PRIVADA: Private user-centric Data Aggregation

Betül Aşkın Özdemir*, Beyza Bozdemir, Ionut Groza**, and Melek Önen**
**COSIC, KU Leuven, Leuven, Belgium*
***Digital Security, EURECOM, Sophia Antipolis, France*
betul.askinozdemir@kuleuven.be, {beyza.bozdemir,ionut.groza,melek.onen}@eurecom.fr

Keywords: Data Aggregation, User Privacy, Multiple Data Customers, Secure Two-party Computation, SPDZ.

Abstract: In this work, we introduce PRIVADA, a maliciously secure data aggregation solution that uses MPC in the SPDZ framework. Unlike prior data aggregation schemes using MPC with/without SPDZ, PRIVADA supports multiple data customers while preventing inference of user participation and resisting collusions in real-world data aggregation applications. Moreover, our work guarantees *user privacy* and *result privacy*, in addition to *input privacy*. PRIVADA outperforms the state-of-the-art solutions by providing security against participating parties, including malicious data owners, aggregators, and data customers. Our proof-of-concept implementation also supports the new privacy-preserving data aggregation by combining malicious security, being available for multiple data customers, and ensuring strong privacy guarantees in large-scale deployments. The aggregation operation on the aggregator side becomes simpler with PRIVADA, and experimental results show a 12–15× speedup compared to the state-of-the-art. This confirms that malicious security and strong privacy guarantees can be achievable without sacrificing practicality.

1 INTRODUCTION

Data aggregation analytics are essential for functionality, safety, and decision-making in real-world use cases, but they introduce potential risks for individuals' privacy. Another problem emerges due to the data aggregation performed by cloud servers and/or multiple stakeholders that may attempt to exploit data during the data aggregation process. Moreover, with data protection regulations, the need for user and data privacy protection has become essential, and calls for the essential use of privacy enhancing technologies (PETs) to enable aggregation over individuals' data, without disclosing the underlying data. Therefore, protecting **user privacy** and **data privacy** conducted by one or more cloud servers in adversarial settings has become an immediate challenge.

A large number of privacy-preserving (pp) data aggregation solutions exist, leveraging various PET approaches, such as solutions based on multiparty computation (MPC) (Gehlhar et al., 2023; Damgård et al., 2016; Li et al., 2023) or solutions based on Federated Learning (FL) (Rathee et al., 2023; Ma et al., 2023; Guo et al., 2024; Jiang et al., 2025). Most of these works focus on simplified settings where the aggregation operation is not outsourced to an aggregator or where *only one single data customer* is interested in the aggregation result (Shi, 2022; Zhang et al., 2022; Davidson et al., 2022). The SOTA solutions that do consider **multiple data customers**, either involve a single aggregator or do not allow data owners to **control which data customers** can access the aggregation result, or they do not support **user anonymity**, or they do not address potential **collusions among parties** (Corrigan-Gibbs and Boneh, 2017; Sav et al., 2021). Furthermore, some solutions based on differential privacy (Bell et al., 2020; Bonawitz et al., 2017) require direct interaction among data owners, which is often undesirable in practice. A detailed study of prior work can be found in (Askin Özdemir et al., 2026).

To address these limitations: (i) non-existent or only one data customer; (ii) data control for data owners; (iii) potential collusion between parties. A recent solution, PRIDA is designed under the *honest-but-curious security model*. In contrast, the real-world deployment applications often require security against malicious adversaries. This limitation motivates our work, and hence, **PRIVADA** is designed in the malicious security model with the use of SPDZ (Cramer et al., 2018), guaranteeing all privacy properties covered by PRIDA.

We introduce PRIVADA, a secure aggregation protocol in a realistic aggregation setting, and demonstrate a proof-of-concept experimental benchmark by

achieving strong privacy guarantees in the malicious model without sacrificing practicality.

Our contributions are summarized as follows:

Malicious Security with Strong Privacy Guarantees. Our solution, namely PRIVADA, is a maliciously secure data aggregation protocol employing 2PC and leverages a SPDZ-based architecture that shifts heavy cryptographic operations to an offline preprocessing phase, and guarantees *input privacy*, *user privacy* (anonymity and selective disclosure), and *result privacy*, even when data owners (DOs), one aggregator, or data customers (DCs) behave maliciously, with the assumption that the two aggregators do not collude. PRIVADA is designed for realistic cloud deployments involving multiple data customers and considering potential collusions. The protocol prevents aggregators and DCs from inferring DOs' participation or accessing unauthorized results.

Support for Multiple Data Customers with Selective Disclosure. Unlike prior MPC-based (and/or SPDZ-based) data aggregation protocols, PRIVADA supports multiple DCs while enabling DOs to control their data use by deciding which DCs are authorized to access the aggregation result, thereby ensuring both user and result privacy.

Improved Aggregator Efficiency. Our experimental evaluation demonstrates a 12–15$\times$ speedup in aggregation runtime compared to PRIDA, while providing malicious security and strong privacy guarantees.

2 PROBLEM STATEMENT

In this section, we present challenges we have tackled in PRIVADA to propose a maliciously secure and private data aggregation using SPDZ.

Notation. Let $\mathrm{DO} = \{\mathrm{DO}_1, \ldots, \mathrm{DO}_n\}$ be a set of n *data owners* holding their input data dv and choice cv and $\mathrm{DC} = \{\mathrm{DC}_1, \ldots, \mathrm{DC}_m\}$ be a set of m *data customers* interested in the aggregation. The goal is to compute the aggregated result $s_j = \sum \mathrm{dv}_{ij}$, which is intended for authorized DC_j that the majority of data owners have chosen. The secure and private computation is realized in SPDZ, involving two aggregators, namely Agg_1 and Agg_2, that jointly perform the computation and provide outputs to the authorized data customers. Lastly, MAC values are computed in modulo 2^ℓ, and the shared values are represented with $\langle\rangle$.

2.1 Privacy goals

The **user**-centric pp-data aggregation based on MPC must satisfy these privacy properties:

Input Privacy ensures that the individual input data $(\mathrm{dv}, \mathrm{cv})$ contributed by DO_i remains confidential throughout the aggregation process. No party other than DO_i, including Agg, DCs, or other DOs, should be able to learn any information.

Anonymity of data owners should be preserved with respect to data customers and indirectly Agg_2. Their participation in the aggregation should remain confidential from data customers. Even if the intended aggregate result is revealed, DCs should not be able to identify which DOs have contributed their data to that particular aggregation process.

Selective Disclosure is an inherent right of DOs as their data control to decide how and for whom their data is used. This property is particularly important in multi-DC aggregation settings, where DOs can selectively authorize the use of their data for DCs.

Output Privacy guarantees that the aggregated output is disclosed only to authorized DCs. This property is essential in the presence of multiple DCs that increases the risk of information leakage.

2.2 Threat model

In this section, we outline the threat model of a pp-data aggregation serving multiple data customers under the malicious security model with two non-colluding aggregators, which is the standard SPDZ assumption.

PRIVADA involves DO_i holding private choice vectors **cv** and data vectors **dv**, which are sent to Agg_1 and Agg_2, computing outputs for authorized DCs. The protocol assumes malicious security and remains private and secure as long as at most one aggregator is corrupted.

We highlight that the submitted input should not be accessible in plaintext to either any Aggs or DCs; similarly, neither the Aggs nor the DOs should have plaintext access to the aggregated result. Also, we assume that PRIVADA steps execute through secure channels, which are secure against any external adversary who wants to compromise the transmission. When considering the potential attempts of existing parties in PRIVADA, such as colluding to gain unauthorized advantages, we can list potential adversarial parties as follows:

- *Data Owner*: A malicious DO may try to learn the inputs of other DOs or the aggregated output of DCs by colluding with Agg_1;

- *Data Customer*: A malicious DC may attempt to learn private inputs of DOs, access aggregation outputs intended for other DCs by colluding with Agg_2;
- *Aggregators*: A malicious aggregator may attempt to recover individual DO inputs, alter intermediate computations, provide incorrect aggregation results, or collude with certain DCs or DOs.

Additionally, we provide an authorization requirement in a malicious setting, whereby DC_j is authorized to receive an aggregated output only if at least t DOs have selected DC_j. A malicious DC_j may attempt to obtain the result even when the number of contributing DOs is below the threshold t, for example, by colluding with other parties or manipulating protocol messages.

3 PRELIMINARIES

In PRIVADA, we interchangeably use 2PC and MPC since MPC operations are carried out by two non-colluding aggregators. As we leverage SPDZ, we provide a brief presentation.

SPDZ. SPDZ (Damgård et al., 2012) is an actively secure MPC protocol that supports general secure computation and remains secure even if all but one of the participating parties are corrupted. The protocol relies on information-theoretic message authentication codes (MACs), enabling efficient verification of computations performed on secret-shared values. In other words, a value x is split into n secret shares x_i and assigned to each party P_i where $i \leq n$. Once shared, the value x becomes $x = \sum x_i$ with a MAC value αx, where α is a MAC key.

We particularly used SPDZ_{2^k}, which introduces a new additively homomorphic authentication scheme operating over $\mathbb{Z}_{2^k}$, which achieves efficiency comparable to standard approaches defined over fields. The main idea is to sample the MAC key α uniformly from $\mathbb{Z}_{2^s}$, where s is the security parameter, and compute the MAC value αx in $\mathbb{Z}_{2^{k+s}}$.

In standard SPDZ, each party acts as both server and client. In contrast, PRIVADA separates the roles of data owners, aggregation servers, and data customers, using the SPDZ-based input/output delivery protocols of Damgård et al. (Damgård et al., 2016). These protocols allow external users to securely provide inputs and receive outputs without joining the MPC computation among servers. In PRIVADA, MAC values are computed modulo 2^ℓ.

PRIDA. PRIDA (Bozdemir et al., 2024) is a pp-data aggregation that combines threshold homomorphic encryption (Th-FHE) (Asharov et al., 2012; Lopez-Alt et al., 2011) and secure two-party computation (2PC) (Beaver, 1991), while accounting for multiple data customers, ensuring data control for data owners, and evaluating potential collusion between parties. PRIDA is assumed to design realistic deployment scenarios, and provides strong privacy guarantees for DOs, who can participate anonymously and maintain full control over their data, a property that we name **user privacy**. The protocol inherently supports multiple DCs in a setting that has not been simultaneously addressed with user anonymity by existing state-of-the-art solutions. In addition to the by-default **input privacy** guarantee, PRIDA enforces **output privacy**, ensuring that aggregation results are revealed only to DCs, explicitly authorized by the DOs, and remain inaccessible to the aggregators and unauthorized data customers.

4 PRIVADA

We now present **PRIVADA**, achieving the privacy guarantees for data owners and data customers, namely: (i) user privacy, (ii) input privacy, and (iii) output privacy in the malicious model, while employing 2PC solely in the SPDZ framework.

PRIVADA assumes that the two aggregators, denoted by Agg_1 and Agg_2, do not collude. All other parties, including DOs and DCs, are assumed to be malicious. Security against the malicious behavior of the aggregators is ensured by SPDZ via authenticated shares and verification of multiplication triples.

PRIVADA depicted in Protocol 1 proceeds in three phases: (i) Setup, (ii) preliminary counting and aggregation, and (iii) output construction.

1. *Setup:* The aggregators retrieve authenticated random triples and random values from the SPDZ preprocessing. In particular, for each DO_i, authenticated triples $(\langle a\rangle, \langle b\rangle, \langle c\rangle)$ with $a = b \cdot c$ are used to verify the input. Here, b_{cv} and b_{dv} are random masking values from the triple used to hide the choice vector and data vector, respectively. Hence, each DO's input requires an authenticated triple of the choice vector and data vector. Once obtaining and verifying the triples, DO_i computes masked values by adding these random masks to its private inputs and broadcasting them. The aggregators then use affine reconstruction to obtain authenticated secret shares without learning the underlying data. In order to verify the output for each authorized DC, authenticated random values r and v are used during the construction phase.

Protocol 1 PRIVADA

Inputs. DO_i, $i \in \{1,\ldots,n\}$, inputs a choice vector $\mathbf{cv}_i$ and a data vector $\mathbf{dv}_i$. Also, a pre-defined threshold t is public.

Output. If $\mathrm{cv}_{total_j} \geq t$, DC_j obtains the aggregate result s_j, $j \in \{1,\ldots,m\}$. Otherwise, DC_j obtains nothing.

Protocol steps.

1. *Setup executed by Agg_1, Agg_2, and DO_i.*
   a. Agg_1, Agg_2: Retrieve authenticated triples for $\mathbf{cv}_i$ and $\mathbf{dv}_i$ from the pool: $(\langle b\rangle, \langle c\rangle, \langle a\rangle)$ s.t. $a = b.c$.
   b. Agg_1, Agg_2: Send $(\langle a\rangle, \langle b\rangle, \langle c\rangle)$ to DO_i.
   c. DO_i: Reconstruct a, b, c and verify $a = b.c$.
   d. DO_i: Compute masked vectors $\alpha_i = cv_i + b_{\mathrm{cv}}$ and $\beta_i = dv_i + b_{\mathrm{dv}}$.
   e. DO_i: Broadcast the plain values α_i, β_i to Agg_1, Agg_2.
   f. Agg_1, Agg_2: Recompute the shares as presented in affine combination (Cramer et al., 2018):
      - Agg_1: $\langle \mathrm{cv}_i\rangle_1 = \alpha_i - \langle b_{\mathrm{cv}}\rangle_1$ and $\langle \mathrm{dv}_i\rangle_1 = \beta_i - \langle b_{\mathrm{dv}}\rangle_1$.
      - Agg_2: computes: $\langle \mathrm{cv}_i\rangle_2 = \langle b_{\mathrm{cv}}\rangle_2$ and $\langle \mathrm{dv}_i\rangle_2 = \langle b_{\mathrm{dv}}\rangle_2$.
2. *Preliminary counting and Aggregation executed by Agg_1, Agg_2.*
   a. Agg_1, Agg_2: Compute $\langle \mathrm{cv}_j\rangle = \sum_{i=1}^{n} \langle \mathrm{cv}_{ij}\rangle$ for $j \in \{1,\ldots,m\}$.
   b. Agg_1, Agg_2: Exchange $\langle \mathrm{cv}_j\rangle$ to reconstruct cv_{total_j}.
   c. Agg_1, Agg_2: Compute $\langle \mathrm{s}_j\rangle = \sum_{i=1}^{n} \langle \mathrm{dv}_{ij}\rangle$ **if** $\mathrm{cv}_{total_j} \geq t$.
3. *Construction executed by DC_j.*
   a. Agg_1, Agg_2: Retrieve r, v and compute $\langle w\rangle = \langle \mathrm{s}_j.r\rangle$ and $\langle u\rangle = \langle v.r\rangle$.
   b. Agg_1, Agg_2: Send all shares to DC_j.
   c. DC_j: Reconstruct s_j, r, v, w, u and verify s_j.

2. *Preliminary counting and aggregation:* For each DC_j, $j \in \{1,\ldots,m\}$, Agg_1 and Agg_2 first check if cv_{ij} is 0 or 1 via $\mathrm{cv}_{ij}(1 - \mathrm{cv}_{ij}) = 0$. Then aggregators sum the shares of all choice vectors to compute the total count cv_j:

$$\mathrm{cv}_j = \sum_i \mathrm{cv}_{ij},$$

using secure addition on authenticated shares. They obtain the preliminary count and check whether it satisfies the threshold t. If $\mathrm{cv}_j \geq t$, they aggregate the corresponding data values by summing the shares of $\mathbf{dv}_{ij}$ to compute $\langle \mathrm{s}_j\rangle$

$$\langle \mathrm{s}_j\rangle = \sum_i \langle \mathrm{dv}_{ij}\rangle.$$

If $\mathrm{cv}_j < t$, the corresponding DC_j obtains no output.

3. *Output construction:* For each authorized DC_j, the aggregators retrieve an authenticated random value r and v, and locally adjust the output shares to unmask $\langle \mathrm{s}_j\rangle$. Each aggregator sends its share of $\langle \mathrm{s}_j\rangle$ to DC_j, who verifies correctness using the random numbers ($r, v, w,$ and u) and constructs s_j by taking the sum of shares $\mathrm{s}_j = \langle \mathrm{s}_j\rangle_1 + \langle \mathrm{s}_j\rangle_2$, if the verification is approved. Note that unauthorized DCs receive no shares and hence learn nothing.

5 SECURITY ANALYSIS

The protocol ensures privacy, correctness, and security against malicious adversaries. DOs' inputs remain private, only authorized DCs learn the aggregation result, and malicious behaviors cannot compromise correctness. Unlike SPDZ, participant roles in PRIVADA are separated: DOs provide inputs, DCs receive outputs, and two non-colluding aggregators perform the computation while preserving privacy.

We have proved the privacy and correctness of input data and output data in the extended version. We formally state the security guarantees of PRIVADA, and for the proofs, we refer the reader to (Askin Özdemir et al., 2026).

Theorem 1 (Security of PRIVADA)**.** *Assume that the two aggregators Agg_1 and Agg_2 do not collude. Then Protocol 1 securely realizes the ideal functionality $\mathcal{F}_{\mathsf{Agg}}$ for pp-data aggregation in the presence of malicious adversaries, in the $\mathcal{F}_{\mathsf{SPDZ}}$-hybrid model.*

6 PERFORMANCE EVALUATION

In this section, we present a proof-of-concept evaluation for our *user-centric* solution, PRIVADA, in a practical use case scenario, whereby each DO controls whether to participate and which DCs are authorized to access aggregated results derived from its data.

6.1 Asymptotic Analysis

We have worked on the asymptotic computational and communication costs of PRIVADA. Remark that the complexities of basic operations and representation are assumed to be negligible, i.e., $O(1)$.

As presented in Table 1, we compare PRIVADA with DP-based and MPC-based schemes such as SecAggML (Bonawitz et al., 2017), SecAgg (Bell et al., 2020), Prio (Corrigan-Gibbs and Boneh, 2017), Prio+ (Addanki et al., 2021), and PRIDA. Unlike existing solutions, PRIVADA avoids logarithmic and

quadratic growth in data owners' computation and achieves a more balanced complexity profile compared to (Bonawitz et al., 2017; Bell et al., 2020; Corrigan-Gibbs and Boneh, 2017; Addanki et al., 2021; Bozdemir et al., 2024). Compared to PRIDA, which relies on Th-FHE and 2PC under a semi-honest model, PRIVADA uses SPDZ-based 2PC under a stronger malicious adversarial model while maintaining the same asymptotic computational and communication complexity. Moreover, experiments in Table 2 show PRIVADA is 12–15× more efficient in practice while preserving equivalent privacy guarantees compared to PRIDA.

Table 1: Computational and communication complexity of PRIVADA and SoTA.

	SoTA	DO	DC	Agg_1	Agg_2
Computation	SegAggML	$\mathcal{O}(n^2+n\lambda)$	NA	$\mathcal{O}(n^2\lambda)$	NA
	SegAgg	$\mathcal{O}(\log^2 n+\lambda\log n)$	NA	$\mathcal{O}(n(\log^2 n+\lambda\log n))$	NA
	Prio	$\mathcal{O}(M\log M)$	NA	$\mathcal{O}(n(M\log M))$	NA
	Prio+	$\mathcal{O}(\ell)$	NA	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
	PRIDA	$\mathcal{O}(m)$	$\mathcal{O}(1)$	$\mathcal{O}(nm)$	$\mathcal{O}(nm)$
	PRIVADA	$\mathcal{O}(m)$	$\mathcal{O}(1)$	$\mathcal{O}(nm)$	$\mathcal{O}(nm)$
Communication	SegAggML	$\mathcal{O}(n+\lambda)$	NA	$\mathcal{O}(n^2+\lambda n)$	NA
	SegAgg	$\mathcal{O}(\log^2 n+\lambda)$	NA	$\mathcal{O}(n(\log^2 n+\lambda))$	NA
	Prio	NA	NA	$\mathcal{O}(n)$	NA
	Prio+	$\mathcal{O}(\ell)$	NA	$\mathcal{O}(n+\ell^2)$	$\mathcal{O}(n+\ell^2)$
	PRIDA	$\mathcal{O}(m)$	$\mathcal{O}(1)$	$\mathcal{O}(nm)$	$\mathcal{O}(nm)$
	PRIVADA	$\mathcal{O}(m)$	$\mathcal{O}(1)$	$\mathcal{O}(nm)$	$\mathcal{O}(nm)$

NA: Not Applicable, n: #DOs, ℓ: bit size, λ : size of the vector, M: #Multiplications

Overall, compared to SecAgg, SecAggML, Prio, and Prio+, PRIVADA achieves comparable or improved asymptotic complexity while additionally ensuring anonymity, selective disclosure, and collusion resistance, making it suitable for large-scale settings where typically $n > \ell$ and $M > \ell$. For details, we refer the reader to (Askin Özdemir et al., 2026).

6.2 Comparison with the SoTA

We implemented PRIVADA as a proof-of-concept using the SPDZ_{2^k} framework via MP-SPDZ (Keller, 2020) in a desktop environment with an Intel i7-12700H 2.30 GHz processor, 20 cores, and 8 GB RAM. The arithmetic is performed in modulo 2^{64} and a 64-bit security parameter, where the 2PC values are $\ell = 128$-bit.

PRIVADA preserves the privacy goals of PRIDA while strengthening the security model against malicious adversaries. Unlike PRIDA, which relies on threshold homomorphic encryption and 2PC, PRIVADA uses an SPDZ-based output delivery protocol (Damgård et al., 2016), mainly for efficiency gains. In PRIDA, aggregating data from 2500 DOs takes about 60 seconds, whereas PRIVADA reduces this to roughly 10 seconds (over 80%improvement). Similarly, the approach in (Damgård et al., 2016) requires about 25 seconds for 2500 users but assumes a more centralized trust setting where one party holds plaintext data and broadcasts results to aggregators. In contrast, PRIVADA achieves stronger guarantees of user privacy and data privacy, supporting multiple data customers.

Table 2: Runtime comparison with PRIDA (in seconds).

Protocols	Data Owner	Data Customer	Aggregation
PRIDA with Th-BFV	0.009	0.22	1.48+1.51
PRIDA with Th-CKKS	0.009	0.34	1.17+1.22
PRIVADA	5e−5	3e−5	**0.102+0.102**

Table 2[1] presents a runtime comparison between PRIVADA and PRIDA in a setting with 100 DOs and 1 DC. Although both systems aim to achieve user-centric privacy, they rely on different PET constructions and security assumptions: PRIDA combines Th-FHE in BFV and CKKS with 2PC under a semi-honest adversarial model, whereas PRIVADA employs SPDZ-based 2PC to achieve security against malicious adversaries.

Typically, upgrading from semi-honest to malicious security introduces significant computational overhead. However, PRIVADA avoids this cost by eliminating expensive FHE ciphertext operations and instead relying on SPDZ, which shifts most cryptographic complexity to an offline, data-independent preprocessing phase. Consequently, the online phase becomes lightweight, involving only secret-sharing operations, modular arithmetic, and MAC-based checks.

This design leads to efficiency gains across all parties. For DO, runtime is reduced to 5×10^{-5} seconds, representing an approximately 180-fold improvement over PRIDA (0.009 seconds). For DC, the result reconstruction and verification take only 3×10^{-5} seconds, achieving roughly 7K–11K× speedup compared to PRIDA. At the system level, the aggregation phase completes in 0.204 seconds, yielding about 12–15× improvement over PRIDA.

We have also discussed optimizations and future work in (Askin Özdemir et al., 2026).

[1] https://github.com/gionut/PRIVADA

7 CONCLUSION

We propose PRIVADA, a user-centric private aggregation protocol based on SPDZ that supports multiple DCs and allows DOs to control which DCs can access aggregated results through two non-colluding aggregators. PRIVADA guarantees user, data, and output privacy, achieves security against malicious adversaries, and is 12–15× more efficient than PRIDA and other state-of-the-art solutions supporting similar privacy properties.

Acknowledgements This work was supported by CyberSecurity Research Flanders with reference number VR20192203. In addition, this work was partially supported by the Research Council KU Leuven, C16/18/004, through the IF/C1 on New Block Cipher Structures and by the French government, through the 3IA Côte d'Azur Investments in the project managed by the National Research Agency (ANR) with reference number ANR-23-IACL-0001.

REFERENCES

Addanki, R., Corrigan-Gibbs, H., Meiklejohn, S., and Weaver, N. (2021). Prio+: Privacy preserving aggregate statistics via boolean shares. In *S&P*.

Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., and Wichs, D. (2012). Multiparty computation with low communication, computation and interaction via threshold fhe. In *EUROCRYPT*.

Askin Özdemir, B., Bozdemir, B., Groza, I., and Önen, M. (2026). PRIVADA: Private user-centric data aggregation. In *ePrint, 2026/579*.

Beaver, D. (1991). Efficient multiparty protocols using circuit randomization. In *CRYPTO*.

Bell, J., Bonawitz, K. A., Gascón, A., Lepoint, T., and Raykova, M. (2020). Secure single-server aggregation with (poly)logarithmic overhead. In *CCS*.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *CCS*.

Bozdemir, B., Askin Özdemir, B., and Önen, M. (2024). PRIDA: PRIvacy-preserving Data Aggregation with multiple data customers. In *IFIP SEC*.

Corrigan-Gibbs, H. and Boneh, D. (2017). Prio: Private, robust, and scalable computation of aggregate statistics. In *NSDI*.

Cramer, R., Damgård, I., Escudero, D. E., Scholl, P., and Xing, C. (2018). SPDZ2k: Efficient MPC mod 2k for Dishonest Majority. In *ePrint*.

Damgård, I., Damgård, K., Nielsen, K., Nordholt, P. S., and Toft, T. (2016). Confidential benchmarking based on multiparty computation. In *FC*.

Damgård, I., Pastro, V., Smart, N. P., and Zakarias, S. (2012). Multiparty computation from somewhat homomorphic encryption. In *CRYPTO*.

Davidson, S., Agrawal, N., and Karri, R. (2022). Efficient secure aggregation for federated learning. In *CCS*.

Gehlhar, T., Marx, F., Schneider, T., Suresh, A., Wehrle, T., and Yalame, H. (2023). SAFEFL: MPC-friendly framework for private and robust federated learning. In *SPW*.

Guo, Y., Polychroniadou, A., Shi, E., Byrd, D., and Balch, T. (2024). MicroSecAgg: Streamlined single-server secure aggregation. In *POPETs*.

Jiang, Y., Zarezadeh, M., Dai, T., and Köpsell, S. (2025). Alphafl: Secure aggregation with malicious2 security for federated learning against dishonest majority. In *POPETs*.

Keller, M. (2020). MP-SPDZ: A versatile framework for multi-party computation. In *CCS*.

Li, H., Lin, H., Polychroniadou, A., and Tessaro, S. (2023). Lerna: Secure single-server aggregation and key-homomorphic masking. In *ASIACRYPT*.

Lopez-Alt, A., Tromer, E., and Vaikuntanathan, V. (2011). Cloud-assisted multiparty computation from fully homomorphic encryption. In *ePrint, 2011/663*.

Ma, Y., Woods, J., Angel, S., Polychroniadou, A., and Rabin, T. (2023). Flamingo: Multi-Round Single-Server Secure Aggregation with Applications to Private Federated Learning . In *S&P*.

Rathee, M., Shen, C., Wagh, S., and Popa, R. A. (2023). Elsa: Secure aggregation for federated learning with malicious actors. In *S&P*.

Sav, S., Scherf, G., and Schneider, T. (2021). Lightweight and scalable secure aggregation for federated learning. In *AsiaCCS*.

Shi, E. (2022). Foundations of differentially private machine learning. In *STOC*.

Zhang, R., Gaboardi, M., and Smith, A. (2022). Improved secure aggregation for federated learning. In *USENIX*.