Privacy-Preserving Data Center Demand Response using Multi-Party Computation

Seyda Nur Guzelhan*[1][0000−0002−5384−2397], Fatih Acun[1][0000−0002−2479−8972], Can Hankendi[1][0000−0002−5717−8905], Ayse K. Coskun[1][0000−0002−6554−088X], and Ajay Joshi[1][0000−0002−3256−9942]

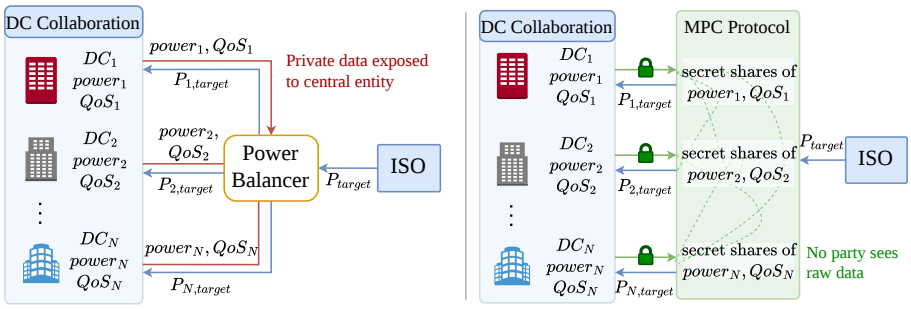
Boston University, Boston, MA 02215, USA
{seyda, acun, hankendi, acoskun, joshi}@bu.edu

Abstract. The rapid growth of AI has significantly increased data center energy demand, placing increasing pressure on power grids. Demand Response (DR) programs utilize the flexibility of power consumers, such as data centers, to help balance the supply and demand in power grids. In collaborative data center DR participation, multiple data centers share information with an external coordinator for improved power dispatch and quality of service for their workloads. However, disclosing sensitive information such as power usage and workload performance to an untrusted third party raises significant privacy concerns. To address this, we use Multi-Party Computation (MPC) to perform secure power dispatch without revealing sensitive inputs. Standard MPC has heavy communication overhead, which challenges real-time DR requirements. To meet real-time DR requirements, we optimize our system by tailoring fixed-point bitwidths and substituting expensive divisions with polynomial approximations and Newton-Raphson iterations. Evaluated using the MP-SPDZ library, our optimizations yield up to 33× fewer communication rounds and a 45× speedup, delivering sub-second latency for up to 16 data centers while matching the power dispatch accuracy of the baseline.

Keywords: Collaborative Data Center Demand Response · Privacy · Multi-Party Computation

1 Introduction

The rapid rise of artificial intelligence (AI) and cloud computing applications has driven a sharp increase in both the number and the scale of data centers [15]. Being large-scale electricity consumers with unique power consumption profiles, data center loads create new operational challenges for power grids [26]. However, with their real-time workload modulation capabilities, data centers are uniquely suited to support power grids by dynamically adjusting their power consumption through participation in the demand response (DR) programs. Independent System Operators (ISOs) run DR programs to coordinate with flexible electricity users, requesting increases or decreases in power consumption to balance the grid supply and demand. Prior work on data center DR (DCDR) participation employs methods such as hardware power capping and workload scheduling to



(a) Plain collaborative DCDR. (b) Proposed secure collaborative DCDR.

Fig. 1: Overview of the plain and secure collaborative DCDR frameworks. In the plain setting, data centers reveal some of their operational constraints to a third-party aggregator. In the secure setting, data centers secret-share their constraints, and the aggregator computes the power allocation without accessing raw data.

modulate data center power consumption while satisfying workloads' quality-of-service (QoS) requirements [16, 30]. To further reduce the risk of QoS violations and increase compatibility with DR requirements, collaborative DCDR enables workload migration across data centers that are owned by the same company [29]. Recent frameworks introduce collaboration across data centers without needing workload migration: a third-party aggregates real-time metrics from multiple data centers to determine power targets that respect each facility's QoS requirements [2], as illustrated in Figure 1a. However, this approach creates a fundamental tension between coordination and privacy. To participate, data centers may need to share operational information, such as power usage and workload latency, with the coordinating entity. Such data can be commercially sensitive, as power consumption patterns may reflect aspects of facility utilization and operational scale, while QoS-related metrics can provide insight into service-level objectives and system performance margins. These concerns are not hypothetical: industry surveys identify privacy as a primary barrier to inter-operator coordination in energy markets [15]. Thus, the reluctance to share operational details is a key barrier to broader adoption of collaborative DCDR.

In addition to the privacy problem, DR programs like Regulation Service Reserves (RSR) are latency-sensitive, leaving little margin for secure computation overhead. Differential Privacy injects noise that degrades dispatch quality at small coalition sizes [10], and Homomorphic Encryption incurs costs orders of magnitude above plaintext, making real-time deployment infeasible [11]. Multi-party computation (MPC) is a natural fit: multiple parties jointly evaluate a function over private inputs without revealing them [12]. MPC produces exact outputs, preserving the correctness of the power dispatch, and its overhead is dominated by communication rounds rather than computation, which can be optimized for the specific operations in the target formula. We therefore focus on MPC in this paper.

In this work, we design a *secure collaborative DCDR system* that replaces the untrusted third-party coordinator with an efficient MPC-based protocol (Figure 1b). Each participant secret-shares its power consumption and QoS parameters, and the coordinating party executes the allocation entirely over the shares, producing optimal power targets without ever accessing the underlying data. However, directly evaluating the allocation formula under MPC is prohibitively expensive due to two sources of overhead. First, the default 64-bit fixed-point representation allocates precision far beyond what physical power sensors provide; since communication volume scales linearly with bitwidth, the excess precision directly inflates message sizes. Second, each secure division on 64-bit operands costs roughly 40 sequential communication rounds, compared to 2 rounds for a multiplication, making divisions the dominant cost driver: at a 5 ms round trip time, a single division alone incurs over 200 ms of network-induced latency. To address these, we tailor the fixed-point bitwidth to the range of power and QoS values, which is publicly known from hardware specifications and service-level agreements, and we replace the costly secure divisions with a two-stage reciprocal method that uses iterative multiplications [5, 25].

We demonstrate the efficacy of our design using a recent collaborative DCDR framework, Conductor [2], which coordinates participation in the RSR program by dynamically allocating the ISO's aggregate power target across data centers in a QoS- and power-aware manner. We choose Conductor because it does not require inter–data center workload migration and relies only on plaintext information sharing for power allocation, leaving room for stronger privacy enhancements. We implement the secure power allocation in MP-SPDZ [17], a highly optimized multi-protocol MPC library, and benchmark performance across different security settings. Our contributions are as follows:

- We design and evaluate an MPC-based secure collaborative DCDR system that eliminates the need to disclose private data to any coordinating entity.
- We optimize our system by introducing tailored fixed-point representations for secure arithmetic and by replacing costly secure division operations with a two-stage Chebyshev–Newton–Raphson reciprocal method.
- We provide a comprehensive empirical evaluation along four axes: (1) security settings spanning semi-honest to malicious adversaries, (2) scalability up to 64 data centers, (3) arithmetic precision sensitivity of the power dispatch, and (4) MPC communication costs under local- and wide-area networks.

Our optimizations yield up to a 33× reduction in communication rounds and a 45× reduction in end-to-end latency compared to a naïve MPC implementation using default MP-SPDZ arithmetic. As a result, our framework delivers sub-second execution for up to 16 data centers and sub-minute execution for 64 data centers, while matching the power dispatch accuracy of the plaintext Conductor.

The rest of the paper is as follows: Section 2 reviews related work on DCDR and privacy-preserving smart grid protocols. Section 3 provides preliminaries on collaborative DCDR and MPC. Section 4 presents our MPC-based DCDR method. Section 5 evaluates the framework across security models, arithmetic precisions, and network conditions. Section 6 concludes the paper.

2 Related Work

Securing collaborative DCDR sits at the intersection of two lines of research: mechanisms for DCDR programs, and privacy-preserving protocols for smart grid applications. We survey each and identify the gap that motivates our work.

Data Center Demand Response. Data centers have been widely studied as candidates for participation in DR programs, including emergency DR and RSR [13, 6]. Prior work proposes a range of mechanisms to modulate data center power draw in accordance with program requirements, such as hardware power capping, job co-location, and temporal workload shifting [16, 20]. An important challenge in these approaches is preserving workload QoS requirements while enforcing power constraints. To mitigate QoS degradation, QoS-aware techniques proactively meet service targets by allocating compute resources according to application-level requirements [30]. Prior work considers collaborative DCDR to further improve QoS and enhance DR performance by migrating workloads between data centers [29]. Complementary studies consider collaboration across data centers without migrating workloads, instead coordinating DR participation by dynamically redistributing power budgets across data centers to meet a shared target [2]. Similar to multi-data-center collaboration, tenant-level collaboration has been explored in multi-tenant data centers via game-theoretic and market-based frameworks for DR participation [7].

Privacy-Preserving Protocols in Smart Power Grids. MPC has seen growing adoption in the smart grid [9]. However, the computational profile of existing MPC deployments differs fundamentally from what DCDR requires. Privacy-preserving smart metering protocols [22] aggregate integer-valued power readings over billing intervals, where both the input domain and the update frequency are modest enough that MPC's communication overhead remains manageable. Peer-to-peer energy trading extends MPC to market-clearing mechanisms [28], but the computation is triggered once per auction round, so latency pressure is limited. Power flow studies using MPC [14] demonstrate that richer numerical workloads are feasible in principle; however, these analyses solve a single static snapshot offline and do not contend with repeated, time-critical updates. Tian et al. [27] propose a private multi-area economic dispatch under a semi-honest threat model, demonstrating that MPC can handle real-valued grid optimizations, but their framework solves a single optimization problem without real-time latency requirements and does not analyze stronger adversary models. What is missing from this body of work is an MPC pipeline that meets collaborative DCDR requirements: real-valued inputs, sub-second re-execution, and robustness across threat models.

3 Preliminaries

In this section, we provide the background for the collaborative DCDR setting and the sensitive data problem (Section 3.1), and the MPC primitives that enable a secure evaluation (Section 3.2).

3.1 Collaborative Data Center Demand Response

Data centers can effectively engage in DR programs by utilizing hardware power capping and power-aware scheduling to modulate their power consumption [16]. They can adjust their power consumption rapidly and provide seconds-granularity DR, as required in RSR programs [23]. In the RSR program, participants declare their average power consumption, $\bar{P}$, and reserve capacity, R, before program execution. During execution, the participant is required to track a power target, updated every few seconds (e.g., 2 seconds), calculated as follows:

$$P_{\text{target}}(t) = \bar{P} + y(t)R, \tag{1}$$

where $y(t) \in [-1, 1]$ denotes the regulation signal. Accordingly, the target power ranges from $\bar{P} - R$ to $\bar{P} + R$. Performance on power tracking (tracking error) is measured as the absolute distance between the power target and the consumption (P_{consumed}) normalized by the provided reserve, $|P_{\text{target}}(t) - P_{\text{consumed}}(t)|/R$.

In collaborative RSR participation, multiple data centers form a coalition and interact with the grid as a single aggregated resource. The coalition receives a single power target, $P_{\text{target}}(t)$, which is then distributed among the participating data centers as individual targets, $P_{i,\text{target}}(t)$. This collaborative setting is motivated by the observation that aggregate DR targets can often be satisfied more effectively through coordinated power allocation across sites than by enforcing identical power constraints independently at each data center. By accounting for differences in workload QoS sensitivity across data centers, such coordination can reduce QoS violations while improving tracking of the aggregate DR target. Prior work, Conductor [2], has demonstrated the potential of this collaborative approach for mitigating QoS violations in data centers and improving the power tracking capability as a collaboration. Our implementation for the baseline collaborative DCDR follows the design introduced in Conductor, in which each data center provides real-time operational signals, such as QoS-related metrics, to a power balancer, which determines QoS-aware per-data-center power targets. The power balancer calculates the individual power target by distributing the collaborative power target based on proportional allocation with respect to the real-time QoS (Q_i^R) and the average power consumption ($\bar{P}_i$) of each data center, denoted by subscript i, as follows:

$$P_{\text{i,target}}(t) = P_{\text{target}}(t) \left(\tau \frac{Q_i^R(t)}{\sum_{k=1}^{N} Q_k^R(t)} + \phi \frac{\bar{P}_i}{\sum_{k=1}^{N} \bar{P}_k} \right), \tag{2}$$

where $P_{\text{target}}(t)$ is the total power target for data center collaboration at time t, $P_{\text{i,target}}(t)$ is the power budget assigned by the power balancer to a data center, τ and ϕ are parameters to balance the impact of Q^R and $\bar{P}$ on power allocations, and N is the total number of data centers in the collaboration.

In this collaborative DCDR setting, data centers currently exchange information with the power balancer in plaintext. However, both QoS and power consumption can be commercially sensitive, especially when data centers belong to different organizations. To address this confidentiality gap, we use MPC

to compute QoS-aware power allocations without learning any individual data center's raw QoS metrics or power consumption.

3.2 Multi-Party Computation (MPC)

MPC enables N mutually distrusting parties to jointly evaluate a function f while guaranteeing two properties: 1) correctness, such that the output matches the result of computing f on plaintext inputs, and 2) privacy, such that no coalition of corrupted parties learns anything beyond what can be inferred from their own inputs and the output [12]. In our setting, the function f is the power allocation in Equation 2, and each data center's QoS metric and power consumption serve as private inputs.

Threat models. MPC security guarantees depend on assumptions about the adversary's behavior and reach. A *semi-honest* adversary follows the protocol but tries to infer private information from observed messages, while a *malicious* adversary may deviate arbitrarily to learn private data or corrupt the output [12]. The corruption threshold t specifies how many of the N parties the adversary may control: protocols with $t < N/2$ operate in the *honest-majority* setting, while those tolerating $t \geq N/2$ assume *dishonest majority* and require heavier cryptographic primitives such as oblivious transfer [19, 18]. Stronger threat models provide broader protection but at higher communication cost, a trade-off we quantify across four protocol families in Section 5.

Cost of secure arithmetic. MPC protocols split each private input into secret shares distributed across all parties, such that any subset below the corruption threshold learns nothing about the original value [24]. Local arithmetic on these shares is lightweight, but any operation that combines values held by different parties requires all parties to exchange messages and wait for a full network round-trip time (RTT) before the protocol can proceed. It is this communication that dominates MPC cost, as total latency grows linearly with the number of sequential communication rounds.

The number of required rounds and the amount of data that must be exchanged depend on the operation. Addition of shared values is non-interactive: each party simply adds its local shares. Multiplication requires 2 rounds to consume a precomputed triple [4], with message size scaling with operand bitwidth. Division is far more expensive on both counts: MPC provides no native inverse for encoded real numbers, so a single secure division on 64-bit operands costs roughly 40 sequential rounds, compared to 2 rounds for a multiplication [5, 17]. For parties connected over a wide-area network with an RTT of 5 ms, this means a single division alone incurs at least $40 \times 5 = 200$ ms of network-induced latency, independent of compute power, while a multiplication adds only 5–10 ms. Division is therefore the primary cost driver in secure collaborative DCDR, and the main target for the optimizations presented in Section 4.

Fixed-point arithmetic. MPC protocols natively operate over finite fields or rings, not real numbers [8]. To handle the real-valued quantities in Equation 2, values are encoded in fixed-point representation with ℓ bits of precision, with

Algorithm 1 Secure Multi-Data-Center Power Balancing via MPC

Require: Secret shares $\langle \bar{P}_i \rangle$, $\langle Q_i^R \rangle$ of each DC_i, $i \in \{1, \ldots, N\}$; collaborative power target P_{target} from ISO; balancing factors τ, ϕ
Ensure: Individual power target $P_{i,\text{target}}$ revealed only to DC_i
1: **for** $i = 1$ **to** N **do**
2: Secret-share $\langle \bar{P}_i \rangle$ and $\langle Q_i^R \rangle$ to all parties ▷ Send private inputs
3: – sync –
4: $\langle P_{i,\text{target}} \rangle \leftarrow P_{\text{target}} \cdot \left(\tau \frac{\langle Q_i^R \rangle}{\langle Q_{\text{sum}}^R \rangle} + \phi \frac{\langle \bar{P}_i \rangle}{\langle P_{\text{sum}} \rangle} \right)$ ▷ Secure power balancing
5: – sync –
6: Reconstruct $P_{i,\text{target}}$ from $\langle P_{i,\text{target}} \rangle$ ▷ Receive private outputs
7: **end for**
8: **return** $P_{i,\text{target}}$

each multiplication incurring an additional truncation round [5]. The choice of ℓ directly affects both accuracy and cost: too few bits introduce numerical errors, while too many inflate the field size and increase communication [5, 21]. When input values are bounded to a known range, as is the case for power and QoS metrics, the bitwidth can be tailored accordingly rather than set conservatively. We exploit this in Section 4.

Online and offline phases. The offline phase precomputes the cryptographic material needed for secure multiplications (e.g., multiplication triples) independently of the inputs, and can therefore be amortized or pipelined across control intervals. The online phase begins once parties provide their inputs and consume this precomputed material to evaluate the target function. For latency-sensitive applications such as DR, where allocations must be computed every few seconds, the online phase is the binding constraint when offline preprocessing is enabled.

4 MPC-Based Collaborative DCDR

Our goal is to design a collaborative DCDR system to efficiently compute individual power targets for each data center without exposing any participant's private data. The core idea is as follows: every data center secret-shares its private inputs (power and QoS), the MPC protocol performs the balancing arithmetic on these shares, and each data center receives shares belonging only to its own target to reconstruct the output. Algorithm 1 summarizes the end-to-end procedure. Two practical challenges arise when moving from this conceptual protocol to an efficient implementation: (1) choosing the right number representation for secure arithmetic, and (2) eliminating expensive secure division operations. We address each challenge below.

4.1 Secure Arithmetic Representation

The MP-SPDZ library offers two ways to represent real numbers inside a secure protocol: secure fixed-point (`sfix`) and secure floating-point (`sfloat`) [17]. The

Algorithm 2 Division-Free Power Balancing via Newton–Raphson Iterations

Require: Secret-shared sums $\langle Q^R_{\text{sum}} \rangle$, $\langle \bar{P}_{\text{sum}} \rangle$; Newton–Raphson iterations K
Ensure: $\langle 1/Q^R_{\text{sum}} \rangle$, $\langle 1/\bar{P}_{\text{sum}} \rangle$ (secret-shared reciprocals)

for each denominator $\langle x \rangle \in \{\langle Q^R_{\text{sum}} \rangle, \langle \bar{P}_{\text{sum}} \rangle\}$ **do**
  $\langle w \rangle \leftarrow \texttt{PolyApprox}_{1/x}(\langle x \rangle)$ ▷ Chebyshev approximation of $1/x$
  for $k = 1$ **to** K **do**
    $\langle w \rangle \leftarrow \langle w \rangle \cdot (2 - \langle x \rangle \cdot \langle w \rangle)$ ▷ Newton–Raphson refinement
  end for
end for
Replace each division in Alg. 1 with multiplication by the corresponding $\langle w \rangle$

choice matters because it directly affects both computational cost and numerical range. MPC arithmetic is emulated over large prime fields (or integer rings) to maintain security guarantees, so traditional hardware floating-point units, which complete an operation in a single clock cycle, cannot be used [17]. Secure fixed-point is the cheaper format for the operations in our protocol. Addition is a free local operation (0 rounds), and multiplication requires 2 communication rounds (one field multiplication plus one truncation) [5]. The corresponding secure floating-point operations are far more expensive: sfloat addition requires tens of rounds (the exact value depends on the security setting and parameters) due to exponent alignment and renormalization, and sfloat multiplication requires roughly 10 rounds [3]. Plaintext hardware arithmetic, of course, requires no communication at all. The trade-off is a limited dynamic range, determined by the chosen bit-width. The bit-width must be set conservatively to prevent overflow across all possible inputs, which means a fixed-point type often reserves a larger range than most values actually need.

The default `sfix` implementations use 53 precision bits to match the IEEE fp64 precision. In a DCDR setting, however, inputs originate from physical sensors and actuators whose own measurement precision is typically well below 53 bits. Allocating precision that the input data does not need wastes communication and compute in every secure operation. We therefore define custom fixed-point types, denoted $\widetilde{\texttt{sfix}}$, whose bit-widths are tailored to the DCDR value range, and evaluate both 64-bit and 32-bit variants. Reducing from 64 to 32 bits has two direct effects. First, every message exchanged between parties shrinks proportionally, cutting total communication volume by roughly half. Second, field-arithmetic operations on smaller elements are faster, reducing per-operation compute time. The round count for basic multiplications is unchanged, but sub-operations whose depth scales with bit-width, such as post-multiply truncation, require fewer rounds at 32 bits, an efficiency gain we quantify against the loss in numerical precision in Section 5.

4.2 Replacing Secure Division with Multiplication

Algorithm 1 requires division to compute each data center's power target. In plaintext arithmetic, this is trivial, but inside an MPC protocol using fixed-point

arithmetic, a single division requires approximately 40 communication rounds (dominated by normalization and iterative reciprocal computation), compared to just 2 rounds for a multiplication [5], a difference of over 20×. Since communication latency dominates MPC runtime (over wide-area links), eliminating divisions can cut protocol time dramatically. Our approach follows the strategy of von der Heyden et al. [14], who use iterative methods to replace costly inverse operations in secure power-flow computations. The core idea is to compute secret-shared reciprocals in two stages, summarized in Algorithm 2.

Stage 1: Chebyshev initial estimate. The Newton–Raphson iteration for reciprocals converges only when the initial guess w_0 is close enough to the true value; specifically, convergence is guaranteed when $w_0 \in (0,\ 2/x)$ [25]. In a plaintext setting, one could simply inspect x and pick a reasonable starting point, but in MPC the input is secret-shared and cannot be examined. We solve this by fitting a degree-m Chebyshev polynomial to the function $1/x$ over a bounded input range. Evaluating this polynomial on a secret-shared input requires only secure multiplications and additions, producing an estimate $w_0 \approx 1/x$ that falls within the convergence range. In practice, a small degree $m \in [3, 7]$ is sufficient.

Stage 2: Newton–Raphson refinement. Starting from the Chebyshev estimate w_0, we iteratively refine the approximation to $\frac{1}{x}$ with the Newton–Raphson update for division [25]: $w_{k+1} = w_k\,(2 - x\,w_k)$, where k denotes the iteration number. Each iteration doubles the number of correct digits (quadratic convergence) and costs only two secure multiplications and one subtraction from the public constant 2. In practice, 3 to 5 iterations suffice to reach full fixed-point precision. Once the secret-shared reciprocals $\langle 1/Q^R_{\text{sum}} \rangle$ and $\langle 1/\bar{P}_{\text{sum}} \rangle$ are available, every division in Algorithm 1 is replaced by a multiplication, yielding a large reduction in total communication rounds and wall-clock protocol time.

5 Experimental Evaluation

We evaluate the secure power-balancing protocol to answer two questions: how much error, if any, does MPC introduce relative to a plaintext baseline and is that error acceptable, and can the optimized protocol scale to realistic multi-data-center deployments? We evaluate along four axes, summarized in Table 1: MPC protocol, i.e., security setting (four protocols spanning honest/dishonest

Table 1: Experimental evaluation axes: security setting and corresponding MPC protocol, number of parties, arithmetic methods, and network conditions. SH: Semi-honest, Mal: Malicious, HM: Honest majority, DM: Dishonest majority.

MPC Protocol	**Number of Parties**	**Arithmetic**	**Network**
SH-HM: Shamir [8]	2-16 (Case Studies 5.1)	`sfloat`	LAN (1 ms)
SH-DM: Temi [19]	2-64 (Scalability 5.2)	$\widetilde{\texttt{sfix}}_{64}$,$\widetilde{\texttt{sfix}}_{32}$	WAN (5 ms)
Mal-HM: Mal-Shamir [8]		NR_{64}, NR_{32}	WAN (10 ms)
Mal-DM: MASCOT [18]			

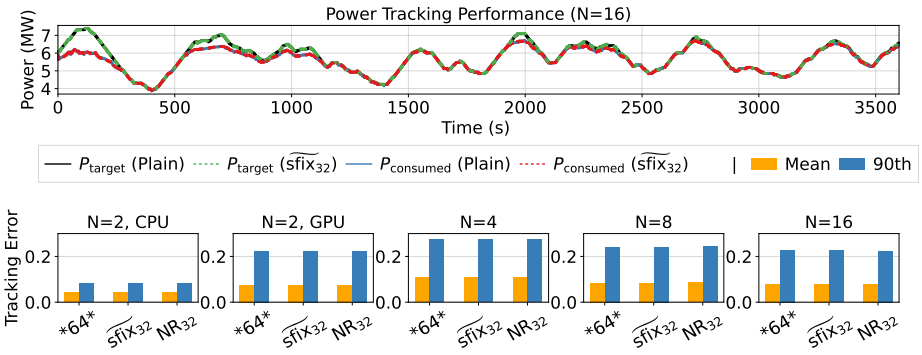


Fig. 2: Power tracking performance (N = number of data centers). **Top:** Aggregate power target (P_{target}) versus consumed power (P_{consumed}) for $N = 16$. The 64-bit MPC traces are identical to the plaintext baseline and thus omitted; the 32-bit traces closely track plaintext with minor deviations. **Bottom:** Mean and 90th-percentile tracking error ($|P_{\text{target}} - P_{\text{consumed}}|/R$) across plaintext and MPC optimized methods from Section 4. 64-bit variants match the plaintext exactly, and 32-bit variants introduce negligible error.

majority and semi-honest/malicious corruption), scalability (2–64 parties), arithmetic precision (64- and 32-bit fixed-point versus floating-point), and network type (LAN and WAN with up to 10 ms RTT). All experiments run on a single node with an Intel Xeon Gold 6242 CPU at 2.80 GHz with 32 cores. We model and execute experiments using FlexDC-Sim [1], an open-source data center DR simulator, with the data center configurations and workloads from prior work [2]. Each data center is driven by CPU- or GPU-based workload traces [1, 2].

5.1 Case Studies

The goal of this subsection is to verify that our secure protocol preserves the correctness of the plaintext power allocation. We measure this by tracking performance: the normalized distance between the power target and actual consumption, $|P_{\text{target}} - P_{\text{consumed}}|/R$, across five collaboration scenarios with 2 to 16 data centers. $\widetilde{\texttt{sfix}}$ denotes tailored fixed-point representation (Section 4.1) while NR denotes the division-free variant (Section 4.2). We group our arithmetic methods into two precision tiers: 1) The 64-bit tier ($\widetilde{\texttt{sfix}}_{64}$ and NR_{64}) matches the numerical precision of the plaintext IEEE fp64 baseline, producing identical power tracking with no increase in mean error; and 2) The 32-bit tier ($\widetilde{\texttt{sfix}}_{32}$ and NR_{32}) halves the bitwidth to reduce communication cost, at the expense of small rounding errors from the reduced fractional precision. We do not consider 16-bit arithmetic because it introduces notable deviations in power tracking, even in plaintext. Since MPC matches the requested bit precision exactly, this error is purely a numerical precision issue, not an MPC artifact. Overall, 32 bits is the lowest practical bitwidth for this application.

Figure 2 shows the power tracking accuracy of the secure protocol across all five scenarios. The top panel plots the aggregate power target against consumed

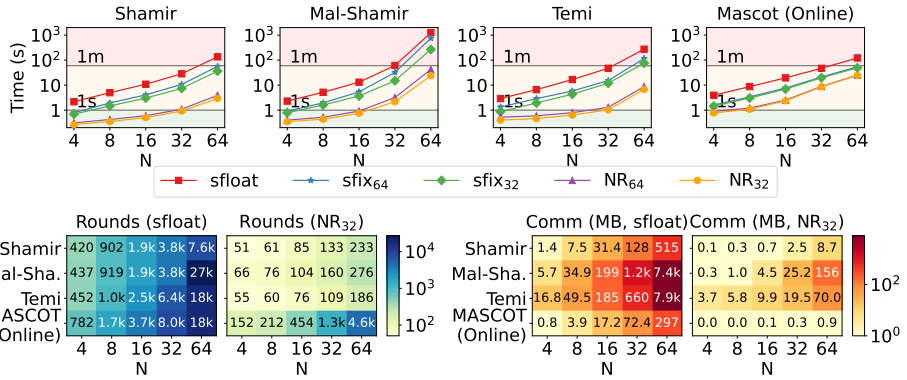


Fig. 3: Execution time (top) and communication cost (bottom) of the secure power-balancing protocol as a function of the number of data centers $N \in \{4, 8, 16, 32, 64\}$, across four security settings. Green and red bands mark sub-second and over-one-minute execution times.

power for $N = 16$ over one hour. The 64-bit tier produces outputs identical to the plaintext baseline and is omitted while 32-bit tier tracks the target with only minor deviations. These confirm that MPC adds zero accuracy penalty when the bitwidth preserves the full precision of the original computation. The bottom panel quantifies this across all N: The 64-bit tier matches the plaintext mean and 90th-percentile error exactly, while the 32-bit tier introduces only a small additional error that remains comparable to the plaintext baseline across all N, making it a viable choice when the communication savings from halved bitwidth (Section 5.2) outweigh the marginal accuracy cost.

5.2 Scalability Analysis

MPC protocols are communication-bound: latency is largely determined by the number of sequential communication rounds and the round-trip time (RTT) between parties (Section 3). To isolate the impact of these two factors, we benchmark every protocol–N–arithmetic combination listed in Table 1 under three network settings that reflect realistic data center interconnection scenarios: LAN with 1 ms RTT, representing co-located parties with minimal network delay; WAN with 5 ms RTT, representing geographically distributed parties within the same region/city; and WAN with 10 ms RTT, representing parties spread across distant regions. We simulate these conditions by injecting one-way latency into the MP-SPDZ networking layer: for a target RTT of r, each party receives incoming messages with a delay of r. Because protocol runtime scales roughly linearly with both the round count and the RTT, even modest reductions in the number of communication rounds translate into proportionally large time savings. This is precisely why the optimizations in Section 4 matter: replacing `sfloat` with `sfix` and then reducing the precision, and the subsequent move to the NR variant reduces both compute and communication costs.

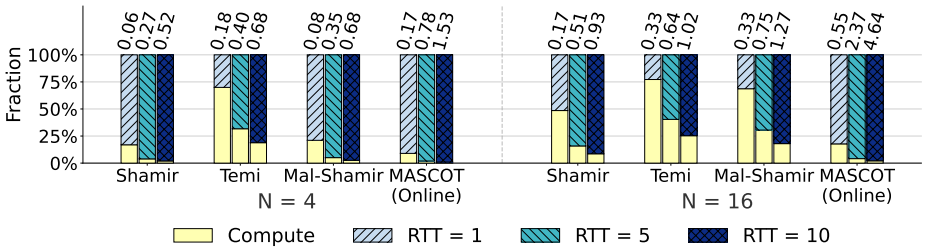


Fig. 4: The impact of network delays on total execution time (in seconds) for $N \in \{4, 16\}$ data centers using the NR_{32} variant. Bars are broken down by base computation time versus network latency penalties under LAN (RTT = 1 ms) and WAN (RTT = 5 and 10 ms) conditions. Numbers above each bar indicate total execution time in seconds. Network latency dominates total execution time.

We present our results of scalability benchmarks in Figure 3. The top row shows that each successive optimization reduces execution time up to an order of magnitude. For Shamir at $N = 16$, `sfloat` takes over 30 seconds, $\widetilde{\texttt{sfix}}_{64}$ brings this down to a few seconds, and NR_{64} completes in under 1 second. This pattern holds across all four protocols: the NR variants reach the sub-second region for up to $N = 16$ across all threat models. The bottom row explains where these savings come from. The round count heatmaps show that NR_{32} reduces rounds by 8 to 33 times compared to `sfloat` for honest majority protocols (e.g., Shamir drops from 1862 to 85 rounds at $N = 16$). Communication volume follows a similar pattern: Shamir at $N = 16$ transmits 31.4 MB under `sfloat` versus 0.7 MB under NR_{32}, a 45 times reduction. We note that the reported latency for protocols Shamir, Temi, and malicious Shamir correspond to the total (online + offline); however, our analysis of MASCOT focuses strictly on its online phase. This is because MASCOT (with no preprocessing optimizations) fails to meet the latency constraints of second-scale data exchange even at minimal network sizes (e.g., N=4). Furthermore, networks larger than N=16 result in memory exhaustion errors across several fixed-point precision configurations. However, Keller et al. [18] propose a series of optimizations that amortize the cost of MASCOT by reusing the cryptographic metadata created in the preprocessing phase. Thus, we assume that data centers can execute this offline phase prior to the actual computations [18].

To isolate the effect of network latency, we compared execution times across the three simulated environments. In a localized LAN setting (1 ms RTT), the baseline execution times remain naturally low; however, moving to geographically distributed WAN settings (5 ms and 10 ms RTT) fundamentally exacerbates the communication bottleneck. In these scenarios, the severe penalty of sequential communication exposes the fragility of traditional protocols. Consequently, the aggressive round reduction achieved by our NR variants transforms from a mere optimization into the primary driver of performance, ensuring total execution times remain strictly under the one-second threshold up to 5 ms network delays.

6 Conclusion

In this work, we present an MPC-based system for secure collaborative DCDR. By tailoring fixed-point representations and replacing secure divisions with a Chebyshev–Newton–Raphson reciprocal method, we reduce the communication rounds by up to 36× and execution time by up to 45×. Our evaluation shows that the optimized protocol satisfies the DCDR constraints: completing at sub-second latency for up to 16 parties under WAN conditions, with negligible impact on power tracking error. These results hold across protocol settings spanning semi-honest and malicious adversary models. While we use Conductor as a concrete case study, our optimizations apply to any power dispatch setting where participants share real-valued inputs and the allocation involves division, offering a practical path to privacy-preserving collaborative DR as data center coalitions continue to grow.

Acknowledgments. We thank the Rafik B. Hariri Institute for Computing and Computational Science and Engineering (Focused Research Program award #2024-07-003). We thank Connor Wagaman, Palak Jain, and Prof. Adam Smith for valuable discussions on privacy-preserving algorithms. This work has been partially funded by the National Science Foundation DESC program under Award No. 2450111.

Disclosure of Interests The authors have no competing interests to declare that are relevant to the content of this article.

LLM Usage Considerations We used LLMs to assist with text editing. All ideas, experiments, and results presented in this work are entirely the authors' own.

References

1. Acun, F., Hankendi, C., Levine, E., Reynolds, H., Bardwick, J., Coskun, A.K.: Investigating power consumption flexibility of ai data centers for demand response participation. In: Proceedings of the 17th ACM International Conference on Future and Sustainable Energy Systems. p. 5. E-Energy '26, ACM (2026)
2. Acun, F., Paschalidis, I.C., Coskun, A.K.: Conductor: A collaboration framework for multi-data-center demand response. In: 2024 IEEE 15th International Green and Sustainable Computing Conference (IGSC). pp. 22–28. IEEE (2024)
3. Aliasgari, M., Blanton, M., Zhang, Y., Steele, A.: Secure computation on floating point numbers. Network and Distributed System Security Symposium p. 405 (2013)
4. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Advances in Cryptology — CRYPTO '91. pp. 420–432. Springer (1991)
5. Catrina, O., Saxena, A.: Secure computation with fixed-point numbers. In: Financial Cryptography and Data Security. pp. 35–50. Springer (2010)
6. Chen, H., Zhang, Y., Caramanis, M.C., Coskun, A.K.: Energyqare: Qos-aware data center participation in smart grid regulation service reserve provision. ACM Trans. Model. Perform. Eval. Comput. Syst. **4**(1) (2019)
7. Chen, N., Ren, X., Ren, S., Wierman, A.: Greening multi-tenant data center demand response. SIGMETRICS Perform. Eval. Rev. **43**(2), 36–38 (2015)
8. Cramer, R., Damgård, I.B., Nielsen, J.B.: Secure Multiparty Computation and Secret Sharing. Cambridge University Press (2015)
9. Deokar, P., Arora, S.: Multiparty computation for privacy-preserving communication in the smart grid. In: Smart Trends in Computing and Communications. pp. 299–309. Springer Nature Singapore (2026)

10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proc. Third Conference on Theory of Cryptography. p. 265–284. Springer (2006)
11. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proc. 41st ACM Symposium on Theory of Computing. pp. 169–178 (2009)
12. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proc. 19th ACM Symposium on Theory of Computing. pp. 218–229 (1987)
13. Guo, Y., Li, H., Pan, M.: Colocation data center demand response using nash bargaining theory. IEEE Transactions on Smart Grid **9**(5), 4017–4026 (2018)
14. von der Heyden, J., Schlüter, N., Binfet, P., Asman, M., Zdrallek, M., Jager, T., Schulze Darup, M.: Privacy-preserving power flow analysis via secure multi-party computation. IEEE Trans. Smart Grid **16**(1), 344–355 (2025)
15. International Energy Agency (IEA): Energy and ai (2025), https://www.iea.org/reports/energy-and-ai, licence: CC BY 4.0.
16. Jahanshahi, A., Yu, N., Wong, D.: Powermorph: Qos-aware server power reshaping for data center regulation service. ACM Trans. Archit. Code Optim. **19**(3) (2022)
17. Keller, M.: Mp-spdz: A versatile framework for multi-party computation. In: Proc. ACM SIGSAC Conference on Computer and Communications Security (2020)
18. Keller, M., Orsini, E., Scholl, P.: Mascot: Faster malicious arithmetic secure computation with oblivious transfer. In: Proc. ACM SIGSAC Conference on Computer and Communications Security. p. 830–842 (2016)
19. Keller, M., Sun, K.: Secure quantized training for deep learning. In: International Conference on Machine Learning. pp. 10912–10938. PMLR (2022)
20. Liu, Z., Wierman, A., Chen, Y., Razon, B., Chen, N.: Data center demand response: avoiding the coincident peak via workload shifting and local generation. SIGMETRICS Perform. Eval. Rev. **41**(1), 341–342 (2013)
21. Mohassel, P., Zhang, Y.: Secureml: A system for scalable privacy-preserving machine learning. In: IEEE Symposium on Security and Privacy. pp. 19–38 (2017)
22. Mustafa, M.A., Cleemput, S., Aly, A., Abidin, A.: A secure and privacy-preserving protocol for smart metering operational data collection. IEEE Trans. on Smart Grid **10**(6), 6481–6490 (2019)
23. PJM Interconnection: PJM Manual 12: Balancing Operations, Revision 44. https://www.pjm.com/-/media/DotCom/documents/manuals/archive/m12/m12v44-balancing-operations-03-01-2022.pdf (2022)
24. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
25. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis. Springer (2002)
26. The Wall Street Journal: A new threat to power grids: Data centers unplugging at once (2026), https://www.wsj.com/business/energy-oil/a-new-threat-to-power-grids-data-centers-unplugging-at-once-741f1bda
27. Tian, N., Guo, Q., Sun, H., Zhou, X.: Fully privacy-preserving distributed optimization in power systems based on secret sharing. iEnergy **1**(3), 351–362 (2022)
28. Wang, B., Xu, L., Wang, J.: A privacy-preserving trading strategy for blockchain-based p2p electricity transactions. Applied Energy **335**, 120664 (2023)
29. Zhang, Y., Tsiligkaridis, A., Paschalidis, I.C., Coskun, A.K.: Data center and load aggregator coordination towards electricity demand response. Sustainable Computing: Informatics and Systems **42**, 100957 (2024)
30. Zhang, Y., Wilson, D.C., Paschalidis, I.C., Coskun, A.K.: Hpc data center participation in demand response: An adaptive policy with qos assurance. IEEE Transactions on Sustainable Computing **7**(1), 157–171 (2022)