RESEARCH-ARTICLE

Byzantine-Resilient Federated Computation of Differentially Private Summary Statistics

GIULIO SEGALINI, University of Neuchâtel, Neuchatel, NE, Switzerland

MARIA FERNANDES, Novo Nordisk Foundation Center for Basic Metabolic Research, Copenhagen, Hovedstaden, Denmark

JÉRÉMIE DECOUCHANT, Delft University of Technology, Delft, Zuid-Holland, Netherlands

Open Access Support provided by:

University of Neuchâtel

Novo Nordisk Foundation Center for Basic Metabolic Research

Delft University of Technology

Published: 15 December 2025

Citation in BibTeX format

Middleware '25: 26th International Middleware Conference
December 15 - 19, 2025
TN, Nashville, USA

https://doi.org/10.1145/3721462.3770766
ISBN: 9798400715549

Byzantine-Resilient Federated Computation of Differentially Private Summary Statistics

Giulio Segalini
giulio.segalini@unine.ch
Université de Neuchâtel
Neuchâtel, Switzerland
Delft University of Technology
Delft, the Netherlands

Maria Fernandes
maria.fernandes@sund.ku.dk
Novo Nordisk Foundation Center for Basic Metabolic Research, University of Copenhagen
Copenhagen, Denmark
LASIGE, Universidade de Lisboa
Lisboa, Portugal

Jérémie Decouchant
j.decouchant@tudelft.nl
Delft University of Technology
Delft, the Netherlands

ABSTRACT

Summary statistics are essential to analyse large datasets in various fields, including financial and medical research. Federated computations enhance statistical power by combining geo-distributed datasets while ensuring compliance with data protection regulations, privacy guarantees, and resilience against intrusions. We present Tides, a federated framework leveraging Trusted Execution Environments (TEEs) to defend against adversaries controlling up to f of the N datacenters. We present an instantiation of Tides using genomic (GWAS) statistics. We address TEE-specific attack vectors, including communication blocking and side-channel attacks. Tides follows the following three key steps: (1) TEEs share statistical results through reliable broadcast and run a randomized crash-tolerant binary consensus algorithm to identify the datasets that are available; (2) TEEs enforce differential privacy with ad hoc noise; and (3) TEEs run memory-oblivious algorithms to compute the final summary statistics. We implemented Tides with Intel SGX enclaves and demonstrated its practicality with three datasets.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; **Privacy-preserving protocols**; • **Computing methodologies → Distributed algorithms**.

ACM Reference Format:
Giulio Segalini, Maria Fernandes, and Jérémie Decouchant. 2025. Byzantine-Resilient Federated Computation of Differentially Private Summary Statistics. In *26th International Middleware Conference (MIDDLEWARE '25), December 15–19, 2025, Nashville, TN, USA*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3721462.3770766

1 INTRODUCTION

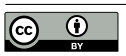
Data digitization and technological advances have led to the availability of large-scale datasets. Analysis of such datasets has become increasingly valuable in various fields, such as financial and medical research, allowing market study and risk assessment, population health studies, and personalized treatment approaches, to name a few applications [56]. Summary statistics play a crucial role in extracting actionable insights from these datasets, allowing the efficient identification of patterns, trends, and correlations. However, data is often geo-distributed, i.e., it is collected and stored in multiple datacenters around the world, and cannot be centralized due to regulatory, logistic or privacy concerns [46]. In the last decade, methods have been described to improve privacy guarantees [78] or accuracy [63] of federated computations.

However, it is still an open challenge to efficiently compute and release summary statistics with strong privacy guarantees in a federated environment that include colluding and malicious datacenters. The adversary that controls those malicious datacenters can attempt to obtain private data from the final released statistics, but also from observing the data manipulated and the computations executed by the datacenters it controls. Furthermore, malicious datacenters can also generate arbitrary messages and avoid sending or processing messages as expected [29], which might prevent summary statistics from being computed and eventually released. Although existing research has addressed certain aspects of these threats and requirements, no current solution effectively tackles all of them simultaneously while maintaining practical performance. For example, potential solutions that would use zero-knowledge proofs would have a low performance. Consequently, designing a federated computational framework that provides strong privacy, security, and robustness without compromising accuracy or scalability remains an open and complex problem.

In this paper, we address this problem and propose Tides, the first distributed framework for statistics computation that enforces differential privacy under the f-out-of-N malicious threat model. Tides makes the following key **contributions**:

• **A secure and resilient framework for distributed statistics computation:** We introduce Tides, the first distributed system for privacy-preserving statistics computation under $f \leq \lfloor (N-1)/2 \rfloor$ malicious thread model. Tides combines reliable broadcast, crash-tolerant consensus (adapted from the Ben-Or algorithm), and a pull-based protocol to enable collaboration between TEEs (Intel SGX enclaves). These TEEs exchange encrypted intermediate results only and agree on which datasets to use and their location before computing the final statistics.

• **Optimized differential privacy under f malicious participants setting:** Tides minimizes the amount of noise necessary to enforce differential privacy, even when up to f-out-of-N datacenters are malicious. Unlike previous work, noise is added after



MIDDLEWARE '25, December 15–19, 2025, Nashville, TN, USA

ACM ISBN 979-8-4007-1554-9/2025/12
https://doi.org/10.1145/3721462.3770766

consensus is reached on the datasets to use. We analytically determine the minimum necessary noise based on the worst-case data exposure across f corrupted datacenter.

• **Robust protection against TEEs side-channel attacks:** To mitigate known side-channel vulnerabilities of TEEs, Tides employs memory oblivious algorithms during the statistical computations. After consensus on the datasets to use, f+1 TEEs securely gather the required datasets, in our case, summary statistics of genome-wide association studies (GWAS), and compute the desired downstream statistics using memory oblivious algorithms before releasing the top-K results. We evaluate two practical methods for selecting top-K results—bubbling and full sorting—and demonstrate that side-channel protections do not compromise performance. For completeness, we compare the computed noise level with that of centralized DP, which only considers external adversaries (i.e., $f = 0$). Despite the noise added, Tides maintains high utility: we show that the KL-divergence between the true and perturbed statistics remains below 1 when $f \leq \lfloor (N-1)/2 \rfloor$. Furthermore, since Tides only exchanges intermediary results (e.g., contingency tables), both network and memory usage remain low.

• **Comprehensive evaluation and real-world biomedical application:** We apply Tides to compute GWAS summary statistics using 3-value genotype encodings – offering more precision than the commonly used 2-value encodings. We establish the sensitivity bounds for pairwise allele frequencies and linkage disequilibrium statistics. Our experiments evaluate performance, utility, noise impact (via KL-divergence), and false positive rates, confirming Tides 's practicality under both honest and faulty datacenter scenarios.

This paper is organized as follows. Sec. 2 discusses the related work. Sec. 3 provides necessary background on genomic data and the foundations of GWAS. Sec. 4 presents our system and threat models. Sec. 5 describes the distributed algorithms that Tides uses to agree on the datasets to use for the GWAS and their location. Sec. 6 establishes the sensitivity of GWAS metrics, i.e., allele frequencies, χ^2 and linkage disequilibrium, in presence of an f-out-of-N malicious adversary, which is necessary to tailor the DP noise level. Sec. 7 describes our memory-oblivious algorithms that TEEs employ to compute the GWAS metrics. Sec. 8 presents our performance evaluation. Finally, Sec. 9 concludes this paper.

2 RELATED WORK

Privacy-preserving statistics. Processing sensitive data is a cross-field challenge, including financial and medical research, highlighting the need for private and secure methods for summary statistics computation. Various techniques have been used to ensure privacy. Differential privacy (DP) has been applied to summary statistics. For example, Smith et al. [65] showed that private estimators can be nearly as accurate as unprotected results. Jawurek et al. [40] developed a fault-tolerant distributed approach using homomorphic encryption (HE) to enable secure DP data exchange with an untrusted party. Secure multiparty computation (SMC) techniques, combined with oblivious transfers, allow privacy-preserving data aggregation [27], statistics computation without revealing individual inputs [11], and privacy-preserving exact regression fitting [62]. Other works applied multiparty computations and local DP to perform computations of key-value data, however the nodes were assumed semi-honest and no crash-resistance was guaranteed [38].

We now focus on privacy and security-oriented methods for GWAS computations as this is our case study. Table 1 summarizes our analysis of the most recent and relevant federated GWAS frameworks, and shows that Tides is the first to enforce privacy against a malicious adversary that controls a limited number f of datacenters.

HE or SMC-based GWAS computations. Several works have used SMC [22, 23, 37, 69] or HE [10, 75, 82] to compute GWAS statistics with privacy. In practice, these methods inherently required present higher communications complexity than Trusted Execution Environments (TEEs), which run confidential code over private data at close to native performance. In addition, tolerating malicious participants, e.g., using methods such as zero-knowledge proofs [18, 85] or delayed consistency checks, would further decrease the performance of HE or SMC-based methods.

Privacy-preserving GWAS releases. DP has been shown to provide a high level of security against individual membership attacks and provide high accuracy for large datasets [32]. Naturally, several works have used DP in genomic or medical contexts [4, 33, 74]. Among them, Tramer et al. [70] increased the utility of GWAS studies conducted with DP by using stronger assumptions on the adversary's prior knowledge. Yu et al. proposed two systems that leverage DP to perform GWAS, one that computes linear regression problem [81] and one for the χ^2 test [80]. Yamamoto et al. [77] compute χ^2 statistics and p-values. Pascoal et al. [57, 58] proposed dynamic privacy-preserving federated GWAS. Yilmaz et al. [79] use local DP to prevent privacy attacks that would leverage genomic data correlations. Jiang et al. [41] described leveraging of DP to share genomic datasets. Our framework also relies on DP and is the first to leverage the assumption that at most f datacenters are controlled by the adversary to decrease the amount of noise applied.

TEE-based GWAS computations. Carpov and Tortech [16] described a method to compute the χ^2 statistic using Intel SGX enclaves. Mandal et al. [48] integrated memory oblivious communications in [16]. PRINCESS [20] performs GWAS tests within SGX enclaves for rare-disease collaboration studies, while PRESAGE [19] applies encoding and indexing methods on genomic data to answer private queries with high-performance. Asvadishirehjini et al. [3] presented a framework for genomic data analysis using TEEs and memory oblivious computations. Recently, Rosenblum et al. [61] proposed confidential population-scale GWAS, enabling joint genetic analysis relying on a centralized Intel SGX-based cloud computations. However, TEEs introduce specific attack vectors and computation limitations (See Sec. 2 for TEE limitations). They can be subject to Denial-of-Service (DoS) attacks [21, 71], where an adversary prevents them from interacting with the network, or side-channel attacks [34, 53]. One way to eliminate possible side-channel attacks is to use memory oblivious algorithms [35]. Our work, Tides, is the first GWAS framework that defends against DoS, side-channel, and privacy attacks against a malicious adversary. Tides's TEE is implemented using Intel SGX.

TEE-based Asynchronous Common Subset (ACS). Correct datacenters in Tides rely on their TEE to solve the ACS problem, i.e., to agree on a set of datasets to use to compute statistics on. To do so, they replicate their intermediary statistics among other TEEs

Table 1: Overview of GWAS frameworks.

	Method	Mem. oblivious	f mal. adversary	DP	SAF	PAF	χ^2	LD
[33, 70, 74, 77, 80, 81]	N/A	N/A	✗	✓	✓	✓	✓	✗
[10, 75, 82]	HE	N/A	✗	✗	✓	✓	✓	✗
[22, 23, 37, 69]	SMC	N/A	✗	✗	✓	✓	✓	✗
[3]	TEE	✓	✗	✗	✓	✓	✓	✗
[57, 58]	TEE	✗	✗	✗	✓	✓	✓	✓
Tides (this work)	TEE	✓	✓	✓	✓	✓	✓	✓

Mem. oblivious: memory oblivious; **mal. adversary**: malicious adversary; **DP**: differential privacy; **SAF**: single allele frequencies; **PAF**: pairwise allele frequencies; **LD**: Linkage disequilibrium; **HE**: homomorphic encryption; **SMC**: secure multiparty computations; **TEE**: trusted execution environment; **N/A**: not applied

and subsequently participate in crash-tolerant binary consensus instances, one per datacenter in the system. Together, these consensus instances identify a subset of the intermediary statistics that are stored in at least $f+1$ TEEs and therefore can eventually be used to compute and release statistics. This approach follows Ben-Or et al.'s traditional asynchronous common subset framework [8], which consists in a reliable broadcast component and an asynchronous binary consensus components. This construction of ACS is not the only possible one, but it has been used in several impactful and recent asynchronous BFT protocols such as HoneyBadgerBFT [51], BEAT [28], EPIC [47], Dumbo [36] and DBFT [25].

In a system of $3f+1$ processes, Ben-or et al.'s ACS can be implemented using Bracha's reliable broadcast [13] and Ben-Or's binary consensus [7]. Bracha's reliable broadcast [13] requires $n = 3f+1$ processes to tolerate f Byzantine processes and assumes an asynchronous fully-connected network. It consists in three communication phases (Send, Echo and Ready), which explains why it is sometimes referred to as double echo authenticated broadcast. Overall, Bracha's algorithm generates $2n^2 - n - 1$ messages. The literature contains many reliable broadcast algorithms that have improved over Bracha's, e.g., ensuring a lower network consumption [15] or latency [24, 39], real-time properties [43, 44], or wider applicability [12].

Tides considers a system of $2f+1$ datacenters equipped with TEEs, which allows the ACS abstraction to be solved more efficiently. In particular, it has been shown that reliable broadcast can be implemented with $2f+1$ processes that are equipped with trusted components [24, 83, 84]. We use Zhao et al. [83]'s reliable broadcast, which requires two messages types (Send and Echo). In this protocol and in a nutshell, upon receiving a Send or an Echo message, a TEE forward the value it contains to all other TEEs, and delivers a value when it has received it $f+1$ times. The binary consensus instances of Tides use Ben-Or's algorithm [7], which can use $2f+1$ processes to tolerate f crash faults [1]. We use Ben-Or's algorithm because it is immediate to adapt it to our system model (i.e., with TEEs). Note that more recent and efficient binary consensus algorithms [25, 28, 36, 47, 51] all consider the Byzantine fault model. We believe that they could be modified to operate with $2f+1$ processes and tolerate crashes instead of Byzantine faults, but such modifications would have to be demonstrated and we refrain from doing so here for space reasons.

Intel SGX limitations. Intel SGX enclaves offer a hardware isolated executing environment that enhances the confidentiality and integrity of computations. However, they present some limitations. First, SGX enclaves have a limited memory size, allowing only 96 MB for the execution of applications within the enclave. With the introduction of SGX2, support for dynamic memory management and paging was introduced, allowing enclaves to expand their practical memory space at runtime [50]. This limitation may be mitigated in the future, as some processors already support enclaves with significantly larger memory capacities, up to 512 GB in some cases [45]. Despite their benefits, enclaves are vulnerable to a range of attacks.

Numerous studies have shown that SGX enclaves are susceptible to **side-channel attacks** that exploit variations in cache access patterns, and page-fault timing to infer sensitive data from within enclaves [14]. Several countermeasures have been proposed. One prominent approach is the design of memory-oblivious algorithms, which ensure that memory access patterns remain consistent and independent of the input data [48]. This strategy makes different runs of the algorithm indistinguishable to resources outside the enclave, thus mitigating information leakage through side-channel attacks [2]. Additionally, ad-hoc solutions include data encoding techniques that fit data into fixed-size blocks (e.g., 4 KB page-sized blocks) [19] or processing a limited number of entries at a time [20] to thwart paging attacks. Tides employs memory-oblivious algorithms to defend against side-channel attacks.

Furthermore, enclaves are also susceptible to Denial-of-Service (DoS) attacks, where an adversarial host may stall or terminate the enclave execution [21, 71]. To address this, Tides uses fault-tolerance protocols and memory-oblivious algorithms. Another critical threat are rollback attacks, where the enclave would lose their internal state, potentially compromising integrity. While Tides does not directly address rollback attacks, well-establish approaches exist [49, 54, 76]. Additionally, restrictive responsiveness, where users cannot verity the enclave's liveness or state, can be addressed by transferring certificates to clients. Rollback attacks can therefore be addressed by known defense solutions [49, 54], and restrictive responsiveness can be addressed by transferring certificates to clients. A possible way to maintain parallelism is to rely on parallel executions [6, 66, 67].

While TEEs offer strong hardware isolation improving computations confidentiality and integrity, their security guarantees

are not absolute. Recent surveys highlighted the need for further systems that account for the range of threats of SGX and similar technologies [17, 55, 64]. To address the above vulnerabilities, Tides combines multiple layers of defence. Specifically, it uses memory-oblivious computations that ensure fixed data-independent memory access patterns. This eliminated the variations that side-channel attacks rely on. In addition,Tides applies differential privacy to the outputs from the computations, in order to provide formal privacy guarantees even in the presence of external knowledge. Together, these mechanisms ensure that adversaries observing memory and communication patterns cannot infer private information about genetic data or statistical results.

3 BACKGROUND: GENOMIC DATA AND GWAS

Genome-Wide Association Studies (GWAS) are mathematical approaches used to identify genetic variants associated with diseases. They consist of the analysis of genetic variants in large datasets of individuals and a comparison of their traits (i.e., disease status). We are aware of adjustment for covariates and take into account higher-order correlations between SNPs. However, those are outside the scope of Tides implementation.

In this section, we introduce the genotype representation in our framework, 3-value encoding. For a detailed explanation of how our federated framework, Tides, computes and releases privately single-wise and pair-wise allele frequencies, χ^2 statistics and LD, we refer to Sec. 3.2, which provides both computational details and privacy guarantees.

3.1 Genotyping data encoding

Given that most genotypes (i.e., an individual's set of genes) share significant similarities, we can represent a genotype pointing its differences when compared to a reference genome, what we call variants. Single nucleotide polymorphisms define variants where a single nucleotide differs and are the most common type. Therefore, Tides considered only SNPs for the described computations. Referring to SNPs on a population, for each variant position we have a nucleotide shown by the majority of the individuals (*major allele* or reference allele) and one or multiple different nucleotide presented by a small number of individuals (*minor allele* or alternative allele, usually with frequency $< 5\%$) [68]).

The 3-value encoding (used by Tides) represents the three possible genotypes ar each variant position with 0, 1, or 2. These numbers correspond to the count of minor alleles present in the genotype. Specifically: *0* represents individuals with two copies of the reference allele, *1* represents individuals with one copy of the reference and one copy of the alternative allele, and *2* represents individuals with two copies of the alternative allele. Tab. 2 illustrates an example of N genotypes {g1, …, gN} that consider L variants {SNP1, …, SNPL}.

3.2 GWAS statistics

We recall the definitions of the GWAS statistics Tides computes [5]. These metrics allow the identification of the top alleles whose frequencies differ the most between the case and control groups.

Table 2: GWAS genome 3-value encoding.

	SNP_1	SNP_2	…	SNP_L	Phenotype
Genome g_1	0	2		1	Case
Genome g_2	1	0		1	Control
⋮					
Genome g_N	0	2		0	Case

Minor allele frequency (MAF):. is the proportion of individuals in a population that carry the less common allele. Variants with a low MAF (e.g., less than 0.01) are considered rare, while variants with an high MAF (e.g., greater than 0.05) are more common.

Single allele frequency: represents the proportion of a specific allele (e.g., minor allele) in a population.

Pairwise allele frequency: describes the joint distribution of alleles at two different SNP positions.

χ^2 ***statistic:*** is a test statistic computed to determine if there is significant association between alleles at a specific position and a phenotype (e.g., diseases). Eq. 1 shows the formula used for computing the χ^2 metric [72], where exp and obs refer to expected and observed SNP counts, respectively.

$$\chi^2 = \frac{(obs_{SNP_1} - exp_{SNP_1})^2}{exp_{SNP_1}} + \frac{(obs_{SNP_2} - exp_{SNP_2})^2}{exp_{SNP_2}} \quad (1)$$

Linkage Disequilibrium: is a test statistic to measure the non-independence of alleles at different sites [60]. In the context of GWAS this measure is performed over a pair of SNPs and it can be computed as shown in Equation 2, where C_{XY} refers to the count of genotypes presenting value X for the first SNP and Y for the second and G is the total number of genotypes on which the analysis is performed.

$$LD = \frac{C_{00}}{2G} - \left(\frac{C_{0-}}{C_{0-} + C_{1-}} \cdot \frac{C_{-0}}{C_{-0} + C_{-1}} \right) \quad (2)$$

To compute the GWAS statistics described in this section, we need to compute the allele contingency tables, which summarize the counts of the possible alleles per study group (case and control). Table 3 illustrates a contingency table for a single SNP and a certain phenotype p. N_i^{pop} is the count of major/minor alleles $i \in \{M, m\}$ in the case/control population $pop \in \{\text{case}, \text{control}\}$. N^{case} and $N^{control}$ are the size of the case and the control population, respectively. N^{g0}, N^{g1} and N^{g2} are the number of individuals with a given genotype, i.e., 0, 1, or 2, respectively.

Table 3: A single contingency table for phenotype p.

	Genotype			
	0	1	2	Total
Case	C_0^{case}	C_1^{case}	C_2^{case}	N^{case}
Control	$C_0^{control}$	$C_0^{control}$	$C_2^{control}$	$N^{control}$
Total	N^{g0}	N^{g1}	N^{g2}	

Table 4: A GWAS pairwise contingency table for two variants, SNP_i and SNP_j, where $i, j \in \{1, \cdots, L\}$.

	Phenotype$_p$ = case				Phenotype$_p$ = control			
	SNP_j				SNP_j			
SNP_i	0	1	2	Total	0	1	2	Total
0	C_{00}^{case}	C_{01}^{case}	C_{02}^{case}	C_{0-}^{case}	$C_{00}^{control}$	$C_{01}^{control}$	$C_{02}^{control}$	$C_{0-}^{control}$
1	C_{10}^{case}	C_{11}^{case}	C_{12}^{case}	C_{1-}^{case}	$C_{10}^{control}$	$C_{11}^{control}$	$C_{12}^{control}$	$C_{1-}^{control}$
2	C_{20}^{case}	C_{21}^{case}	C_{22}^{case}	C_{2-}^{case}	$C_{20}^{control}$	$C_{21}^{control}$	$C_{22}^{control}$	$C_{2-}^{control}$
Total	C_{-0}^{case}	C_{-1}^{case}	C_{-2}^{case}	$2N^{case}$	$C_{-0}^{control}$	$C_{-1}^{control}$	$C_{-2}^{control}$	$2N^{control}$

Similarly, Table 4 illustrates the pairwise allele frequencies of two SNPs, SNP_i and SNP_j. C_{ij}^{pop} reports the number of occurrences of the nine possible combinations of alleles {0/0, 0/1, 0/2, 1/0, 1/1, 1/2, 2/0, 2/1, 2/2} in a population.

The χ^2 hypothesis test determines whether or not to reject the null hypothesis, which states that allele frequencies in the case and control populations follow a similar distribution. The χ^2 statistic of a single SNP is defined as $\chi^2 = \sum_{i \in \{0,1,2\}} \frac{(N_i^{case} - N_i^{control})^2}{N_i^{control}}$. From the χ^2 statistics, one can compute the *p-value* of each SNP. A *p-value* lower than a given threshold (e.g., 10^{-8}) indicates a strong correlation between the variant and the phenotype of interest [5].

4 TIDES OVERVIEW

Tides comprises four steps:
Step A: Each datacenter transfers its data to its local TEE, which computes intermediary statistics (e.g., the allele counts).
Step B: TEEs share their counts with others over the network.
Step C: TEEs run a consensus algorithm to agree on the counts and datacenters to use in the summary statistics computations.
Step D: f+1 predefined TEEs collects all allele counts to use, compute the summary statistics and release them.

Tides allows N datacenters that communicate over an asynchronous network to collaboratively compute summary statistics. Each of the datacenters hosts a TEE to manipulate encrypted data locally without getting access to them. Each TEE holds an asymmetric key pair, and we assume that all TEEs initially know the public key of all other TEEs. All communications between TEEs are sent encrypted over the network, and data are stored encrypted on the host end. A TEE can load encrypted data, decrypt them, and then run code on them. TEEs also dispose of a way to request platform and code attestation from other parties before starting the protocol and when exchanging intermediary data. This is to guarantee that enclaves run the same code and therefore cannot manipulate messages. This is a fundamental step as it enable parts of the protocol to only aim at crash-tolerance instead of byzantine fault tolerance.

Threat model. Tides considers an adversary that can control $f \leq \lfloor (N-1)/2 \rfloor$ datacenter. This adversary can access the datasets of these f datacenters, isolate their TEE from the network, and observe the final computed statistics. However, we assume that the adversary is unable to inject malicious data. This adversary can attempt to disrupt a protocol by arbitrarily deviating from it, and can attempt to launch privacy attacks to obtain information on the datasets of the correct datacenters. TEEs do not deviate from a given protocol but they might appear to be crashed (hence the $f \leq \lfloor (N-1)/2 \rfloor$ assumption). The adversary can attempt to gain additional information about data their TEEs access through side-channel attacks. We assume that malicious datacenters know their own data, they might still try to gain information during the later steps of the protocol, such as intermediate results from other datacenters or values before they are ready for release.

Implementation case study – GWAS statistics: We implemented Tides for GWAS statistics computation, on genomic datasets. Briefly, this consists in analysing genetic variations at a specific location in the genome, where each individual's genotype is encoded as 0, 1, or 2. This encoding reflects the presence or absence of a certain genetic variant at each location in the individual's genome, considering the double-strand nature of the human genome ($0 < number\_variants < 2$). Tides computes the top K key GWAS statistics, i.e., single or pairwise allele frequencies, χ^2, and Linkage Disequilibrium (LD) across datasets held by the datacenters [5, 60, 72]. For further description of genomic data and GWAS statistics we refer the reader to Sec. 3.

5 AVAILABILITY OF INTERMEDIARY COUNTS AND AGREEMENT

In Tides, the datacenters compute summary statistics in a distributed and intrusion-tolerant manner, by leveraging a crash-fault randomized consensus algorithm executed by TEEs. This allows Tides to progress even when up to f datacenters are malicious.

Reliable broadcast of intermediary counts. All TEEs initially compute intermediary counts, i.e., contingency tables (GWAS case study – cf. Sec. 3), on their local datasets, and broadcast them encrypted and signed to all the other TEEs using a reliable broadcast protocol [24, 83], When a datacenter receives the intermediary counts of a given remote dataset, it verifies their authenticity (i.e., that they have the datacenter's TEE signature) and, upon success, stores them encrypted and signed in its local filesystem [26, 73]. TEEs never equivocate, but the adversary might prevent a TEE from sending its local intermediary counts to all the other TEEs. In this case, the intermediary counts of a given datacenter might never be received by any correct datacenter or might only be delivered by a subset of the datacenters. After a configurable time interval, the TEEs agree on the datasets whose intermediary counts are reliably stored in at least f+1 datacenters [1]. Indeed, if a datacenter's intermediary counts are stored in f+1 datacenters then it is guaranteed that they will always be available: a correct datacenter in any set

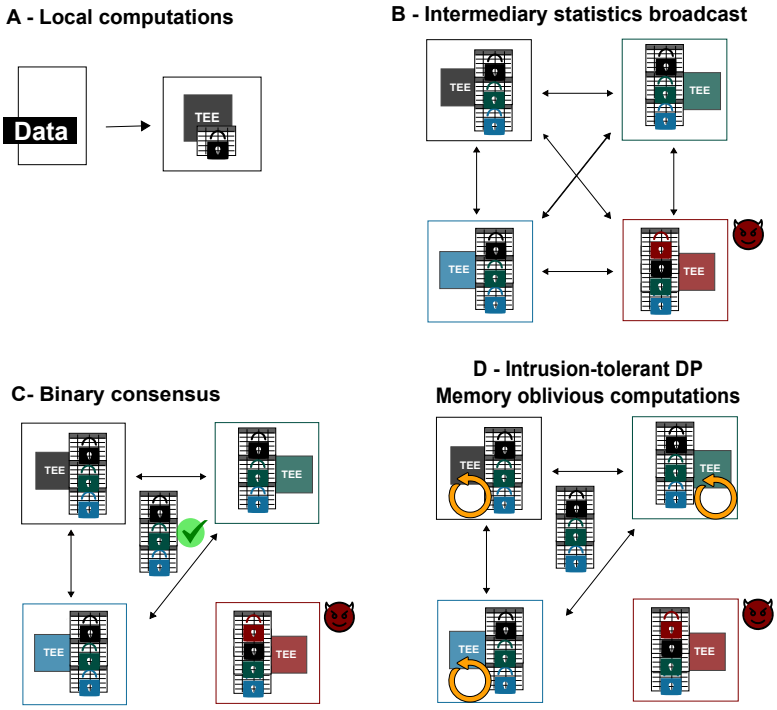


Figure 1: The main steps of Tides in a system of 4 datacenters that can tolerate $f = 1$ faulty datacenter. In this example, DC_4 (in red) is faulty and does not participate in all steps. Step A consists in having datacenters sending their encrypted data to their local TEE, which then computes intermediary statistics. Step B requires each TEE to reliably broadcast its intermediary statistics to other TEEs. In Step C, TEEs execute binary consensus instances to agree on a common subset of datasets to use to compute the final summary statistics. Finally, in step D, $f + 1$ TEEs compute the summary statistics with memory-oblivious algorithms and release them publicly.

of f+1 datacenters is assured to exist and would always send its intermediary counts to a requesting datacenter. The datasets that have been received by f+1 datacenters are therefore those over which the summary statistics can be computed and released.

Parallel crash-tolerant binary consensus. Using one binary consensus instance per datacenter D in the system Tides leads all datacenters to an agreement and decides whether D's dataset is usable in the summary statistics computations. Tides adopts the Ben-Or [7] leaderless randomized consensus algorithm. It has been shown that this algorithm tolerates up to f crashes in a system of $2f$+1 replicas [1], with minimal modifications from the original work, which assumed that $N > 5f$ [7]. While Tides currently executes Ben-Or's algorithm entirely inside TEEs, it is known that TEE-based consensus algorithms can use TEEs only to maintain trusted counters [26, 73]. We leave such optimization for future work. For the justification on the use of Ben-Or's algorithm we refer the reader to Sec. 2.

Replicating the summary statistics computation at f+1 datacenters. Eventually, each consensus instance leads all correct datacenters to decide the same binary value. If a datacenter has received a dataset, it proposes value 1 in the corresponding instance. A binary consensus that outputs 1 indicates that the datacenter's intermediary results are reliably stored among the datacenters. Once $N - f$ consensus instances decide 1, correct datacenters propose a 0 in the remaining consensus instances [51]. Once consensus is reached for the N datasets, f+1 predetermined datacenters request the intermediary results of the retained datasets statistics from other datacenters if they do not have them. After receiving all required datasets, the f+1 datacenters compute the final summary statistics. By replicating the computation in f+1 different locations, we are guaranteed that at least one of them will release the correctly computed value. As replicas only exchange intermediary results, this does not result in malicious datacenters obtaining datasets protected by regional regulations.

6 INTRUSION-TOLERANT DIFFERENTIAL PRIVACY

We now recall the notions of differential privacy (DP) and sensitivity. We then establish the sensitivity of the GWAS statistics when SNPs are encoded in three possible values. We finally determine the amount of noise to add to the final GWAS results to enforce DP under our threat model.

6.1 Classical definitions

Consider a randomized algorithm κ. In the following, $Range(\kappa)$ denotes the set of every possible output of κ and ϵ is a positive real number.

Definition 6.1 (Differential privacy [31]). κ satisfies ϵ-differential privacy if, for all datasets D and D' that differ on a single element, and all $S \subseteq Range(\kappa)$, $\mathrm{P}(\kappa(D) = S) \leq e^{\epsilon} \cdot \mathrm{P}(\kappa(D') = S)$.

DP ensures that an adversary that observes the output of Alg. κ and knows a potential dataset entry cannot determine whether that entry was part of the data. DP does not maintain its guarantees against an adversary that knows a subset of an input dataset, contrary to local DP.

Definition 6.2 (Local DP [30, 42]). κ satisfies ϵ-local differential privacy if, for all inputs x and x', and all $y \in Range(\kappa)$, $P(\kappa(x) = y) \leq e^{\epsilon} \cdot P(\kappa(x') = y)$.

Definition 6.3 (Sensitivity). The sensitivity of a function $f : D^N \rightarrow \mathbb{R}^d$, where D^N denotes the set of all datasets with N individuals, is the smallest number $S(f)$ such that $\|f(D) - f(D')\|_1 \leq S(f)$ for all datasets $D, D' \in D^N$ differing in a single individual.

Once the sensitivity of a function f, for example, a GWAS statistics, is known, releasing $f(D) + b$, where b is a random noise drawn from a Laplace distribution with mean 0 and scale $S(f)/\epsilon$ enforces ϵ-differential privacy.

6.2 Sensitivity of GWAS statistics

Tides relies on the sensitivity of GWAS statistics to adapt the DP noise to the adversarial model considered. We again refer the reader to Sec. 3 for the background on genomic data encoding and GWAS statistics. The sensitivity of the single allele frequency and the χ^2 metrics are directly reproduced from [33, 72], while the sensitivity of the pairwise allele frequency and the Linkage Disequilibrium (LD) metrics are evaluated here for the first time.

From this point, we refer to the dataset size as G and the number of SNPs per genotype as M. In the following, C_i and C'_i are the respective numbers of minor alleles in two datasets D and D' for SNP i. Similarly, $C_{00}^{i,j}$, $C_{01}^{i,j}$ and $C_{11}^{i,j}$ represent the pairwise allele frequencies for values (0,0), (0,1).

• **Single allele frequencies [72].** The maximal difference between two datasets at a given SNP is equal to 2, since the minor allele counts per genome can take values 0, 1, or 2. The corresponding sensitivity for a given SNP over two populations of N genomes is $\frac{1}{G}\left|\sum_{i=1}^{G} C_i - \sum_{i=1}^{G} C'_i\right| \leq \frac{2}{G}$. The sensitivity over G genomes is therefore $\frac{2}{G}$.

• **Pairwise allele frequency.** For this statistic the method is similar to the one above, but we release the top K frequencies. The possible values for pairs of SNPS are 0/0, 0/1, 0/2, 1/0, 1/1, 1/2, 2/0, 2/1, 2/2, where, for example, 2/1 means that the individual has two alternative alleles in one SNP position and a single alternative allele in the other SNP position of the pair (see Sec. 3 for genotype encoding per SNP position). Changing one allele in a genome results in a difference equal to 2, therefore $\frac{1}{G}\left|\sum_{i,j} C_{i,j} - \sum_{i,j} C'_{i,j}\right| = \frac{2}{G}$.

• χ^2 **statistic.** The sensitivity of the χ^2 statistic is $\frac{4G}{G+2}$ [72].

• **Linkage Disequilibrium.** As linkage disequilibrium is easier to interpret when genomes are encoded over 2 possible values than with 3, Tides computes contingency tables over 2 values. To do so, Tides therefore computes the frequency table for any two chosen SNPs, that from now on will be called l_1 and l_2. With $C_{-0} + C_{-1} = C_{0-} + C_{1-} = G$, the table looks as follows:

	0	1	Total
0	C_{00}	C_{01}	C_{0-}
1	C_{10}	C_{11}	C_{1-}
Total	C_{-0}	C_{-1}	$2G$

The LD statistic is then $LD = \frac{C_{00}}{2G} - \left(\frac{C_{0-}}{C_{0-}+C_{1-}} \cdot \frac{C_{-0}}{C_{-0}+C_{-1}}\right) = \frac{C_{00}}{2G} - \left(\frac{C_{0-}}{G} \cdot \frac{C_{-0}}{G}\right) = \frac{C_{00}}{2G} - \left(\frac{C_{00}+C_{01}}{G} \cdot \frac{C_{00}+C_{10}}{G}\right)$. We now need to maximize $|LD - LD'|$, with the two values being, respectively, the statistics on two datasets D and D' that differ in exactly one element. To do this, we can maximize the directional derivative of LD in all possible directions of change. First, we need the partial derivative over C_{00}, C_{01} and C_{10}, which we note x, y, z and compute as follows: $\frac{dLD}{dx} = \frac{1}{2G} - \frac{2x}{G^2} - \frac{y}{G^2} - \frac{z}{G^2}$, $\frac{dLD}{dy} = -\frac{x}{G^2} - \frac{z}{G^2}$, and $\frac{dLD}{dz} = -\frac{x}{G^2} - \frac{y}{G^2}$. Given that the original formula for the statistic is symmetric in y and z we can restrict our analysis, without loss of generality, to the case that concerns only one of those 2 variables. The directions to consider over $(x, y, z, G-x-y-z)$ are then: $(1, -1, 0, 0)$, $(0, 1, -1, 0)$, $(0, 1, 0, -1)$, $(1, 0, 0, -1)$ and their inverse. In all these cases, the maximum value is obtained when $x = y = z = 0$ and so the sensitivity is $|LD(0,0,0) - LD(1,0,0)| = \left|0 - \left(\frac{1}{2G} - \frac{1}{G^2}\right)\right| \leq \frac{G-2}{2G^2}$.

6.3 Intrusion-Tolerant Sensitivity

The noise to be applied by a datacenter to the statistics depends on the number of samples that each datacenter holds, and more precisely on the largest or smallest combination of $N-f$ datasets. In Tides, the datacenters are aware of the sizes of the datasets held by the others, as explained in Sec. 4. The datacenters determine the noise to apply using the sensitivity of statistics S and the privacy parameter ϵ. Larger ϵ values result in less added noise and weaker privacy guarantees, while smaller values provide stronger privacy guarantees at the cost of reduced data utility.

Consider a statistic S that Tides aims to compute and securely release. Let $D_S(K, G)$ denote the amount of noise required for K values of statistic S to be securely released over a dataset of size G. Define B as the vector that contains the size of each dataset, and let G_{N-f} be the smallest (or largest) sum of $N-f$ elements from B. The smallest sum will be used in the case where the sensitivity decreases with larger datasets, while in the case where the value increases, the largest sum is considered. The minimum noise required to release K statistics of type S with differential privacy is $D_S(K, G_{N-f})$. In practice, depending on the value of f, $D_S(K, G_{N-f})$ can continuously vary between the noise level produced by local DP (when $f = N-1$) and the one produced by global DP (when $f = 0$).

With DP, the noisy value S' of statistics S is obtained by adding a random Laplace noise with scale parameter b, i.e., $S' = S + Lap(b)$. The value of parameter b is proportional to the sensitivity of the statistic and inversely proportional to the privacy parameter (ϵ), and is computed as $b = k \cdot \frac{\triangle_{B,N-f}}{\epsilon}$, where $\triangle_{B,N-f}$ refers to the sensitivity of the statistic computed by the datacenters, which also depends on the dataset sizes B and $N-f$.

To achieve differential privacy the parameter b needs to be higher than a specific threshold. This threshold depends on the sensitivity of the statistic computed over the dataset size. In the ideal scenario of local differential privacy, we can reduce b to be exactly the value of this threshold. In the distributed scenario, we need to calculate a

Algorithm 1 Computation of noise to be applied on a datacenter's released statistic

Require: $D; \epsilon; \triangle; f; B$
Ensure: noise = the amount of noise to add to the statistic to be released for datacenter i
size $\leftarrow \min(\text{sbs}_{N-f}(B)) + B[i]$ (or max)
scale $\leftarrow \frac{D_S(k,\text{size})}{\epsilon}$
noise $\leftarrow Lap(\text{scale})$

Algorithm 2 Oblivious count (pair of bits)

Ensure: c_{ab} contains the count of ab
$c_{00} \leftarrow 0; c_{01} \leftarrow 0; c_{10} \leftarrow 0; c_{11} \leftarrow 0$
for $(p_1, p_2) \in D$ **do**
$c_{00} \leftarrow c_{00} + (\neg(p_1 \vee p_2)\ \&\ 1)$
$c_{01} \leftarrow c_{01} + ((\neg p_1 \wedge p_2)\ \&\ 1)$
$c_{10} \leftarrow c_{10} + ((p_1 \wedge \neg p_2)\ \&\ 1)$
$c_{11} \leftarrow c_{11} + (p_1 \wedge p_2\ \&\ 1)$

value that would be higher than the threshold computed over the worst case. This is showcased in Alg. 1 and a proof that the noise scale is always sufficiently high is shown in Theorem 6.3.

Alg. 1 details how each datacenter computes the noise it adds to the statistics. In this algorithm, ϵ is the privacy parameter, $D_s(K, G)$ is the noise scale for statistic S, K denotes the number of statistics to be released, and G is the global dataset size. Additionally, B is a vector containing the sizes of the datasets with B_i denoting the size of the i-th dataset. We also define $\text{sbs}_a(B)$ as a function that returns a vector of all sums of combinations of a elements in B. The noise addition procedure needs to be executed twice per execution of the algorithm: a first time on the clean statistic computed over alleles, and a second time to perturbate the top-K statistics obtained after the desired method is applied. This is also showcased in Algs. 5 and 6. Moreover, the first time noise is added the scale should be multiplied by a factor of $4k$, with k being the wanted amount of values to release, while the second time a factor of $2k$ suffices [9, 33, 72]. We now state and prove that applying the noise of Alg. 1 ensures DP.

Theorem 6.4 (Intrusion-Tolerant Differential Privacy). *Consider a statistic S with sensitivity $\triangle_G$, G as the dataset size for the statistic, a system of N parties, of which up to f may be malicious and colluding, and a vector B containing each party's dataset size for all N parties. If the sensitivity $\triangle_G$ is inversely or directly proportional to G, then adding noise to the data using Alg. 1 ensures DP.*

Proof. Up to f parties can collude to infer information over the remaining $N-f$ datasets. Thus, to successfully protect those datasets the sensitivity of the metrics can be computed based on the sum of their sizes, referred as S_{true}. In Alg. 1 the sensitivity is computed over the smallest $N-f$ datasets in B when it is inversely proportional to a dataset size, or over the largest ones otherwise. This implies that the computed sensitivity value S_c used for the noise computation will always be larger than S_{true} and therefore enforce DP. □

6.4 Intrusion-Tolerant Sensitivity for GWAS Statistics

We discuss in this section the methods we use to compute the sensitivity of GWAS statistics.

Singlewise Allele Frequency (single-AF). The worst case for this statistic happens when an adversary controls the f largest datasets, with total size G_f, while the aggregate size of the other ones is G_{N-f}, such that $G = G_f + G_{N-f}$. The adversary can remove the contribution of the datasets it controls from the released statistic S. If we denote C_{N-f} as the counts for a specific SNP in the benign datasets, C_f as the count for the malicious ones, and S_{N-f} as the statistic computed over the non-colluding subsets, the adversary can compute $\left(\frac{C_{N-f}+C_f}{G} \cdot G - C_f\right) \cdot \frac{1}{C_{N-f}} = \frac{C_{N-f}}{G_{N-f}} = S_{N-f}$. DP is ensured by introducing noise using a sensitivity computed over the smallest $N-f$ datasets.

Pairwise Allele Frequency (pairwise-AF). The sensitivity computation for the release of pairwise-AF is analogous to the one used for single-AF. Here again, an adversary can fully extract the frequency of a specific pair of SNPs from the datasets of the benign parties, which implies that one must compute the sensitivity over the smallest $N-f$ datasets.

χ^2 statistic. The sensitivity of the χ^2 statistic is asymptotically constant, increases slowly with the dataset size and converges to 4 [33, 72].

Linkage Disequilibrium (LD). We established in Sec. 6.2 the sensitivity formula of the LD statistic over a given dataset. In our terminology, where we define $2G_{N-f}$ as the aggregate size of the honest datasets, the remaining dataset that needs to be protected is $\frac{G_{N-f}-2}{2G_{N-f}^2}$, which results in the used sensitivity being computed over the subset of datasets not controlled by the adversary.

7 MEMORY-OBLIVIOUS COMPUTATIONS

Tides uses primitive oblivious operations to compute summary statistics within TEEs. In particular, counting frequencies needs to be done without branching and by accessing every intermediate count. Alg. 2 is an example that maintains 4 counters, one for every possible combination of a pair of binary variables.

The other main primitive used is an algorithm that given a list of computed statistics returns its top-K values. In case K is a small value, indicatively smaller than $\log N$, we use a bubbling algorithm (Alg. 3 in [3]). Another possibility is to save all possible computed values and then sort them using an oblivious sorting algorithm [48], which we present this algorithm in Alg. 3. This method requires an oblivious algorithm that shuffles an array, presented in the second half of Alg. 3. By first shuffling in an oblivious way and then sorting the now shuffled array memory accesses are randomized and do not leak information.

Given these primitive operations, Tides first counts the frequencies of necessary SNPs and then computes a desired GWAS statistic. This is showcased in Alg. 4 Obtaining the memory oblivious and differentially private version of this algorithm for releasing the top-K GWAS statistics is then straightforward. We detail the bubbling-based algorithm from [3] in Alg. 5, and its oblivious sorting alternative in Alg. 6.

Algorithm 3 Oblivious sort and shuffle

Ensure: Sort input array A
  Oblivious shuffle A
  Sort A with any sorting algorithm
Ensure: Shuffle input array A
  $A' \leftarrow$ empty array
  for $v \in A$ **do**
    append a secure random value into A'
  Sort A based on the values of A' with any sorting algorithm

Algorithm 4 Oblivious generic statistic

Require: database D and needed SNP s. $f(c)$ calculate statistic f over counted frequencies c
Ensure: R has value equal to $f(s)$, computed in a memory oblivious way.
  $R \leftarrow$ array of size K
  for genome $G \in D$ **do**
    update frequency counts c for SNP s using Algorithm 2
  $R \leftarrow f(c)$

Algorithm 5 Oblivious top-K statistic (bubbling method)

Require: database D and needed top-K.
Require: $f(D, s)$ calculates statistic f over SNP s and database D using Alg. 4
Ensure: R contains top K statistics and relative SNPs
  $R \leftarrow$ array of size K
  for SNP $s \in D$ **do**
    $n_s \leftarrow$ random Laplace noise `// Apply noise to statistics`
    append $(f(D,s)+n_s, f(D,s))$ to R with oblivious bubble `// Append tuple including original value as well`
  for $(\_, s) \in R$ **do** `// Extract clean value`
    $R_i \leftarrow s +$ random Laplace noise `// Apply noise to top-K`

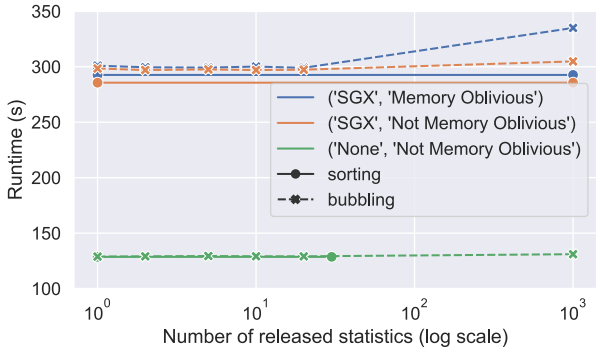
Algorithm 6 Oblivious top-K statistic (sorting method)

Require: database D and needed top-K.
Require: $f(D, s)$ calculates statistic f over SNP s and database D using Alg. 4
Ensure: R contains top K statistics and relative SNPs
  $S \leftarrow$ empty list
  $R \leftarrow$ empty array of size K
  for SNP $s \in D$ **do**
    $n_s \leftarrow$random Laplace noise `// Apply noise`
    $S \leftarrow S \cup (f(D,s) + n_s, f(D,s))$ `// Append tuple including original value as well`
  Obliviously sort S with Alg. 3
  for $i \in (0, 1, \cdots, k)$ **do**
    $(\_, s) \leftarrow S_i$ `// Extract clean value`
    $R_i \leftarrow s+$ random Laplace noise `// Apply noise to top-K`

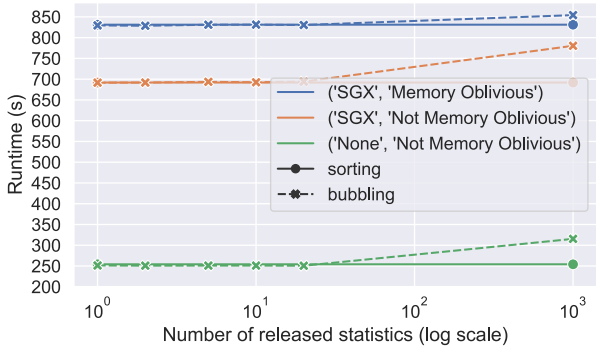
8 PERFORMANCE EVALUATION

We evaluated Tides on a laptop equipped with an Intel(R) Core(TM) i7-10510U CPU using Gramine[1] on GWAS statistics (cf. Sec 3). We used three datasets: (i) the **iDASH dataset**[2] that contains 2000 genomes over 1.04M SNPs; (ii) a **dbGaP dataset**, namely dbGaP's PAGE: Multiethnic Cohort study (dbGaP Study Accession: phs000220.v2.p2) where we used 548 genomes and 1.7M SNPs; and (iii) the **1000GP dataset** that contains 2,548 genomes and 2.5M SNPs. We evaluate runtime, utility, network consumption, and false positives metrics, and exclude Tides's consensus latency as it is inherited from its binary consensus algorithm and because Tides's latency is largely dominated by the memory oblivious computations within the TEEs. For single-AF and χ^2, we compared the bubbling and sorting-based oblivious methods to compute the top-K values.

[1]https://gramineproject.io/
[2]This dataset was generated for the iDASH challenge Track 2 in 2017.



(a) Single allele frequency

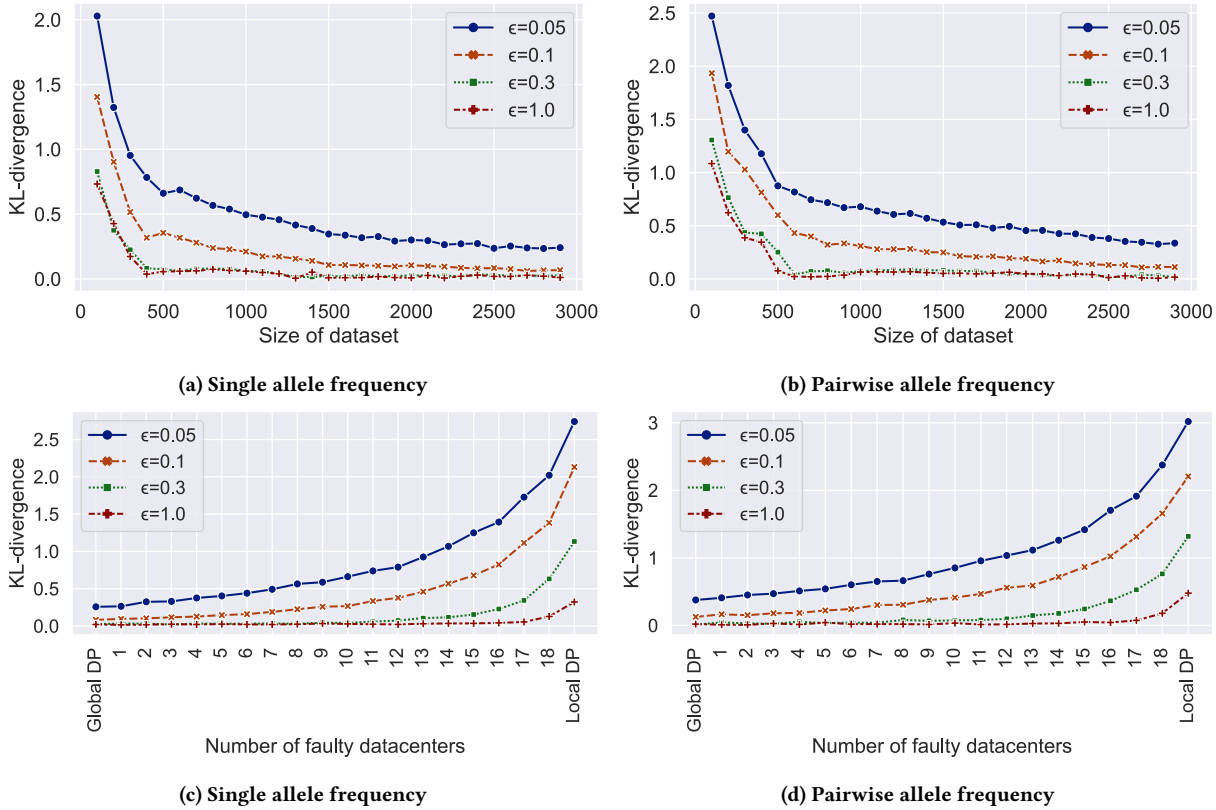


(b) χ^2

Figure 2: Runtime for single allele frequency and χ^2 with the 1000GP dataset.

8.1 Runtime

We evaluate the runtime for the single-AF (Fig. 2a) and the χ^2 statistics (Fig. 2b) on all datasets. Given the space constraints, only the results for the 1000 GP dataset are shown, as the others show the same trend. The baseline (green line) shows the runtime without TEEs and memory oblivious algorithms. We evaluated the runtime for different number of released values (1 to 1,000), and for three different security settings: no security measures (green line), SGX execution (secure hardware only, orange line), and SGX with memory oblivious computations (full security, blue line) For this experiment, Tides's code is executed on a single machine. The runtime is the highest when applying the most secure implementation (TEE and memory oblivious algorithm, blue line) for all datasets, but it remains practical (e.g., less than 30 s for single-AF). With



(a) Single allele frequency

(b) Pairwise allele frequency

(c) Single allele frequency

(d) Pairwise allele frequency

Figure 3: KL-divergence of single and pairwise allele frequencies (AF) depending on the dataset size ($f = 0$) or on the number of faulty datacenters ($0<f<20$).

a low number of SNPs the bubbling method is the fastest, but it becomes slower than the sorting method for larger SNP numbers (10^3). Figs. 2a and 2b show the runtime for the single-AF and the χ^2 computations on the 1000GP dataset. The use of Intel SGX with single-AF has a big performance impact, while adding memory obliviousness does not add any additional complexity, except when releasing a high number of values when using the oblivious bubbling method. The χ^2 statistic requires more memory accesses and the memory-oblivious implementation requires significantly more time than the non-oblivious version. Performance can be slightly improved by employing the bubbling technique when releasing a small number of statistics.

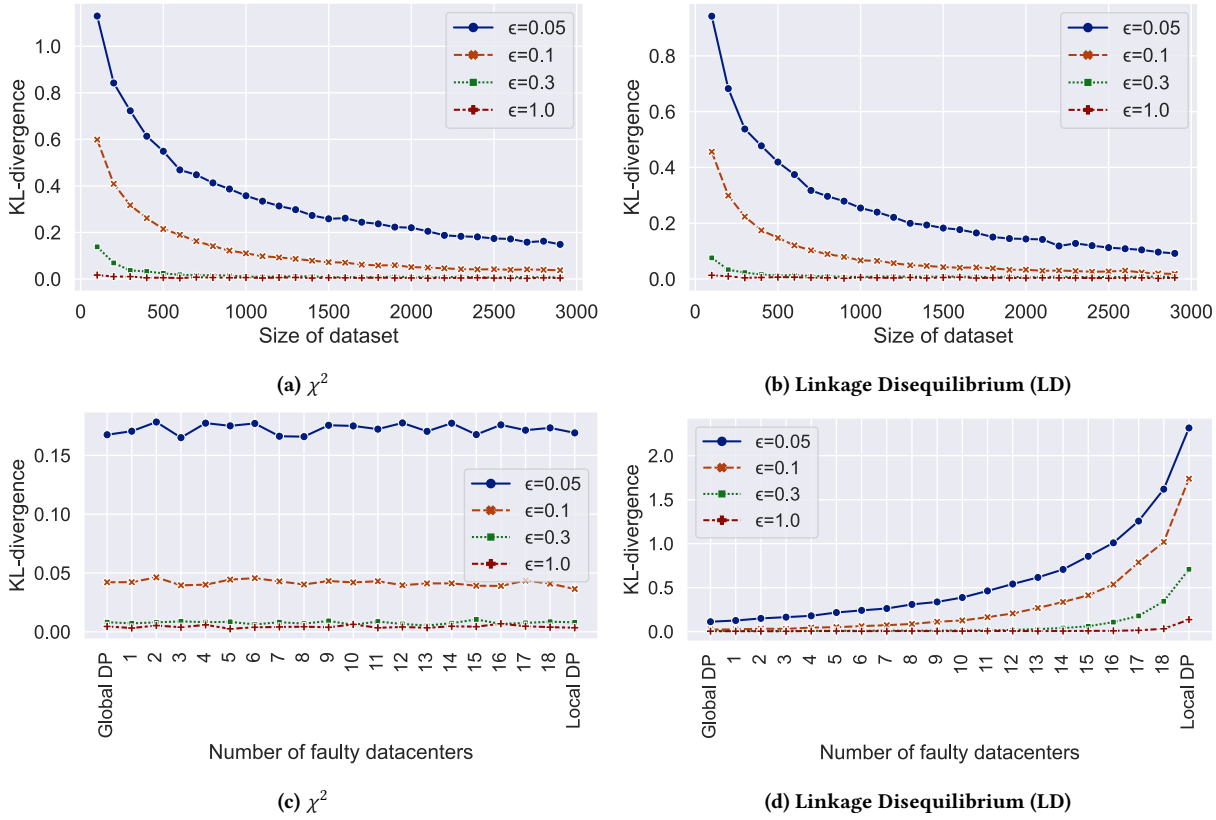
8.2 Statistical utility

We assessed the statistics utility using the setup of Fienberg et al. [33] and f=0. We measure the Kullback-Leibler divergence (KL-divergence) [59], a metric for statistical distance where a small value $D_{KL}(P||Q)$ indicates a small difference between distributions P and Q. We chose two of the four frequency tables from [33] to simulate realistic SNPs. Using these tables, for each statistic and for increasing dataset size N, we randomly generated 10K tables. For single-AF and χ^2 we used the first table, and since they do not use case/control stratified data only the first column was used. For pairwise-AF and LD, we used the outer product of the first column of both tables to obtain realistic frequencies. Using the 10K generated tables, both the original and perturbed statistics were computes, and we compared their similarity using KL-divergence. The results, shown in Figs. 3a, 3b, 4a and 4b, indicate that KL-divergence decreases as dataset size, N, or ϵ increase. For all statistics except χ^2, the noise scale tends to 0 as N grows, while for χ^2, it asymptotically reaches 4. This suggests that as N increases, the KL-divergence should not decrease, but as the value of the χ^2 statistic increases, the noise ratio of the original statistic decreases, reducing divergence.

8.3 Statistical utility with faulty datacenters

We now study the impact of the number f of faulty datacenters on results utility. We fix the number of individuals G to 2500, and the number of datacenters N to 20. We also compare Tides to global and local DP, where each datacenter respectively adds noise assuming that all other datacenters are correct (i.e., $f = 0$) or possibly faulty (i.e., $f = 19$). The individuals were divided among datacenters using a multinomial distribution where the probability of an individual to belong in a dataset is $\frac{1}{20}$. Figs. 3c, 3d, 4c and 4d report the KL-divergence of all statistics we consider when the number of faulty datacenters vary and for different ϵ values.

Increasing the number of faulty datacenters or reducing ϵ increases KL-divergence for single and pair-wise allele frequency,



(a) χ^2 **(b) Linkage Disequilibrium (LD)**

(c) χ^2 **(d) Linkage Disequilibrium (LD)**

Figure 4: KL-divergence of χ^2 and Linkage Disequilibrium (LD) depending on the dataset size ($f = 0$) or on maximum number of faulty datacenters ($0 < f < 19$).

and for linkage disequilibrium (Figs. 3c, 3d, and 4d). In addition, the noise generated by Tides to enforce DP is always lower than the one that local DP would produce. However, one can see that the KL-divergence of the χ^2 statistic does not change when the number of faulty datacenters increases (Fig. 4c). As explained before, this occurs because the scale of the noise added to the perturbed χ^2 statistic remains asymptotically constant. Even as the number of faulty nodes increases, the amount of noise added to the χ^2 statistic remains constant. In this experiment, since the dataset's size and values remain fixed, the difference between the perturbed and the original statistic also stays constant, unlike what is illustrated in Fig 4a where the dataset size varies.

8.4 Network consumption

Tab. 5 and Tab. 6 respectively provide the expected and worst total number of bytes transferred over the network for different number of datacenters and SNPs. We assumed that the number of faulty datacenters is $f = \lceil (N-1)/2 \rceil$. Each cell reports the amount of bytes that a single replica will have to send.

At the beginning of Tides, datacenters send each other the intermediary counts of the computation they performed on their local dataset. In the case of GWAS, each of these messages takes the form of a contingency table that represents the counts of which values the genotypes take in the case and control datasets.

A contingency table contains six values that can each be represented by a 2-byte integer, which is sufficient to encode the counts obtained over a bit more than 65K genomes, for a total of 12 bytes per contingency table. For each SNP (or pair of SNPs for pairwise-allele frequency and Linkage Disequilibrium), each datacenter communicates one table to all other parties. This means that assuming N datacenters, every participating datacenter sends all other participants their table, which therefore involves in the worst case $N \cdot (N-1)$ messages. The total network consumption of the broadcast protocol in bytes finally depends on the number of datacenters N and the number of SNPs I and is equal to $12I \cdot N(N-1)$. The average network consumption per datacenter that participates in this phase is then $12I \cdot (N-1)$.

All datacenters then echo every intermediary dataset they have received in the reliable broadcast phase [83]. This adds an extra $(N-1)^2$ messages for each datacenter, giving us $12I \cdot (N-1)^2$ more data exchanged. Following reliable broadcast, each node participates in binary consensus, which we implement using Ben-Or's algorithm [8]. We do not evaluate the performance of this algorithm as it is well understood and because it is independent from the other steps of Tides. The expected latency of the algorithm is constant in stable networks when a correct sender.

Note that the expected latency of Ben-Or's crash consensus algorithm is high, namely 2^N. Techniques borrowed from Byzantine

Table 5: Expected-case network consumption (MiB) for exchanging intermediary results.

#DCs \ #SNPs	10^3	10^4	10^5	10^6
5	0.06	0.57	5.72	57.22
10	0.26	2.57	25.75	257.49
15	0.60	6.01	60.08	600.81
20	1.09	10.87	108.72	1087.19

Table 6: Worst-case network consumption (MiB) for exchanging and requesting intermediary results.

#DCs \ #SNPs	1e3	1e4	1e5	1e6
5	0.07	0.66	6.58	65.80
10	0.27	2.72	27.18	271.80
15	0.62	6.24	62.37	623.70
20	1.12	11.16	111.58	1115.80

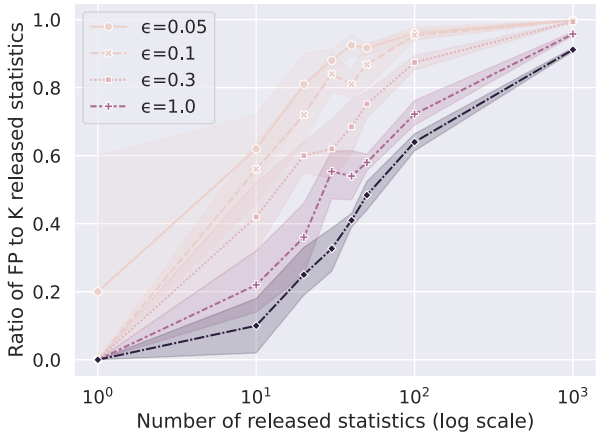
consensus works [25] could be used to reduce it, which we leave for future work.

After the consensus phase, the f+1 TEEs that have been selected to compute the GWAS statistics using memory-oblivious algorithms might have to request the intermediary counts they have not received from other TEEs. In the worst case, each TEE will have to download f datasets, which means $(f+1) \cdot f$ additional transfers of intermediary counts, which account for $12I \cdot (f+1)f$ additional bytes. The average network consumption per datacenter that participates in this phase is then $12I \cdot f$. In the expected case, this pull mechanism is not triggered as every correct replica will have already delivered the reliably broadcast messages. We assume that datacenters have agreed on a common compression algorithm in advance, which reduces the amount of exchanged data. We consider a compression ratio of 25%, which known compression methods achieve on random data [52].

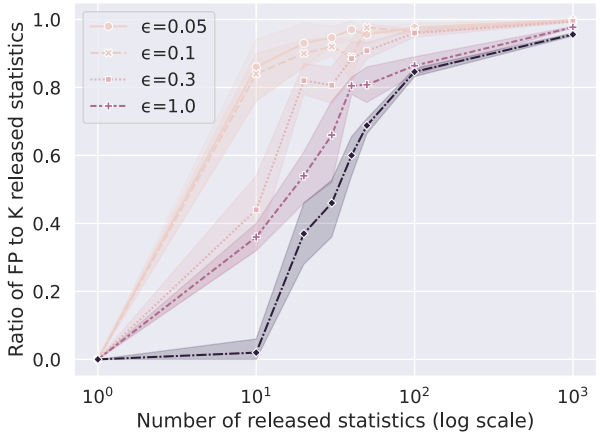
These results show that Tides is practical. For example, each datacenter would send around 1.1 GiB over the network for a GWAS involving 20 datacenters and 10^6 SNPs. Note that this number does not depend on the number of individuals that contribute their data.

8.5 Top-K False Positives

A top-K false positive (FP) is a value reported as part of the top-K statistics that does not belong in the non-perturbed top-K set, replacing a value that should have been released. In this context, the number of false positives and false negatives are equal. We ran the experiments on the same datasets used in the runtime experiments, excluding the χ^2 statistic for the 1000Genomes and dbGaP datasets due to the absence of case/control stratified data. Figs. 5a and 5b show the proportion of FPs in the top-K released statistics depending on ϵ. The FP ratio increases with ϵ and with the number of released statistics but remains reasonable (under 50%) when $\epsilon \leq 1$ and $k \leq 10$.



(a) Single allele frequency



(b) χ^2

Figure 5: False Positives for different numbers of released statistics (K) and privacy parameter ϵ on the iDash dataset.

9 CONCLUSION

Summary statistics are essential computation for assessing statistical significance. The datasets size, sensitivity and distributed nature pose new challenges calling for the design of federated summary statistic computation solutions. In this paper, we present Tides, the first privacy-preserving summary statistics framework designed to tolerate a minority of malicious datacenters. Tides combines fault-tolerant protocols, including a crash-tolerant randomized consensus algorithm that Trusted Execution Environments run, and memory oblivious algorithms to protect the data during the computation of summary statistics against a powerful adversary that can use side-channel attacks, and applies a personalized DP noise to release the statistics. Our case study experiments were run over three representative genomic datasets showing that Tides remains practical in GWAS computations despite the computational overhead of its memory oblivious algorithms and the use of TEEs, while preserving data utility better than local differential privacy.

REFERENCES

[1] Marcos K Aguilera and Sam Toueg. 2012. The correctness proof of Ben-Or's randomized consensus algorithm. *Distributed Computing* 25, 5 (2012), 371–381.

[2] AKM Mubashwir Alam and Keke Chen. 2023. Making your program oblivious: a comparative study for side-channel-safe confidential computing. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. IEEE, 282–289.

[3] Aref Asvadishirehjini, Murat Kantarcioglu, and Bradley Malin. 2020. A Framework for Privacy-Preserving Genomic Data Analysis Using Trusted Execution Environments. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 138–147.

[4] Md Momin Al Aziz, Shahin Kamali, Noman Mohammed, and Xiaoqian Jiang. 2021. Online Algorithm for Differentially Private Genome-wide Association Studies. *ACM Transactions on Computing for Healthcare* 2, 2 (2021), 1–27.

[5] Gregory S Barsh, Gregory P Copenhaver, Greg Gibson, and Scott M Williams. 2012. Guidelines for genome-wide association studies. *PLoS Genet* 8, 7 (2012).

[6] Johannes Behl, Tobias Distler, and Rüdiger Kapitza. 2017. Hybrids on Steroids: SGX-Based High Performance BFT. 222–237.

[7] Michael Ben-Or. 1983. Another advantage of free choice (extended abstract) completely asynchronous agreement protocols. In *Proceedings of the second annual ACM symposium on Principles of distributed computing*. 27–30.

[8] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. 1994. Asynchronous secure computations with optimal resilience. In *Proceedings of the 13th annual symposium on Principles of distributed computing*. ACM, 183–192.

[9] Raghav Bhaskar, Srivatsan Laxman, Adam Smith, and Abhradeep Thakurta. 2010. Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 503–512.

[10] Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser. 2020. Secure large-scale genome-wide association studies using homomorphic encryption. *National Academy of Sciences* 117, 21 (2020), 11608–11613.

[11] Dan Bogdanov, Liina Kamm, Sven Laur, Pille Pruulmann-Vengerfeldt, Riivo Talviste, and Jan Willemson. 2014. Privacy-preserving statistical data analysis on federated databases. In *Privacy Technologies and Policy: Second Annual Privacy Forum, APF 2014, Athens, Greece, May 20-21, 2014. Proceedings 2*. Springer, 30–55.

[12] Silvia Bonomi, Jérémie Decouchant, Giovanni Farina, Vincent Rahli, and Sébastien Tixeuil. 2021. Practical Byzantine reliable broadcast on partially connected networks. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 506–516.

[13] Gabriel Bracha. 1987. Asynchronous Byzantine agreement protocols. *Information and Computation* 75, 2 (1987), 130–143.

[14] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. 2017. Software Grand Exposure:{SGX} Cache Attacks Are Practical. In *WOOT*.

[15] Christian Cachin and Stefano Tessaro. 2005. Asynchronous verifiable information dispersal. In *24th IEEE Symposium on Reliable Distributed Systems (SRDS'05)*. IEEE, 191–201.

[16] Sergiu Carpov and Thibaud Tortech. 2018. Secure top most significant genome variants search: iDASH 2017. *BMC medical genomics* 11, 4 (2018), 82.

[17] David Cerdeira, Nuno Santos, Pedro Fonseca, and Sandro Pinto. 2020. Sok: Understanding the prevailing security vulnerabilities in trustzone-assisted tee systems. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1416–1432.

[18] Sylvain Chatel, Christian Mouchet, Ali Utkan Sahin, Apostolos Pyrgelis, Carmela Troncoso, and Jean-Pierre Hubaux. 2023. PELTA – Shielding Multiparty-FHE against Malicious Adversaries. Cryptology ePrint Archive, Paper 2023/642.

[19] Feng Chen, Chenghong Wang, Wenrui Dai, Xiaoqian Jiang, Noman Mohammed, Md Momin Al Aziz, Md Nazmus Sadat, Cenk Sahinalp, Kristin Lauter, and Shuang Wang. 2017. PRESAGE: Privacy-preserving genetic testing via software guard extension. *BMC medical genomics* 10, 2 (2017), 48.

[20] Feng Chen, Shuang Wang, Xiaoqian Jiang, Sijie Ding, Yao Lu, Jihoon Kim, S Cenk Sahinalp, Chisato Shimizu, Jane C Burns, Victoria J Wright, et al. 2016. Princess: Privacy-protecting rare disease international network collaboration via encryption through software guard extensions. *Bioinformatics* 33, 6 (2016), 871–878.

[21] Wang Chenghong, Yichen Jiang, Noman Mohammed, Feng Chen, Xiaoqian Jiang, Md Momin Al Aziz, Md Nazmus Sadat, and Shuang Wang. 2017. SCOTCH: Secure Counting Of encrypTed genomiC data using a Hybrid approach. In *AMIA Annual Symposium Proceedings*, Vol. 2017. 1744.

[22] Hyunghoon Cho, David J Wu, and Bonnie Berger. 2018. Secure genome-wide association analysis using multiparty computation. *Nature biotechnology* 36, 6 (2018), 547.

[23] Scott D Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. 2015. Privacy-preserving GWAS analysis on federated genomic datasets. *BMC medical informatics and decision making* 15, 5 (2015), S2.

[24] Miguel Correia, Giuliana S Veronese, and Lau Cheuk Lung. 2010. Asynchronous Byzantine consensus with 2f+ 1 processes. In *Proceedings of the 2010 ACM symposium on applied computing*. 475–480.

[25] Tyler Crain, Vincent Gramoli, Mikel Larrea, and Michel Raynal. 2018. Dbft: Efficient leaderless byzantine consensus and its application to blockchains. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. IEEE, 1–8.

[26] Jérémie Decouchant, David Kozhaya, Vincent Rahli, and Jiangshan Yu. 2022. DAMYSUS: streamlined BFT consensus leveraging trusted components. In *Proceedings of the Seventeenth European Conference on Computer Systems*. 1–16.

[27] Wenliang Du and Mikhail J Atallah. 2001. Privacy-preserving cooperative statistical analysis. In *Seventeenth Annual Computer Security Applications Conference*. IEEE, 102–110.

[28] Sisi Duan, Michael K Reiter, and Haibin Zhang. 2018. BEAT: Asynchronous BFT made practical. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2028–2041.

[29] Yitao Duan, John Canny, and Justin Zhan. 2010. {P4P}: Practical {Large-Scale}{Privacy-Preserving} distributed computation robust against malicious users. In *USENIX Security*.

[30] John C Duchi, Michael I Jordan, and Martin J Wainwright. 2013. Local privacy and statistical minimax rates. In *2013 IEEE 54th annual symposium on foundations of computer science*. IEEE, 429–438.

[31] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*. Springer, 1–19.

[32] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. 2010. Differential privacy under continual observation. In *STOC*. 715–724.

[33] Stephen E Fienberg, Aleksandra Slavkovic, and Caroline Uhler. 2011. Privacy preserving GWAS data sharing. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 628–635.

[34] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel, and Tilo Müller. 2017. Cache attacks on Intel SGX. In *Proceedings of the 10th European Workshop on Systems Security*. 1–6.

[35] Tianyao Gu, Yilei Wang, Bingnan Chen, Afonso Tinoco, Elaine Shi, and Ke Yi. 2023. Efficient Oblivious Sorting and Shuffling for Hardware Enclaves. *Cryptology ePrint Archive* (2023).

[36] Bingyong Guo, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2020. Dumbo: Faster Asynchronous BFT Protocols.. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*.

[37] Mohammad Zahidul Hasan, Md Safiur Rahman Mahdi, Md Nazmus Sadat, and Noman Mohammed. 2018. Secure count query on encrypted genomic data. *Journal of biomedical informatics* 81 (2018), 41–52.

[38] Thomas Humphries, Rasoul Akhavan Mahdavi, Shannon Veitch, and Florian Kerschbaum. 2022. Selective mpc: Distributed computation of differentially private key-value statistics. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1459–1472.

[39] Damien Imbs and Michel Raynal. 2016. Trading off t-Resilience for Efficiency in Asynchronous Byzantine Reliable Broadcast. *Parallel Processing Letters* 26, 04 (Dec. 2016), 1650017. https://doi.org/10.1142/s0129626416500171

[40] Marek Jawurek and Florian Kerschbaum. 2012. Fault-tolerant privacy-preserving statistics. In *Privacy Enhancing Technologies: 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings 12*. Springer, 221–238.

[41] Yuzhou Jiang, Tianxi Ji, and Erman Ayday. 2022. Differentially Private Genomic Data Release For GWAS Reproducibility. *arXiv preprint arXiv:2209.06327* (2022).

[42] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What can we learn privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[43] David Kozhaya, Jérémie Decouchant, and Paulo Esteves-Verissimo. 2018. RT-ByzCast: Byzantine-resilient real-time reliable broadcast. *IEEE Trans. Comput.* 68, 3 (2018), 440–454.

[44] David Kozhaya, Jérémie Decouchant, Vincent Rahli, and Paulo Esteves-Verissimo. 2021. Pistis: an event-triggered real-time byzantine-resilient protocol suite. *IEEE Transactions on Parallel and Distributed Systems* 32, 9 (2021), 2277–2290.

[45] Lenovo. 2022. Intel Xeon Processor Scalable processors specifications. https://lenovopress.com/lp1262-intel-xeon-sp-processor-reference.

[46] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.

[47] Chao Liu, Sisi Duan, and Haibin Zhang. 2020. EPIC: Efficient Asynchronous BFT with Adaptive Security.. In *DSN*.

[48] Avradip Mandal, John C Mitchell, Hart Montgomery, and Arnab Roy. 2018. Data Oblivious Genome Variants Search on Intel SGX. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer.

[49] Sinisa Matetic, Mansoor Ahmed, Kari Kostiainen, Aritra Dhar, David Sommer, Arthur Gervais, Ari Juels, and Srdjan Capkun. 2017. ROTE: rollback protection for trusted execution. In *USENIX Security*. 1289–1306.

[50] Frank McKeen, Ilya Alexandrovich, Ittai Anati, Dror Caspi, Simon Johnson, Rebekah Leslie-Hurd, and Carlos Rozas. 2016. Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave. In *HASP*.

[51] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The honey badger of BFT protocols. In *Proceedings of the SIGSAC Conference on Computer*

and Communications Security. ACM, 31–42.
[52] Sparsh Mittal and Jeffrey S Vetter. 2015. A survey of architectural approaches for data compression in cache and main memory systems. *IEEE Transactions on Parallel and Distributed Systems* 27, 5 (2015), 1524–1536.
[53] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. 2020. A survey of published attacks on Intel SGX. *arXiv preprint arXiv:2006.13598* (2020).
[54] Jianyu Niu, Wei Peng, Xiaokuan Zhang, and Yinqian Zhang. 2022. NARRATOR: Secure and Practical State Continuity for Trusted Execution in the Cloud. In *ACM CCS*. 2385–2399.
[55] Arttu Paju, Muhammad Owais Javed, Juha Nurmi, Juha Savimäki, Brian McGillion, and Billy Bob Brumley. 2023. Sok: A systematic review of tee usage for developing trusted applications. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*. 1–15.
[56] Chandra Shekhar Pareek, Rafal Smoczynski, and Andrzej Tretyn. 2011. Sequencing technologies and genome sequencing. *Journal of applied genetics* 52 (2011), 413–435.
[57] Túlio Pascoal, Jérémie Decouchant, Antoine Boutet, and Paulo Esteves-Verissimo. 2021. DyPS: Dynamic, Private and Secure GWAS. *PETS* (2021).
[58] Túlio Pascoal, Jérémie Decouchant, Antoine Boutet, and Marcus Völp. 2023. I-GWAS: Privacy-preserving interdependent genome-wide association studies. *Proceedings on Privacy Enhancing Technologies* 1 (2023), 437–454.
[59] Donlapark Ponnoprat. 2022. Dirichlet Mechanism for Differentially Private KL Divergence Minimization. *Transactions on Machine Learning Research* (2022).
[60] Jonathan K Pritchard and Molly Przeworski. 2001. Linkage disequilibrium in humans: models and data. *The American Journal of Human Genetics* 69, 1 (2001), 1–14.
[61] Jonah Rosenblum, Juechu Dong, and Satish Narayanasamy. 2025. Confidential computing for population-scale genome-wide association studies with SECRET-GWAS. *Nature Computational Science* 5, 9 (2025), 825–835. https://doi.org/10.1038/s43588-025-00856-z
[62] Ashish P Sanil, Alan F Karr, Xiaodong Lin, and Jerome P Reiter. 2004. Privacy preserving regression modelling via distributed computation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 677–682.
[63] Karthik V Sarma, Stephanie Harmon, Thomas Sanford, Holger R Roth, Ziyue Xu, Jesse Tetreault, Daguang Xu, Mona G Flores, Alex G Raman, Rushikesh Kulkarni, Bradford J Wood, Peter L Choyke, Alan M Priester, Leonard S Marks, Steven S Raman, Dieter Enzmann, Baris Turkbey, William Speier, and Corey W Arnold. 2021. Federated learning improves site performance in multicenter deep learning without data sharing. *Journal of the American Medical Informatics Association* 28, 6 (2021), 1259–1264.
[64] Moritz Schneider, Ramya Jayaram Masti, Shweta Shinde, Srdjan Capkun, and Ronald Perez. 2022. Sok: Hardware-supported trusted execution environments. *arXiv preprint arXiv:2205.12742* (2022).
[65] Adam Smith. 2011. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. 813–822.
[66] Chrysoula Stathakopoulou, Tudor David, Matej Pavlovic, and Marko Vukolic. 2022. Mir-BFT: Scalable and Robust BFT for Decentralized Networks. *J. Syst. Res.* 2, 1 (2022).
[67] Chrysoula Stathakopoulou, Matej Pavlovic, and Marko Vukolic. 2022. State machine replication scalability made simple. In *EuroSys '22*. ACM, 17–33.
[68] Wasana Sukhumsirichart. 2018. Polymorphisms. In *Genetic Diversity and Disease Susceptibility*, Yamin Liu (Ed.). IntechOpen, Rijeka, Chapter 1.
[69] Oleksandr Tkachenko, Christian Weinert, Thomas Schneider, and Kay Hamacher. 2018. Large-scale privacy-preserving statistical computations for distributed genome-wide association studies. In *Asia CCS*.
[70] Florian Tramèr, Zhicong Huang, Jean-Pierre Hubaux, and Erman Ayday. 2015. Differential Privacy with Bounded Priors: Reconciling Utility and Privacy in Genome-Wide Association Studies. In *CCS*, Indrajit Ray, Ninghui Li, and Christopher Kruegel (Eds.). ACM, 1286–1297.
[71] Chia-Che Tsai, Donald E Porter, and Mona Vij. 2017. Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX. In *USENIX ATC*.
[72] Caroline Uhlerop, Aleksandra Slavković, and Stephen E Fienberg. 2013. Privacy-preserving data sharing for genome-wide association studies. *The Journal of privacy and confidentiality* 5, 1 (2013), 137.
[73] Giuliana Santos Veronese, Miguel Correia, Alysson Neves Bessani, Lau Cheuk Lung, and Paulo Verissimo. 2011. Efficient byzantine fault-tolerance. *IEEE Trans. Comput.* 62, 1 (2011), 16–30.
[74] Duy Vu and Aleksandra Slavkovic. 2009. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 138–143.
[75] Shuang Wang, Yuchen Zhang, Wenrui Dai, Kristin Lauter, Miran Kim, Yuzhe Tang, Hongkai Xiong, and Xiaoqian Jiang. 2016. HEALER: homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS. *Bioinformatics* 32, 2 (2016), 211–218.
[76] Weili Wang, Sen Deng, Jianyu Niu, Michael K Reiter, and Yinqian Zhang. 2022. ENGRAFT: Enclave-guarded Raft on Byzantine Faulty Nodes. In *ACM CCS*. 2841–2855.
[77] Akito Yamamoto and Tetsuo Shibuya. 2021. More practical differentially private publication of key statistics in GWAS. *Bioinformatics Advances* 1, 1 (2021), vbab004.
[78] Abbas Yazdinejad, Ali Dehghantanha, Hadis Karimipour, Gautam Srivastava, and Reza M. Parizi. 2024. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. *IEEE Transactions on Information Forensics and Security* 19 (2024), 6693–6708.
[79] Emre Yilmaz, Tianxi Ji, Erman Ayday, and Pan Li. 2022. Genomic data sharing under dependent local differential privacy. In *Proceedings of the twelfth ACM conference on data and application security and privacy*. 77–88.
[80] Fei Yu, Stephen E Fienberg, Aleksandra B Slavković, and Caroline Uhler. 2014. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of biomedical informatics* 50 (2014), 133–141.
[81] Fei Yu, Michal Rybar, Caroline Uhler, and Stephen E Fienberg. 2014. Differentially-private logistic regression for detecting multiple-SNP association in GWAS databases. In *International Conference on Privacy in Statistical Databases*. Springer, 170–184.
[82] Yuchen Zhang, Wenrui Dai, Xiaoqian Jiang, Hongkai Xiong, and Shuang Wang. 2015. Foresee: Fully outsourced secure genome study based on homomorphic encryption. 15, 5 (2015), S5.
[83] Liangrong Zhao, Jérémie Decouchant, Joseph K Liu, Qinghua Lu, and Jiangshan Yu. 2024. Trusted hardware-assisted leaderless byzantine fault tolerance consensus. *IEEE Transactions on Dependable and Secure Computing* 21, 6 (2024), 5086–5097.
[84] Liangrong Zhao, Hans Schmiedel, Qin Wang, and Jiangshan Yu. 2024. Janus: Enhancing Asynchronous Common Subset with Trusted Hardware. In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE, 488–504.
[85] Wenting Zheng, Raluca Ada Popa, Joseph E. Gonzalez, and Ion Stoica. 2019. Helen: Maliciously Secure Coopetitive Learning for Linear Models. *CoRR* abs/1907.07212 (2019). arXiv:1907.07212