1), c \longmapsto b_c \ .$$

This is done using Yao's garbled circuit construction [Yao86]. Here we restate this reduction to OT more generally, in the language of DRE.

Theorem 5.3 (From DRE and OT to ARE [HIKR23])**.** *Let $n(\tau), k(\tau), m(\tau)$ be polynomially bounded functions and let $f : (\{0,1\}^n)^k \to \{0,1\}^m$ be a k-party function (family).*

- *Assume that f, viewed as a function over nk bits, has a DRE of size $c' = c'(\tau)$ per party and statistical correctness and security errors at most $\varepsilon = \varepsilon(\tau)$.*
- *Assume an ARE for $c'n$ instances of bit OT of size $c = c(\tau)$ and statistical correctness and security errors at most $\delta = \delta(\tau)$.*

Then the k-party f (over n-bit inputs) has an ARE of size $O(ck)$ and statistical correctness and security errors at most $\varepsilon + \delta$. If the DRE and OT ARE parties run in (uniform) time $\mathrm{poly}(\tau)$ so do the final ARE parties.

An analogous theorem holds if the DRE/OT is computationally secure (in which case τ is viewed as a computational security parameter). Here the final ARE is also computationally secure.

We next discuss two ways of instantiating OT ARE based on Theorem 4.1 and resulting corollaries when combined with DREs from the literature. The first is a direct one yielding polynomial rate OT, and the second uses more advanced tools to obtain constant rate OT.

We consider throughout function (families) $f : (\{0,1\}^n)^k \to \{0,1\}^m$ parameterized by τ where $n = n(\tau), m = m(\tau), k = k(\tau)$ are polynomially bounded functions.

5.3 A Direct OT Instantiation

Applying theorem 4.1 with a group $\mathbb{G}$ of size $\approx 2^{O(\tau)}$, we obtain a $\mathrm{poly}(\tau)$-efficient ARE for OT with correctness/security errors $2^{-\tau}$. We then get the following corollaries using DREs from the literature.

- **Perfect DRE [FKN94, IK97]:** Any function $f \in NL/\mathrm{poly}$, has a $\mathrm{poly}(\tau)$-efficient ARE with statistical correctness and security errors $2^{-\tau}$. More generally, any mod-p *counting branching program* can be compiled in polynomial time into such an ARE.
- **Computational Garbling [Yao86]:** Assuming one-way functions, any function $f \in P/\mathrm{poly}$, has a $\mathrm{poly}(\tau)$-efficient ARE with statistical correctness error $2^{-\tau}$ and computational security against $\mathrm{poly}(\tau)$-distinguishers. More generally, any Boolean circuit of size S can be efficiently transformed into such an ARE of size $\mathrm{poly}(\tau) \cdot S$.

- **Best Known DRE for General Functions [BKN18]:**[7] Any function f has an ARE of size $2^{nk/2} \cdot \mathrm{poly}(\tau)$, and statistical correctness and security errors $2^{-\tau}$.

The ARE OT instantiated in the above approach generally has polynomial size blowup. That is, t instances of bit OT require $t \cdot \mathrm{poly}(\tau)$ communication for soundness $2^{-\tau}$. We next discuss a more efficient approach, based on more advanced tools, which achieves constant rate, and its implications.

5.4 Constant-Rate ARE for Batch-OT

In a constant-rate ARE for batch-OT, one party holds a batch of t pairs $(s_{1,0}, s_{1,1}), \ldots, (s_{t,0}, s_{t,1})$, while the other party holds t choice bits $c_1, \ldots, c_t$. Our goal is for the evaluator to learn $s_{c_1}, \ldots, s_{c_t}$ and we aim to have encoding size $O(t)$ with soundness and correctness errors $2^{-\Omega(t)}$. Note that Theorem 4.1 gives us a single OT ARE of constant encoding size, but with (arbitrarily) small constant correctness and security errors ε. Our goal is therefore reduced to emulating t OT instances with negligible errors, given access to $O(t)$ ε-erroneous instances.

An analogous amplification problem is solved in the context of MPC. Indeed, the work of [IKOS09] leverages algebraic-geometric codes to achieve a similar goal of obtaining a protocol realizing t instances of OT with negligible errors invoking $O(t)$ instances of a protocol with constant communication and errors. We cannot quite invoke their amplification theorem as is. First, our adversarial model is different, we wish to defend against an external evaluator, whereas in the MPC context we defend against a corrupted party (holding either the pairs $(s_{i,0}, s_{i,1})$ or the choices c_i. Second, their solution involves interaction. Nevertheless, a variant of their amplification technique also works in the context of ARE. We next sketch the construction and the argument behind its correctness and security.

Amplifiers from Combiners. The construction follows a common paradigm for amplification in the statistical setting (c.f. [MPR07, IKO+11, LM20, ABG26]): to construct an amplifier, it is sufficient to construct an appropriate *combiner*. Restricting attention to our setting, we aim to construct an (n, α, β)-combiner for batch-OT ARE for some constants $\alpha, \beta < 1$. Such a combiner is given as input n single-OT ARE schemes and combines them into a single ARE for a batch of βn OT instances. The guarantee is that as long as at least $(1-\alpha)n$ schemes (out of the n schemes) are perfectly secure and at least $(1-\alpha)n$ are perfectly correct, the resulting scheme is perfectly (or almost perfectly) correct and secure. Once such a combiner is achieved, it can be shown that for some small enough constant $\varepsilon = \varepsilon(\alpha)$ if we instantiate each of the n schemes with an ε-erroneous single-OT ARE, we get a $2^{-\Omega(n)}$ secure and correct ARE for βn instances.

A Combiner Based on Multiplication Codes. In the setting of MPC analogous constant-rate batch-OT combiners are known starting from [HIKN08]. In our context of ARE, we sketch a relatively simple combiner that is a variant of an OT combiner from [IKOS09] and is based on *multiplication codes*. A multiplication code $C \subseteq \mathbb{F}^n$ is a linear code with the property that the pointwise product $c_1 \odot c_2$ of two codewords $c_1, c_2 \in C$ is a codeword in a related code $C^\star \subseteq \mathbb{F}^n$. We require several properties from the code:

- Both $C, C^\star$ are of constant rate: for some constant size $\mathbb{F}$ and constant $\rho < 1$, we can encode words in $\mathbb{F}^{\rho n}$ in either $C, C^\star \subseteq \mathbb{F}^n$.

[7] The result of [BKN18] is stated in the language of multi-party private simultaneousness message protocols, which are equivalent to DRE when restricting attention to single-bit inputs.

- Both $C, C^\star$ have constant dual distance: for some $\delta^\perp < 1$, any word in their dual codes $C^\perp, (C^\star)^\perp$ has hamming weight at least $\delta^\perp n$.
- $C^\star$ has constant distance and efficient decoding: for some $\delta < 1$, inducing at most δn errors to a word in $C^\star$, we can efficiently recover the codeword.

Such efficiently constructible codes follow from the algebraic geometric codes of [GS96], and starting from [CC06] were used in various contexts in cryptography, and in particular for multiplicative secret sharing. Specifically, given multiplication codes as above there exist constants $\beta < \delta, \gamma < 1$ such that we can randomly encode words $\boldsymbol{x} \in \mathbb{F}^{\beta n}$ in the last $(1-\beta)n$ coordinates of codewords in $C, C^\star$, with the guarantee that any γn entries reveal no information about $\boldsymbol{x}$. We will denote the corresponding secret shares as $SS_C(\boldsymbol{x}), SS_{C^\star}(\boldsymbol{x}) \in \mathbb{F}^{(1-\beta)n}$.

Oblivious Linear Evaluation. To take advantage of the structure of multiplication codes, we consider a combiner for an algebraic generalization of OT known as *oblivious linear evaluation* (OLE), which is the two-party function:

$$(u, v), c \longmapsto uc + v \ , \text{ for } u, v, c \in \mathbb{F}.$$

Indeed, the OLE functionality in particular allows implementing the bit-OT functionality by thinking of c as the choice bit, $v = s_0$, and $u = s_1 - s_0$.

Our combiner will obtain an ARE for βn OLE instances with perfect security and correctness given $(1-\beta)n$ instances, such that at least $1-\gamma$ (fraction) are perfectly secure, and at least $1-\delta+\beta$ are perfectly correct. Then for $\alpha < \gamma, (\delta - \beta)$, we get the desired (n, α, β)-combiner. The combiner works as follows:

- The first party, with inputs $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}^{\beta n}$, samples shares $\widehat{\boldsymbol{u}} \leftarrow SS_C(\boldsymbol{u}), \widehat{\boldsymbol{v}} \leftarrow SS_{C^\star}(\boldsymbol{v})$.
- The second party, with input $\boldsymbol{c} \in \mathbb{F}^{\beta n}$, samples shares $\widehat{\boldsymbol{c}} \leftarrow SS_C(\boldsymbol{c})$
- The parties then run $(1-\beta)n$ OLE AREs for each coordinate $i \in [(1-\beta)n]$ with inputs $(\widehat{\boldsymbol{u}}_i, \widehat{\boldsymbol{v}}_i), \widehat{\boldsymbol{c}}_i$.
- The evaluator obtains $\widehat{\boldsymbol{z}} \in \mathbb{F}^{(1-\beta)n}$, treats it as a noisy codeword in $C^\star$, and decodes to obtain $\boldsymbol{u} \odot \boldsymbol{c} + \boldsymbol{v} \in \mathbb{F}^{\beta n}$.

The perfect correctness follows from the fact that all but $(\delta - \beta)n$ OLE ARE instances are perfectly correct, βn (first) code symbols were erased, whereas $C^\star$ can decode given δn errors. For security, it can be shown that $\widehat{\boldsymbol{u}} \odot \widehat{\boldsymbol{c}} + \widehat{\boldsymbol{v}}$ is distributed as a random secret sharing $SS_{C^\star}(\boldsymbol{u} \odot \boldsymbol{c} + \boldsymbol{v})$. Furthermore, since all but γn ARE instances are perfectly secure, from each $\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{c}}$ we leak at most γn symbols. Accordingly, we can perfectly simulate the evaluator view given only the result $\boldsymbol{u} \odot \boldsymbol{c} + \boldsymbol{v} \in \mathbb{F}^{\beta n}$.

Corollary 1: Constant-Rate ARE. Combining the above batch OT with any DRE we can obtain a constant-rate ARE for (many copies of) any finite function. More generally, any $f \in NC^0$ with m output bits has an ARE with total encoding size $O(m)$ and statistical correctness and security errors $2^{-\Omega(m)}$. This applies in particular to string-OT of length m, which can be used to implement Yao's garbled circuit construction in the ARE model with constant communication overhead. That is, every Boolean circuit of size S admits a computationally secure ARE of size $O(\tau \cdot S)$, making a black-box use of a pseudorandom generator with a τ-bit seed.

Corollary 2: Equivalence of ARE and DRE for Boolean Functions. Considering k-party functions with Boolean inputs $f : \{0,1\}^k \to \{0,1\}$, we obtain a tight connection between the size of ARE (per Definition 5.2) and that of DRE. Specifically, assume f has a DRE of (total) encoding size kc and security/correctness error δ. Then f has an ARE of size $O(k(c + \log \varepsilon^{-1}))$ with security/correctness error $\delta + \varepsilon \cdot 2^{-c}$.

This result is essentially optimal in the sense that any ARE of size s for f implies a DRE of the same size and with the same security. To see this, recall that in a size-s ARE over $\mathbb{G} = \mathbb{G}_1 \times \cdots \times \mathbb{G}_t$, the encodings of party i are supported on $\mathbb{G}_{S_i} \times \{0\}_{\overline{S_i}}$ for some $S_i \subseteq [t]$, and $s = \sum_{i \in [k]} \log |\mathbb{G}_{S_i}|$.

In the constructed DRE, the correlated randomness includes for any $j \in [t]$ a secret sharing $\{g_{j,i}\}_{i:j \in S_i} \subseteq \mathbb{G}_j$ of $0_{\mathbb{G}_j}$ among all i such that $j \in S_i$. Then to encode input x_i, we first compute the encoding $\widehat{x}_i \in \mathbb{G}_{S_i}$ and add to the coordinate of any j such that $j \in S_i$ the share $g_{j,i}$. Accordingly, the size of the DRE encoding of each party i is $\log |\mathbb{G}_{S_i}|$. For security, note that the DRE encodings can be perfectly simulated from the ARE sum encoding $\widehat{x}_1 + \cdots + \widehat{x}_k \in \mathbb{G}$, hence security follows from the ARE.

Acknowledgments

We thank Benny Applebaum for pointing out related notions of leaky randomized encodings.

Y. Ishai was supported in part by ISF grants 2774/20 and 3527/24, BSF grant 2022370, and ISF-NSFC grant 3127/23. This work is not associated with AWS. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039655. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

References

[AAP19] Navneet Agarwal, Sanat Anand, and Manoj Prabhakaran. Uncovering algebraic structures in the MPC landscape. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 381–406. Springer, 2019.

[ABG26] Benny Applebaum, Nir Bitansky, and Nathan Geier. Security amplification via robust indistinguishability combiners. In *Advances in Cryptology – CRYPTO 2026*, 2026.

[ABT21] Benny Applebaum, Zvika Brakerski, and Rotem Tsabary. Perfect secure computation in two rounds. *SIAM J. Comput.*, 50(1):68–97, 2021.

[AG26] Benny Applebaum and Nathan Geier. Optimal amplification via bias-resilient combiners. In Dana Dachman-Soled and Daniel Wichs, editors, *Theory of Cryptography – TCC 2026, Part II*, volume 17192 of *Lecture Notes in Computer Science*, pages 223–255. Springer, 2026.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[AIK18] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Minimizing locality of one-way functions via semi-private randomized encodings. *Journal of Cryptology*, 31(1):1–22, 2018.

[AIKP22] Benny Applebaum, Yuval Ishai, Or Karni, and Arpita Patra. Quadratic multiparty randomized encodings beyond honest majority and their applications. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 453–482. Springer, 2022.

[App17] Benny Applebaum. Garbled circuits as randomized encodings of functions: a primer. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 1–44. Springer International Publishing, 2017.

[AR16] Benny Applebaum and Pavel Raykov. From private simultaneous messages to zero-information arthur-merlin protocols and back. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography — 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II (Lecture Notes in Computer Science, vol. 9563)*, pages 65–82, 2016.

[BBGN19] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. *arXiv preprint arXiv:1903.02837*, 2019.

[BBGN20] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. Private summation in the multi-message shuffle model. In *CCS*, 2020.

[BCJM21] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2384–2403. SIAM, 2021.

[BEG25] Nir Bitansky, Saroja Erabelli, and Rachit Garg. Additive randomized encodings from public key encryption. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part IV*, Lecture Notes in Computer Science, pages 241–260. Springer, 2025.

[BEM+17] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP '17)*, pages 441–459, Shanghai, China, October 2017. Association for Computing Machinery (ACM).

[BF24] Nir Bitansky and Sapir Freizeit. Robust additive randomized encodings from IO and pseudo-non-linear codes. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VIII*, volume 14927 of *Lecture Notes in Computer Science*, pages 109–135. Springer, 2024.

[BGI+14] Amos Beimel, Ariel Gabizon, Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, and Anat Paskin-Cherniavsky. Non-interactive secure multiparty computation. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 387–404. Springer, 2014.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988.

[BHI+20] Marshall Ball, Justin Holmgren, Yuval Ishai, Tianren Liu, and Tal Malkin. On the complexity of decomposable randomized encodings, or: How friendly can a garbling-friendly PRF be? In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 86:1–86:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[BHNS20] Amos Beimel, Iftach Haitner, Kobbi Nissim, and Uri Stemmer. On the round complexity of the shuffle model. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part II*, volume 12551 of *Lecture Notes in Computer Science*, pages 683–712. Springer, 2020.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 784–796. ACM, 2012.

[BIW10] Omer Barkol, Yuval Ishai, and Enav Weinreb. On d-multiplicative secret sharing. *J. Cryptol.*, 23(4):580–593, 2010.

[BKN18] Amos Beimel, Eyal Kushilevitz, and Pnina Nissim. The complexity of multiparty psm protocols and related models. In *EUROCRYPT (2)*, volume 10821 of *Lecture Notes in Computer Science*, pages 287–318. Springer, 2018.

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In *Advances in Cryptology – EUROCRYPT 2018*, volume 10821 of *Lecture Notes in Computer Science*, pages 500–532. Springer, 2018.

[CC06] Hao Chen and Ronald Cramer. Algebraic geometric secret sharing schemes and secure multi-party computations over small fields. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 521–536. Springer, 2006.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 11–19. ACM, 1988.

[CCKS22] Wei-Ning Chen, Christopher A. Choquette-Choo, Peter Kairouz, and Ananda Theertha Suresh. The fundamental price of secure aggregation in differentially private federated learning. *CoRR*, abs/2203.03761, 2022.

[Che21] Albert Cheu. Differential privacy in the shuffle model: A survey of separations. *CoRR, abs/2107.11839*, 2021.

[CSU+19] Albert Cheu, Adam D. Smith, Jonathan R. Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *38th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt 2019)*, 2019.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In Frank Thomson Leighton and Michael T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 554–563. ACM, 1994.

[GDDS21] Antonious M. Girgis, Deepesh Data, Suhas Diggavi, and Ananda Theertha Suresh. On the rényi differential privacy of the shuffle model. *arXiv preprint arXiv:2105.05180*, 2021.

[GIK+24] Adrià Gascón, Yuval Ishai, Mahimna Kelkar, Baiyu Li, Yiping Ma, and Mariana Raykova. Computationally secure aggregation and private information retrieval in the shuffle model. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 4122–4136. ACM, 2024.

[GIS18] Sanjam Garg, Yuval Ishai, and Akshayaram Srinivasan. Two-round MPC: information-theoretic and black-box. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 123–151. Springer, 2018.

[GMPV20] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. Private Aggregation from Fewer Anonymous Messages. In *EUROCRYPT*, 2020.

[GMW19] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In Oded Goldreich, editor, *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328. ACM, 2019.

[GS96] Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of number theory*, 61(2):248–273, 1996.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 468–499. Springer, 2018.

[HIJ+17] Shai Halevi, Yuval Ishai, Abhishek Jain, Ilan Komargodski, Amit Sahai, and Eylon Yogev. Non-interactive multiparty computation without correlated randomness. *IACR Cryptol. ePrint Arch.*, page 871, 2017.

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. Ot-combiners via secure computation. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 393–411. Springer, 2008.

[HIKR18] Shai Halevi, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. Best possible information-theoretic MPC. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 255–281. Springer, 2018.

[HIKR23] Shai Halevi, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. Additive randomized encodings and their applications. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part I*, volume 14081 of *Lecture Notes in Computer Science*, pages 203–235. Springer, 2023.

[Hiw25] Keitaro Hiwatashi. Negative results on information-theoretic additive randomized encodings. In Marc Fischlin and Veelasha Moonsamy, editors, *Applied Cryptography and Network Security*, pages 136–157, Cham, 2025. Springer Nature Switzerland.

[IK97] Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997, Ramat-Gan, Israel, June 17-19, 1997, Proceedings*, pages 174–184. IEEE Computer Society, 1997.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304. IEEE Computer Society, 2000.

[IKLM24] Yuval Ishai, Mahimna Kelkar, Daniel Lee, and Yiping Ma. Information-theoretic single-server PIR in the shuffle model. In Divesh Aggarwal, editor, *5th Conference on Information-Theoretic Cryptography, ITC 2024, August 14-16, 2024, Stanford, CA, USA*, volume 304 of *LIPIcs*, pages 6:1–6:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and Jürg Wullschleger. Constant-rate oblivious transfer from noisy channels. In *Advances in Cryptology – CRYPTO 2011, 31st Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14–18, 2011, Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 667–684. Springer, 2011.

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS*

2006), 21-24 October 2006, Berkeley, California, USA, Proceedings, pages 239–248. IEEE Computer Society, 2006.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Extracting correlations. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, Atlanta, Georgia, USA, October 25-27, 2009*, pages 261–270. IEEE Computer Society, 2009.

[IKY26] Yuval Ishai, Eyal Kushilevitz, and Sahar Alon Yerushalmi. On perfect additive randomized encodings. In Dana Dachman-Soled and Daniel Wichs, editors, *Theory of Cryptography – TCC 2026, Part I*, volume 17191 of *Lecture Notes in Computer Science*, pages 84–113. Springer, 2026.

[Ish13] Yuval Ishai. Randomization techniques for secure computation. In Manoj Prabhakaran and Amit Sahai, editors, *Secure Multi-Party Computation*, volume 10 of *Cryptology and Information Security Series*, pages 222–248. IOS Press, 2013.

[KH+21] Antti Koskela, Antti Honkela, et al. Tight accounting in the shuffle model of differential privacy. *arXiv preprint arXiv:2106.00477*, 2021.

[KLS21] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 5201–5212. PMLR, 2021.

[LM20] David Lanzenberger and Ueli Maurer. Coupling of random systems. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography — TCC 2020*, volume 12552 of *Lecture Notes in Computer Science*, pages 207–240, Cham, December 2020. Springer International Publishing.

[MPR07] Ueli M. Maurer, Krzyszf Z. Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *Advances in Cryptology — CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 130–149. Springer-Verlag, August 2007.

[NS95] Moni Naor and Adi Shamir. Visual cryptography. In Alfredo De Santis, editor, *Advances in Cryptology — EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1995.

[SCM22] Mary Scott, Graham Cormode, and Carsten Maple. Applying the shuffle model of differential privacy to vector aggregation. *arXiv preprint arXiv:2112.05464*, 2022.

[SY26] Nobutaka Shimizu and Kenji Yasunaga. Hardness amplification beyond boolean functions. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing – STOC 2026*. ACM, 2026.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

A Las-Vegas Correctness

Recall that our LARE (Theorem 3.3) has a correctness error (vanishing with $|\mathbb{G}|$) that propagates throughout our constructions. In this appendix, we observe that

1. Our LARE in fact has a stronger form of *ε-one-sided* correctness, meaning that whenever the output of the function f is 1, the construction is correct, but when the output is 0, the construction may output 1 with probability at most ε. In our case, $\varepsilon = O(|\mathbb{G}|^{-1})$.

2. A LARE with one-sided correctness can be used to boost the correctness of any ARE to *Las-Vegas correctness*, where the ARE does not error, but does declare failure (outputs $\perp$) with small probability.

Lemma A.1. *Let $f : D \times D \to D$ be a finite function and assume it has ε'-correct and δ'-secure ARE scheme ARE_f where the parties use randomness from domain R. Then there exists a function $g : (D \times R) \times (D \times R) \to \{0,1\}$, any ε-one-sided-correct and δ-secure LARE scheme LARE_g for g implies an ARE scheme LVARE_f for f with $(\varepsilon + \varepsilon')$-las-vegas-correctness and $(\delta + \delta' + \varepsilon')$ security.*

Corollary A.2 (of Theorems 3.3,4.1 and Lemma A.1)**.** *For any (efficient) group $\mathbb{G}$ and parameter $\tau \in \mathbb{N}$, any finite function $f : D \times D \to D$ has a $O(\tau/|\mathbb{G}|)$-las-vegas-correct, $O(\frac{\tau}{|\mathbb{G}|} + 2^{-\tau})$-secure ARE over $\mathbb{G}^{O(\tau)}$. Encoding/decoding efficiency is $\tau \cdot \mathrm{polylog}(|\mathbb{G}|)$.*

Proof of Lemma A.1. We define the function $g((x, r_x), (y, r_y))$ as the function that tests whether the randomness is faulty:

$$g\left((x, r_x), (y, r_y)\right) = 1 \text{ iff } \mathsf{ARE}_f.\mathsf{Dec}(\widehat{x} + \widehat{y}) \neq f(x, y) \ ,$$

where $\widehat{x}, \widehat{y}$ are encodings of x and y by the two parties using randomness r_x and r_y. .

Our Las Vegas scheme LVARE_f operates by running ARE_f and LARE_g simultaneously, where in LARE_g the parties include as part of their inputs the randomness r_x, r_y they used in the instance of ARE_f. The decoder for LVARE_f outputs $\perp$ if LARE_g decoder outputs 1; otherwise, it outputs the decoded result of ARE_f.

Correctness:

- If ARE_f provides an incorrect output (occurring with probability at most ϵ'), the function g outputs 1, and consequently, due to the one-sided correctness of LARE_g, it always decodes to 1. Hence, LVARE_f always outputs $\perp$ and identifies the error correctly.

- If ARE_f correctly decodes, then LVARE_f outputs the correct answer, unless LARE_g falsely decodes to 0, which happens with probability at most ε, and in which case LVARE_f outputs $\perp$.

Overall, the probability of outputting $\perp$ is bounded by $\varepsilon + \varepsilon'$, and otherwise decoding is correct.

Security: The simulator $\mathsf{Sim}(f(x, y))$ outputs the result of the two underlying simulators

$$\mathsf{ARE}_f.\mathsf{Sim}(f(x, y)), \mathsf{LARE}_g.\mathsf{Sim}(0, \perp) \ .$$

When $g((x, r_x), (y, r_y)) = 0$, due to the LARE and ARE security the statistical error of at most $\delta + \delta'$. At the same time, by the ε'-correctness of ARE_f the probability that $g((x, r_x), (y, r_y)) = 1$, over r_x, r_y, is at most ε'. Overall, the security error is at most $\delta + \delta' + \varepsilon'$. □

B Perfect Correctness

This appendix, added in November 2026, shows a perfectly correct two-party ARE for Boolean OR. Since the ARE for OR is the only source of correctness error in our constructions, we obtain a perfectly correct ARE for all functions

Theorem B.1. *There exists an (efficient) group $\mathbb{G}$ of order $2^{O(\tau^3)}$, such that any Boolean function $f : [d] \times [d] \to \{0,1\}$ has a perfectly correct, $2^{-\tau}$-secure ARE over $\mathbb{G}$. Encoding/decoding take time* $\mathrm{poly}(\tau)$.

The corresponding construction is a result of interaction between the authors and ChatGPT Astra. In particular, the initial positive answer to the question of perfect correctness was given by Astra.

B.1 Overview

The high-level idea is based on two gadget AREs.

Equality with a Payload. Here Alice holds an index $i \in D$ and a (payload) bit b, and Bob holds an index $j \in D$. The decoder always learns whether $i = j$. On disagreement it also recovers b. Given agreement, the sum encoding statistically hides both the payload and the common index.

Constructing this gadget ARE extends the one pole, one hole construction for perfectly correct equality in [BEG25]. Let $S = \{2, \ldots, N\}$. Alice places b at coordinate i and independent uniform elements of S elsewhere. Bob places a uniform element of S at coordinate j and zero elsewhere. Coordinates are added modulo $M > 2N$, so no sum wraps around. On disagreement the unique coordinate in $\{0,1\}$ is the payload. On agreement no coordinate is in $\{0,1\}$; the sum is uniform in S at every coordinate except i, where it is uniform in $S + b$, and hence $\approx N^{-1}$-close in statistical distance.

Distributional OR. Here Alice holds (i, x) and Bob holds (j, y), where $i, j \in D$ and $x, y \in \{0,1\}$. On agreement, the sum encoding reveals $\mathrm{OR}(x, y)$ and can be simulated from this output when the common index $i = j$ is uniform. For fixed indices $i \neq j$, the sum encoding statistically hides all information x, y.

The construction works as follows. Take $D = \{0,1\}$. For an n-bit string a, write a' for its integer value. The parties sample independent uniform strings a, b and output

$$(ia, (1-i)a, (-1)^i x + a'), \qquad ((1-j)b, jb, (-1)^j y + b'),$$

adding strings by XOR and the last coordinate modulo 3. On agreement $i = j$ the first two coordinates reveal a, b allowing to recover $(-1)^i(x + y)$, which is zero exactly when $\mathrm{OR}(x, y) = 0$ and otherwise uniform in $\mathbb{Z}_3^* = \{-1, 1\}$. When $i \neq j$, only $a \oplus b$ is revealed, in which case $a' + b' \bmod 3$ is $2^{-\Omega(n)}$-close to uniform.

Combining the Two Gadgets. To obtain an ARE for OR, we invoke $2k$ instances with independent uniform index pairs. In the first k instances, we use random secret shares of x as the payloads. In the last k, we use secret shares of y, reversing the roles. Every instance also encodes the original inputs x, y using distributional OR on the corresponding indices. If any pair agrees, we can decode the OR using the corresponding instance. If none agrees, all shares are revealed and we learn x, y and in particular their OR.

As for security, except with probability $O(2^{-k})$, each half contains an agreement. Then at least one share of each input is hidden, and the revealed shares do not leak information about the inputs. The guarantees of the equality and distributional OR gadgets allow simulating any agreement from $\mathrm{OR}(x, y)$ and disagreement from scratch.

B.2 Formal Construction and Analysis

Fix $\tau \geq 1$. Throughout the construction and analysis, set $k = n = \tau$, $N = 2^\tau + 1$, and $M = 2N + 1$. Let $S = \{2, \ldots, N\}$.

Theorem B.2. *For every $\tau \geq 1$, Boolean OR has a perfectly correct two-party ARE over*

$$\mathbb{G} = \left(\mathbb{Z}_M^2 \times (\mathbb{Z}_2^n)^2 \times \mathbb{Z}_3\right)^{2k}$$

with statistical security error at most $2^{-\Omega(\tau)}$.

Lemma B.3. *For every $a \in \{0,1\}^n$, let a' be the integer it represents. For independent uniform $a, b \in \{0,1\}^n$ and an independent uniform $U \in \mathbb{Z}_3$,*

$$\Delta\big((a \oplus b, a' + b' \bmod 3), (a \oplus b, U)\big) \leq 2^{-\Omega(n)}.$$

Proof. Condition on $z = a \oplus b$, indexing bits by $\ell = 1, \ldots, n$. We have

$$a' + b' = \sum_{\ell=1}^{n} 2^{\ell-1}(a_\ell + b_\ell) = \sum_{\ell : z_\ell = 1} 2^{\ell-1} + \sum_{\ell : z_\ell = 0} 2^\ell a_\ell.$$

Conditioned on z, the bits a_ℓ remain independent and uniform. Thus the first sum is fixed, and each term in the second sum is independently uniform in $\{0, 2^\ell\}$. We now consider this sum modulo 3. Since $2^\ell \neq 0 \bmod 3$, the sum of any two such random contributions assigns probability exactly $1/4$ to two residues x_1, x_2 and exactly $1/2$ to the third x_3. Hence its distribution can be written as

$$\frac{3}{4}U(\mathbb{F}_3) + \frac{1}{4}x_3.$$

If z has m zero coordinates, pairing them shows that the sum is within $(1/4)^{\lfloor m/2 \rfloor}$ of uniform. By a Chernoff bound, z has $m = \Omega(n)$ zero coordinates except with probability $2^{-\Omega(n)}$. Thus the overall statistical distance is $2^{-\Omega(n)}$. (This can also be proved invoking the XOR lemma for finite fields [SY26, AG26].) □

Proof of Theorem B.2. We describe the construction in detail and then analyze it.

Encoding. Alice samples secret shares $x_1, \ldots, x_k$ of x, and Bob independently samples secret shares $y_1, \ldots, y_k$ of y. For every $r \in [2k]$, Alice samples a uniform bit i_r and a uniform string $a_r \in \{0,1\}^n$, and Bob samples a uniform bit j_r and a uniform string $b_r \in \{0,1\}^n$.

The parties sample $A_r, B_r \in \mathbb{Z}_M^2$, with coordinates indexed by $\{0,1\}$ as follows:

- For $r \leq k$, Alice sets $A_r[i_r] = x_r$ and samples $A_r[1 - i_r]$ uniformly from S. Bob samples $B_r[j_r]$ uniformly from S and sets $B_r[1 - j_r] = 0$.
- For $r > k$, Bob sets $B_r[j_r] = y_{r-k}$ and samples $B_r[1 - j_r]$ uniformly from S. Alice samples $A_r[i_r]$ uniformly from S and sets $A_r[1 - i_r] = 0$.

Alice and Bob output, respectively, for every $r \in [2k]$:

$$\left(A_r, i_r a_r, (1 - i_r)a_r, (-1)^{i_r} x + a'_r\right), \qquad \left(B_r, (1 - j_r)b_r, j_r b_r, (-1)^{j_r} y + b'_r\right).$$

Addition, for each r, is done over the group $\mathbb{Z}_M^2 \times \mathbb{Z}_2^n \times \mathbb{Z}_2^n \times \mathbb{Z}_3$.

Decoding. Consider the sum encoding $(V_r, u_r, v_r, t_r)_{r \in [2k]}$. We consider two cases:

- **Case 1: equality occurs.** If for some r neither coordinate of V_r lies in $\{0,1\}$ and return zero if $t_r - u'_r - v'_r = 0 \bmod 3$, and one otherwise.
- **Case 2: no equality occurs.** Otherwise, for each r, let p_r be the value of the (unique) coordinate of V_r in $\{0,1\}$, and return

$$\mathrm{OR}\left(\bigoplus_{r=1}^{k} p_r, \bigoplus_{r=k+1}^{2k} p_r\right).$$

Perfect Correctness. All integer sums in V_r are at most $2N < M$. Case 1 occurs iff $i_r = j_r$ for some r: for any such r, both coordinates of V_r are ≥ 2 and $\{u_r, v_r\} = \{b_r, a_r\}$. Hence

$$t_r - u'_r - v'_r = (-1)^{i_r}(x+y) \pmod 3 = \pm\mathrm{OR}(x,y).$$

In Case 2, all shares of x, y are recovered from the payloads p_r, and $\mathrm{OR}(x,y)$ is computed correctly.

Statistical Security. We describe the simulator, which is given $\mathrm{OR}(x,y)$. Sample i_r, j_r independently and uniformly from $\{0,1\}$ for every $r \in [2k]$. If either half has no index r with $i_r = j_r$, output $\perp$. Otherwise independently generate each instance as follows:

- If $i_r = j_r$, sample V_r uniformly from S^2, and u_r, v_r independently uniformly from $\{0,1\}^n$. Set $t_r = u'_r + v'_r + q_r \bmod 3$, where $q_r = 0$ for $\mathrm{OR}(x,y) = 0$, and q_r is uniform in $\{-1,1\}$ for $\mathrm{OR}(x,y) = 1$.
- If $i_r \neq j_r$, sample a uniform bit p_r. Let $h_r = i_r$ for $r \leq k$, and $h_r = j_r$ for $r > k$. Set $V_r[h_r] = p_r$ and $V_r[1-h_r] = s_r + \widetilde{s_r}$, for independent uniform $s_r, \widetilde{s_r} \in S$. Sample a uniform string z_r and a uniform $t_r \in \mathbb{Z}_3$. Set $(u_r, v_r) = (0, z_r)$ for $i_r = 0$, and $(z_r, 0)$ otherwise.

We bound the statistical distance between the real and simulated sum encodings using the following hybrids.

Hybrid 0 (real encodings). Output the real sum encoding.

Hybrid 1. Output $\perp$ if either half contains no agreement. This changes the distribution by at most $2 \cdot 2^{-k}$.

Hybrid 2. In each agreeing instance, replace V_r by an independent uniform vector in S^2. For every fixed choice of indices and shares, its coordinates are uniform in S and $S + p_r$. Thus the distance from the previous hybrid is at most $2k/(N-1)$.

Hybrid 3. In each disagreeing instance, replace t_r by an independent uniform element of $\mathbb{Z}_3$. The string coordinates reveal only $a_r \oplus b_r$, while $t_r = a'_r + b'_r + (-1)^{i_r}x + (-1)^{j_r}y \bmod 3$. By Lemma B.3, the distance from the previous hybrid is at most $2k \cdot 2^{-\Omega(n)}$.

Hybrid 4. In each agreeing instance, generate (u_r, v_r, t_r) as in the simulator. This does not change the distribution: after Hybrid 2, the uniform common index is hidden and independent of the remaining coordinates. The strings u_r, v_r are independent and uniform for either common index, and the residual $(-1)^{i_r}(x+y)$ is zero when $\mathrm{OR}(x,y) = 0$ and uniform in $\{-1,1\}$ otherwise.

Hybrid 5 (simulated encodings). Replace the payloads exposed in disagreeing instances by independent uniform bits. This does not change the distribution: each half contains an agreement, so the exposed payloads form a proper subset of that input's secret shares and are jointly uniform. The hidden shares no longer affect any coordinate. This is exactly the simulator's distribution.

Summing the distances, the security error is at most

$$O(k)\left(2^{-k} + N^{-1} + 2^{-\Omega(n)}\right) = 2^{-\Omega(\tau)}.$$

□