Analysis and Prevention of Averaging Attacks Against Obfuscation Protocols

Kilian Becher[1(✉)], J. A. Gregor Lagodzinski[2], Javier Parra-Arnau[3,4], and Thorsten Strufe[3]

[1] TU Dresden, Dresden, Germany
kilian.becher@tu-dresden.de
[2] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
gregor.lagodzinski@hpi.de
[3] Karlsruhe Institute of Technology, Karlsruhe, Germany
{javier.parra-arnau,thorsten.strufe}@kit.edu
[4] Universitat Politécnica de Catalunya, Barcelona, Spain

Abstract. Verification and traceability of supply-chain data is a common example for public analysis of confidential data. Finding the correct balance between confidentiality and utility often is anything but trivial. In order to ensure confidentiality and thus protect companies' competitive advantages, existing approaches employ probabilistic output obfuscation. However, it is known that this form of obfuscation might render a system subject to averaging attacks. In these attacks, an adversary repeatedly queries for the same analysis and combines the probabilistic outputs, thus implementing an estimator that eliminates the obfuscation. A clear picture on the performance of such attacks is missing, information that is crucial for mitigating averaging attacks.

Our contributions are threefold: First, using an existing supply-chain verification protocol (RVP) as a particularly efficient example of protocols with output obfuscation, we extensively analyze the risk posed by averaging attacks. We prove rigorously that such attacks perform exceptionally well if obfuscation is based on random values sampled independently in every query. We generalize our analysis to all protocols that employ probabilistic output obfuscation. Second, we propose the paradigm of data-dependent deterministic obfuscation (D^3O) to prevent such attacks. Third, we present mRVP, a D^3O-based version of RVP, and empirically demonstrate practicality and effectiveness of D^3O. The results show that our mitigations add negligible runtime overhead, do not affect accuracy, and effectively retain confidentiality.

Keywords: Averaging attacks · output obfuscation · confidentiality-utility tradeoff, runtime bounds, homomorphic encryption

1 Introduction

Analysis of confidential data is a ubiquitous problem of our age. Prominent examples comprise the computation of salary statistics from human-resource

M. Tibouchi and X. Wang (Eds.): ACNS 2023, LNCS 13905, pp. 451–475, 2023.
https://doi.org/10.1007/978-3-031-33488-7_17

data [17], extraction of demographic statistics from census data [3,13], processing patient records for pharmaceutical research [23], cross-company benchmarks [4, 24], and supply-chain verification [2,5,10,28,36].

Many existing solutions to such problems rely on privacy-enhancing technologies like (fully) homomorphic encryption to ensure the confidentiality of inputs and intermediate results. Fully homomorphic encryption enables addition and multiplication of encrypted data. While addition and multiplication also imply subtraction, division of encrypted data remains anything but trivial.

However, division is crucial for many of the above-mentioned scenarios, such as ratio computation for supply-chain verification. An efficient *Ratio Verification Protocol* (RVP) was proposed in [5]. The protocol performs privacy-preserving division by combining output obfuscation with client-aided computation. The output obfuscation function includes additive and multiplicative blinding. RVP computes division orders of magnitude faster than previous protocols due to the fact that its circuit has a multiplicative depth of 1. We describe RVP in detail in Sect. 2.3. Its efficiency enables numerous use cases for disguised division in the first place. For example, [5] applies RVP to a supply-chain verification scenario where it enables the computation of the ratio between different kinds of cobalt ore. Besides that, RVP can be applied to any scenario that requires the computation of ratios of confidential values, such as the verification of the amount of gold in an alloy or the percentage of fair-trade palm oil in groceries. Unfortunately, RVP's construction for output obfuscation turns out to be vulnerable to *averaging attacks*, which repeat similar computations and average out the random blinding values to obtain the confidential inputs. Moreover, the randomness used for additive blinding in the output obfuscation function is subject to leaking the difference between the dividend and the divisor, which in combination with their ratio, yields the exact dividend and divisor.

We extensively demonstrate and formally analyze averaging attacks and propose suitable mitigations. As a result, we propose mRVP, a modified version of RVP that effectively prevents these attacks, not just for its initial purpose of verifying ratios of commodities in consumer products. Instead, mRVP can be applied to any use case for releasing quotients over homomorphically encrypted data, i.e., privacy-preserving computations where the result is a quotient. Moreover, our analysis and mitigations are universal and can be applied to all classes of privacy-preserving protocols with probabilistic output obfuscation.

1.1 Contributions

Our contributions are threefold. First, we demonstrate averaging attacks against RVP in order to infer confidential information through repeated queries. We then conduct a formal analysis of the risk of leaking information through repeated queries in RVP and generalize our findings to all protocols that use probabilistic output obfuscation. We find that even in a general scenario, an adversary learns the confidential inputs exponentially fast in the number of queries if the randomness used for obfuscation is sampled independently. For instance, already 15 queries can be sufficient to infer confidential inputs with probability at least

99% and confidence 1. We further demonstrate for RVP how the additive blinding used for hiding the confidential dividend and divisor can be removed with a single query. Second, building on the findings of Denning [13] and Francis et al. [17], we propose to use data-dependent deterministic randomness to mitigate averaging attacks. Aiming for utility and confidentiality, we build on the information-theoretic definition of *uncertainty* [12] and introduce the following paradigm.

Definition 1 (Data-Dependent Deterministic Obfuscation (D^3O)). *Given a secret x and a sequence of pseudorandom values $\bar{r} = (r_1, \ldots, r_t)$, an* obfuscation function *takes x and $\bar{r}$ as input to add uncertainty to x. An obfuscation function is* data-dependent deterministic *if the values in $\bar{r}$ are computed by a one-way function f, such that f takes the value x as input (data-dependent) and f always returns the same output for the same x (deterministic).*

We further propose a first instantiation of D^3O. Third, we present mRVP, a modified version of RVP that effectively and efficiently prevents the described attacks based on D^3O and further modifications. We empirically investigate mRVP in the real-world scenario of cobalt supply-chain verification. Compared to the original RVP protocol presented in [5] as well as a second baseline, we find that mRVP achieves high accuracy and effectively mitigates averaging attacks. The necessary modifications only add negligible runtime overhead.

1.2 Related Work

To the best of our knowledge, the idea of using data-dependent randomness to prevent averaging out (zero-mean) random values with repeated queries was first proposed by Denning in [13]. She suggests a new inference control, referred to as *Random Sample Queries*, to protect confidentiality of data in query-based systems. She provides a formal analysis of averaging attacks for repeated similar queries. In order to mitigate certain classes of attacks, she suggests distorting computed outputs, e.g., via rounding or by adding pseudorandom values with mean zero. She points out that using the same pseudorandom value for similar results, by making the noise data-dependent, is preferable. Most notably, Denning's formal analysis is restricted to systems that use Random Sample Queries.

In [17], Francis et al. propose Diffix, an SQL proxy that adds data-dependent *sticky* noise to query results in order to hinder complex analyses of outputs. The security and shortcomings of Diffix with respect to certain kinds of attacks as well as potential defenses were investigated in [7,18]. Furthermore, their randomized algorithms are restricted to query-based systems and SQL proxies.

An analysis of the success probability of averaging attacks against *bounded perturbation algorithms* is presented by Asghar et al. [3]. They apply their attacks to *TableBuilder*, a tool for analysing census data. TableBuilder performs the same perturbation for queries whose responses involve the same set of individuals. This determinism is modeled through a noise dictionary. Notably, they point out the NP-hardness of query auditing for detecting maliciously crafted queries [25]. However, their considerations are restricted to counting queries.

Additionally, we note that the overall concept of obfuscation by adding noise to data is widely used in differential privacy [15].

The aforementioned systems and methods focus on scenarios where data analysts aim to query confidential data in a privacy-preserving form. Our paradigm of data-dependent deterministic obfuscation is not limited to such scenarios and, therefore, is not limited in terms of query syntax. Furthermore, mRVP, which we use in Sect. 5 to demonstrate applicability and effectiveness of D^3O, targets confidentiality of single attributes rather than privacy of data owners. That is, it does not aim to hide whose confidential inputs contributed to a computation.

The chance of inferring confidential data from probabilistically obfuscated outputs has been discussed in the past. Kerschbaum [24] empirically analyzes the leakage of a homomorphic-encryption-based system where participants have access to multiple samples of additively and multiplicatively blinded data. He emphasizes the difficulty of estimating the probability of a particular leakage and, therefore, instead empirically demonstrates substantial leakage in terms of lost entropy. We complement this empirical analysis with a formal analysis for averaging attacks. Pipernik et al. [32] studied the potential of inverse optimization in the well-known JELS model, which does not use output obfuscation. Similar to our results, they found that a party can infer knowledge about the secret input of another party even though the computations are performed in a secure fashion.

2 Preliminaries

2.1 Homomorphic Encryption

Asymmetric cryptosystems are tuples $\mathcal{CS} = (G, E, D)$ consisting of a probabilistic key-generation algorithm $G(\cdot)$ that generates pairs of a (public) encryption key pk and a (secret) decryption key sk, a (probabilistic) encryption algorithm $E(\cdot)$, and a decryption algorithm $D(\cdot)$. We denote the plaintext and ciphertext space by $\mathcal{M}$ and $\mathcal{C}$, respectively. Homomorphic encryption (HE) schemes provide at least one operation "$\circ$" on $\mathcal{C}$ that corresponds to an operation "$\bullet$" on $\mathcal{M}$ such that $E(m_1, pk) \circ E(m_2, pk)$ results in an encryption of the operation $m_1 \bullet m_2$ for two plaintexts m_1, m_2. We fomalize this as follows.

$$D(E(m_1, pk) \circ E(m_2, pk), sk) = m_1 \bullet m_2$$

Partially homomorphic encryption (PHE) schemes enable either addition [31] or multiplication [33] of underlying plaintexts. Fully homomorphic encryption (FHE) schemes [9,11,16,20] offer both addition and multiplication and, thus, enable evaluation of arbitrary arithmetic functions over confidential data.

2.2 Proxy Re-encryption

Re-encryption is an operation that transforms ciphertexts $c_i = E(m, pk_i)$ encrypted under one key into ciphertexts $c_j = E(m, pk_{j \neq i})$ encrypted under a different key while preserving the underyling plaintext. Proxy re-encryption

(PRE) [6] ensures confidentiality even if this transformation is performed by an untrusted party, i.e., without intermediate decryption. A default way to construct PRE from FHE is described in Gentry's seminal work [19].

2.3 Overview of RVP

The RVP protocol uses a tree-like representation of the supply-chain transactions graph, which is considered public industry knowledge [5]. Confidential supply-chain data is homomorphically encrypted at transaction time under the supply-chain participant P_j's key and stored in a public distributed ledger DL. Upon request by a consumer C, a central party R reads the encrypted data from DL, re-encrypts the ciphertexts to bring them under the same key, and homomorphically aggregates the encrypted data to compute encrypted sums of different kinds of ingredients, e.g., artisanally mined (ASM) and industrially mined (LSM) cobalt. The central party uses additive and multiplicative blinding for output obfuscation in order to protect confidential aggregates. This blinding is based on two numbers $0 < r_2 \ll r_1$ that are sampled uniformly at random for each request. The encrypted, obfuscated aggregates are then decrypted by a separate decryption party D and returned to the requesting consumer C as $S_{ASM} = (\Sigma_{ASM}) \cdot r_1 + r_2$ and $S_{Total} = (\Sigma_{ASM+LSM}) \cdot r_1 + r_2$. Then, C computes

$$\rho = \frac{S_{ASM}}{S_{Total}}, \tag{1}$$

which causes the random blindings r_1, r_2 to cancel out with negligible error and yields a close approximation of the ratio.

3 Averaging Attacks Against Obfuscation Protocols

In this section, we demonstrate how the probabilistic output obfuscation of RVP can be exploited to infer confidential information through averaging attacks. We complement this with a formal analysis of the risk of averaging attacks against RVP and all other protocols that employ probabilistic output obfuscation based on the standard approaches of additive and multiplicative blinding of aggregates.

We show that these blinding functions are highly vulnerable to attacks based on results of the concentration around the expected value. In particular, we prove rigorously that information about the confidential aggregates gets revealed exponentially fast in the number of queries. This applies to all scenarios that may arise as long as each query draws the blinding values independent from values drawn in other queries. Our argumentation follows a case distinction depending on which blinding function is used. First, we study the case of a target function which outputs a fraction of aggregates, as in RVP, that is multiplicatively blinded. Second, we argue on the case of general target functions with obfuscation by additive blinding. Third, we argue on the case that both additive and multiplicative blinding may be employed for general target functions. We generalize our findings to all protocols that use additive or multiplicative blinding, a

combination of both, or even random noise with changing distribution. For a positive integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$ in accordance with standard notation.

3.1 Adversary Model

For the averaging attacks described in this section, we assume honest-but-curious adversaries [27]. They act as consumers and query the obfuscating central party R to run verifications. The adversary may collude with a subset of producers aiming to infer non-colluding producers' confidential data.

3.2 Averaging Attacks Against RVP

The adversary runs averaging attacks by repeatedly requesting ratio verifications for the same product and caches the results. Given a sufficient amount of verification results, the adversary combines the obfuscated results in order to reconstruct the random blindings and infer the plaintext aggregates. Given these aggregates and the inputs of colluding producers, the adversary can infer information about the confidential inputs of non-colluding producers. The remainder of this section focuses on the question: How many repeated verification requests are necessary to gain sufficient knowledge about the blinding values?

We assume a set of n supply-chain participants P_j with set of confidential information $\{x_{\mathrm{A}_j}, x_{\mathrm{B}_j}\}$, e.g., ASM and ASM+LSM cobalt amounts. These parties are honest but curious, which yields the possibility that a party - w.l.o.g. assumed to be P_1 - uses the output and its private information $\{x_{\mathrm{A}_1}, x_{\mathrm{B}_1}\}$ in order to obtain additional information on $\{x_{\mathrm{A}_j}, x_{\mathrm{B}_j}\}$ for any $j \in [n]$ with $j \neq 1$.

The output of RVP is given by a modification, e.g., ρ, of the target function

$$f(\bar{x}_{\mathrm{A}}, \bar{x}_{\mathrm{B}}) = \frac{\sum_j x_{\mathrm{A}_j}}{\sum_j x_{\mathrm{B}_j}}, \tag{2}$$

where standard ways of modifications (blinding) were combined to render a reverse optimization infeasible. We study these modifications individually in the following, where we distinguish between a deterministically correct fraction and a random output that approximates the correct fraction with high probability.

In order to study the possibility for reverse optimization, we assume the minimal case of two input producing parties P_1 and P_2. The reason for this minimal case is its worst-case characteristic. The case $n = 1$ is of no interest since every party has full information about their own input. We later argue on the generalization of our findings to the general case with $n > 2$ parties.

A Special Case - Removing the Additive Blinding in RVP. Before we continue with our formal analysis of averaging attacks, we show that the construction of its output obfuscation function renders RVP subject to a special case of this class of attacks that does not require repeated queries.

Recall that the output of RVP is the ratio between a blinded dividend and a blinded divisor. Both are multiplicatively blinded with the same r_1 and additively blinded with the same r_2 such that $0 < r_2 \ll r_1$, i.e., $y = \frac{x_1 \cdot r_1 + r_2}{x_2 \cdot r_1 + r_2}$. Using the same r_1 ensures that the ratio of x_1 and x_2 is preserved despite the obfuscation. In contrast, r_2 is supposed to prevent attacks that remove the multiplicative blinding through factorization. However, choosing the same r_2 for dividend and divisor allows for a simple attack. An adversary can subtract the blinded divisor from the blinded dividend, which causes the r_2 to cancel out. This yields the dividend and the divisor that are now only multiplicatively blinded and therefore vulnerable to factorization. We formally describe this attack in Appendix A.2.

This attack can be effectively prevented by choosing different additive blinding values for the dividend and the divisor, i.e., $y = \frac{x_1 \cdot r_1 + r_2}{x_2 \cdot r_1 + r_3}$ such that $0 < r_2, r_3 \ll r_1$.

3.3 Deterministically Correct

We focus on a target function with output that is a fraction of aggregates as in Eq. (2), skip the trivial case of un-obfuscated outputs, and start our analysis with the case of multiplicatively blinded fractions.

Multiplicative Blinding. We turn towards the usage of a random multiplicative factor r used to blind the information in the two sums $\sum_j x_{\mathrm{A}_j}$ and $\sum_j x_{\mathrm{B}_j}$. There is a large variety of possibilities to generate the factor r randomly. We initialize the study with the choice where in every query k the factor r_k is taken uniformly at random out of $[\nu]$ with $\nu \in \mathbb{Z}^+_{>0}$, i.e., r_k is a positive integer. We argue how changing the distribution affects our findings afterwards.

Party P_1 can use the knowledge about the protocol to infer partial knowledge in a straight-forward manner. Let r be the random factor used to blind the sums $\sum_j x_{\mathrm{A}_j}$ and $\sum_j x_{\mathrm{B}_j}$, where we denote by $A(r)$ the value $r \cdot \sum_j x_{\mathrm{A}_j}$ and by $B(r)$ the value $r \cdot \sum_j x_{\mathrm{B}_j}$. In the setting under study, Party P_1 has access to the blinded sums $A(r)$ and $B(r)$. Given access to the factor r, party P_1 can infer the precise private input of P_2. The factor r is an integer in $[\nu]$ and a common divisor of $A(r)$ and $B(r)$. Computing D, i.e., the set of common divisors of $A(r)$ and $B(r)$, P_1 has a chance of $|D|^{-1}$ to draw r uniformly at random out of D. This motivates a choice of a very large positive integer ν in order to generate a multiplicative factor r with a large set of artificial divisors used to blow up the set D. However, even a large set D is not enough to hinder P_1 from inferring almost perfect knowledge when P_1 can repeat the query. In order to show the existence of a very effective procedure used for reverse optimization, we start with the following classic result by Mertens [30].

Lemma 1. *Let ν be a positive integer. If r_1 and r_2 are taken uniformly at random out of $[\nu]$, then the probability that r_1 and r_2 are coprime is at least $6 \cdot \pi^{-2}$.*

By Lemma 1, we observe that it is unlikely to produce two random multiplicative factors r_1 and r_2 with multiple common divisors. Every multiplicative

factor r_k yields the greatest common divisor D_k of $A(r_k)$ and $B(r_k)$, which is $r_k \cdot \gcd\left(\sum_j x_{\mathrm{A}_j}, \sum_j x_{\mathrm{B}_j}\right)$. If two factors r_1 and r_2 are coprime, then the $\gcd(D_1, D_2)$ is equal to the greatest common divisor of the two unmodified sums $\sum_j x_{\mathrm{A}_j}$ and $\sum_j x_{\mathrm{B}_j}$. With access to this greatest common divisor and D_1, party P_1 can then calculate r_1 by simple division.

Consequently, P_1 gets access to r_1 and this is sufficient to infer the private input $\{x_{\mathrm{A}_2}, x_{\mathrm{B}_2}\}$ of party P_2. In general, the $\gcd(D_1, D_2)$ is equal to the product of $\gcd(r_1, r_2)$ and $\gcd\left(\sum_j x_{\mathrm{A}_j}, \sum_j x_{\mathrm{B}_j}\right)$. This allows for an iterative algorithm by consecutively requesting sums modified by a new independent factor r_k, computing the greatest common divisor of consecutive query outputs until only the greatest common divisor of the sums remains. More precisely, starting with $D^* = D_1$ at step $i = 1$, party P_1 requests at step $i + 1$ the modified sums $A(r_{i+1})$ and $B(r_{i+1})$, computes D_{i+1}, and computes the $\gcd(D^*, D_{i+1})$ stored as the new D^*. If at least two multiplicative factors are coprime, then D^* is equal to the greatest common divisor of the two sums.

It remains to study how many times P_1 has to repeat the query in order to obtain a tuple of coprime multiplicative factors.

Proposition 1. *For a positive integer ν, let $r_1, r_2, \ldots, r_\kappa$ be a sequence of integers taken independently uniformly at random out of $[\nu]$. Given, for every $k \in [\kappa]$, the sum $r_k \cdot \sum_j x_{\mathrm{A}_j}$ denoted by $A(r_k)$, the sum $r_k \cdot \sum_j x_{\mathrm{B}_j}$ denoted by $B(r_k)$, and its private input $\{x_{\mathrm{A}_1}, x_{\mathrm{B}_1}\}$, party P_1 can infer the private input $\{x_{\mathrm{A}_2}, x_{\mathrm{B}_2}\}$ of party P_2 with probability at least*

$$1 - \exp(-\pi^{-2} \cdot (3 \cdot (\kappa - 1))).$$

For the proof of Proposition 1, we apply the well-known Chernoff Bounds.

Theorem 1 (Chernoff Bound). *Let $X_1, \ldots, X_k$ be a sequence of independent and identically distributed Bernoulli trials with $\mathbb{P}[X_i = 1] = p$. Then, for every $\delta \in [0; 1]$,*

$$\mathbb{P}\left[\sum_{i=1}^{k} X_i \leqslant (1 - \delta) \cdot p \cdot k\right] \leqslant \exp\left(-\frac{\delta^2}{2} \cdot p \cdot k\right).$$

Proof (Proof of Proposition 1). We recall the definition of D^* and that it remains to estimate the number of trials κ needed to obtain an instance k in which r_1 and r_2 are coprime. This serves as an upper bound on the number of trials needed such that D^* is equal to the greatest common divisor of the two sums. With this information, party P_1 can infer the random factor r_1 and, consequently, the private input of party P_2.

For every k in $[\kappa]$ with $k > 1$, we define the Bernoulli trial $X_k \in \{0, 1\}$ with $X_k = 1$ if r_1 and r_k are coprime, else $X_k = 0$. By Lemma 1, we have that $\mathbb{P}[X_k = 1] \geqslant 6 \cdot \pi^{-2}$.

We obtain a sequence of $\kappa - 1$ independent Bernoulli trials, which are all unsuccessful if and only if their sum is 0. By Theorem 1 with $\delta = 1$, we deduce

$$\mathbb{P}\left[\sum_{i=1}^{\kappa-1} X_i \leqslant 0\right] \leqslant \exp\left(-\frac{6 \cdot (\kappa - 1)}{2 \cdot \pi^2}\right).$$

This serves as a valid upper bound on the probability that D^* is equal to the greatest common divisor of the two sums because we argued on the greatest common divisor of two elements contrary to all κ elements. The result follows from the fact that the probability under study is lower bounded by the probability to have at least one k with $X_k > 1$. □

By Proposition 1, we observe that the probability of an error in the procedure drops exponentially fast, e.g., already after $\kappa = 9$ queries the probability that P_1 inferred the correct input $\{x_{\mathrm{A}_2}, x_{\mathrm{B}_2}\}$ of party P_2 is larger than 0.91 and after $\kappa = 16$ queries the probability is larger than 0.99.

As a final remark regarding multiplicative blinding, we argue on the impact of changing the distribution that the factor r is taken from. So far we let r be taken uniformly from $[\nu]$, a finite subspace of the positive integers, where the knowledge about the protocol was sufficient to bypass the blinding function by observing the set of common divisors. As long as the factors are integers and drawn independently, the results remain valid regardless of the distribution. The reason is Dirichlet's classic result stating that the density of ordered coprime pairs (r_1, r_2) in $\mathbb{N}^2$ is asymptotically $6 \cdot \pi^{-2}$ (see e.g. Hardy and Wright [21]). Also a consideration of rational values $r \in \mathbb{Q}$ does not introduce meaningful countermeasures as party P_1 can artificially scale the $A(r), B(r)$ by a large enough factor such that the effect is as if r were an integer.

3.4 Random Output

We now consider averaging attacks on arbitrary output functions. The second class of modifications used in RVP uses randomly generated noise to blind the correct information from an adversary. More precisely, if the output of the protocol is the function$f(\bar{x})$, where $\bar{x}$ is any ordered set of private information, the protocol outputs $f(\bar{x}) + s$, where s is a random variable drawn from some distribution such that it does not perturb the function too much. Obvious choices for the distributions are the normal (or Gaussian) distribution and the Laplace distribution, which can both be tailored easily such that the distribution is concentrated around its expected value. However, this desired concentration can be used by an adversary to infer knowledge about private information in a very similar way as in the previous subsection by repeatedly querying the protocol. This information's probability of correctness increases then exponentially fast in the number of queries, rendering the modification (almost) useless. We will show these results in the following.

The normal distribution with expected value $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$ is denoted by $\mathcal{N}(\mu, \sigma^2)$. We denote a random variable X to be drawn from $\mathcal{N}(\mu, \sigma^2)$

by $X \sim \mathcal{N}(\mu, \sigma^2)$. Similarly, the Laplace distribution with expected value $\mu \in \mathbb{R}$ and scale $b \in \mathbb{R}_{>0}$ is denoted by $\mathrm{Lap}(\mu, b)$. We denote a random variable X to be drawn from $\mathrm{Lap}(\mu, b)$ by $X \sim \mathrm{Lap}(\mu, b)$.

We assume for the sake of clarity that the distributions under study are chosen with expected value $\mu = 0$, which appears plausible as it does not introduce a large perturbation of the function value $f(\bar{x})$. The results are not affected by this assumption because one can replace the random variable X with the random variable $X - \mu$.

We reasoned in the previous subsection that the adversary P_1 can infer knowledge of the private input of party P_2 once P_1 has access to $f(\bar{x})$. The adversary P_1 can infer knowledge of the private input of party P_2 with increasing correctness by initializing repeated queries, forcing the protocol to generate new random additive noise which concentrates around the expected value $\mu = 0$. In particular, let P_1 query the protocol $\kappa > 0$ times and $f(\bar{x}) + X_k$ be the output of the k-th query for $k \in [\kappa]$. Party P_1 observes the average $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} (f(\bar{x}) + X_k) = f(\bar{x}) + \kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k$. If $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k$ is highly concentrated around the expected value μ equal to 0, then P_1 infers knowledge of the private input of P_2 with small error.

Observation 2. *Let $f(\bar{x})$ be a functional value of private inputs of party P_1 and P_2. Let $X_1, \ldots, X_\kappa$ be a sequence of independent and identically distributed random variables with expected value $\mu = 0$. Given $f_k = f(\bar{x}) + X_k$ for $1 \leqslant k \leqslant \kappa$ and any real number $t > 0$, party P_1 can infer $f(\bar{x})$ to be in the interval $(f(\bar{x}) - t, f(\bar{x}) + t)$ with probability*

$$\mathbb{P}\left[\left|\kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k\right| < t\right]$$

which is

$$1 - \mathbb{P}\left[\left|\kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k\right| \geqslant t\right].$$

The obvious choices of a normal distribution and a Laplace distribution (see e.g. [14,15]) belong to a large class of distributions called *sub-exponential* and *sub-Gaussian*, which allow for a concentration bound known as *Hoeffding's inequality* (see e.g. [35]). In the following, we provide a full technical argumentation culminating in Corollary 2; the very curious reader may skip to this corollary.

Definition 2. *A random variable X with finite expected value μ is called* sub-exponential *if there exists a pair of non-negative parameters (ν, b) such that*

$$\mathbb{E}\left[\exp(\lambda \cdot (X - \mu))\right] \leqslant \exp\left(\frac{\nu^2 \cdot \lambda^2}{2}\right), \quad \text{for all } |\lambda| < \frac{1}{b}. \tag{3}$$

In the case that Eq. (3) holds for all λ, then X is called sub-Gaussian *and has parameter ν.*

The Hoeffding bounds are concentration bounds of sub-Gaussian and sub-exponential random variables. They can be found in [35]. Utilizing these, a standard application of the union bound yields the following well-known corollary, which states concentration bounds of the sum of identically distributed random variables provided that these are either sub-Gaussian or sub-exponential.

Corollary 1. *Let $X_1, \ldots, X_\kappa$ be a sequence of independent and identically distributed random variables with finite expected value μ. Further, let X be the random variable with $X = \sum_{k=1}^{\kappa} X_k$.*

1. *If X_k is sub-Gaussian with parameter ν, then for all $t \geqslant 0$*

$$\mathbb{P}\left[|X| \geqslant \kappa \cdot (\mu + t)\right] \leqslant 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa}{2 \cdot \nu^2}\right);$$

2. *if X_k is sub-exponential with parameters (ν, b), then*

$$\mathbb{P}\left[|X| \geqslant \kappa \cdot (\mu + t)\right] \leqslant \begin{cases} 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa}{2 \cdot \nu^2}\right), & \text{if } 0 \leqslant t \leqslant \frac{\nu^2}{\kappa \cdot b}; \\ 2 \cdot \exp\left(-\frac{t \cdot \kappa}{2 \cdot b}\right), & \text{for } t \geqslant \frac{\nu^2}{\kappa \cdot b}. \end{cases}$$

The following lemma states the well-known fact that normal distributed random variables are sub-Gaussian and Laplace distributed random variables are sub-exponential. For the sake of completeness, we provide a short proof.

Lemma 2. *Let X be a random variable.*

1. *If $X \sim \mathcal{N}(\mu, \sigma^2)$, then X is sub-Gaussian with parameter ν that is equal to σ;*
2. *if $X \sim \mathrm{Lap}(0, b)$, then X is sub-exponential with parameters (ν, b'), where $\nu = 2 \cdot b$ and $b' = \sqrt{2} \cdot b$.*

Proof. The first case follows immediately from the definition of a sub-Gaussian random variable. For the second case, we observe that $(1-x)^{-1} \leqslant 1 + 2 \cdot x$ holds for every x with $0 \leqslant x \leqslant 2^{-1}$. Since $1 + x \leqslant \exp(x)$ holds for all real numbers x, we obtain for the pair of real numbers b and λ with $0 \leqslant b^2 \cdot \lambda^2 \leqslant 2^{-1}$

$$\frac{1}{1 - b^2 \cdot \lambda^2} \leqslant \exp\left(2 \cdot \lambda^2 \cdot b^2\right) = \exp\left(\frac{\nu^2 \cdot \lambda^2}{2}\right). \tag{4}$$

By $X \sim \mathrm{Lap}(0, b)$, we have that, for $|\lambda| < b^{-1}$, the moment-generating function satisfies $\mathbb{E}\left[\exp(\lambda \cdot X)\right] = (1 - b^2 \cdot \lambda^2)^{-1}$ (see [26, Equation (2.1.10)]). By Eq. (4), we conclude, for $|\lambda| < (\sqrt{2} \cdot b)^{-1}$, that the moment generating function satisfies

$$\mathbb{E}\left[\exp(\lambda \cdot X)\right] \leqslant \exp\left(\frac{\nu^2 \cdot \lambda^2}{2}\right).$$

□

Corollary 1 in conjunction with Lemma 2 now yields the following.

Corollary 2. *Let $X_1, \ldots, X_\kappa$ be a sequence of independent and identically distributed random variables with expected value μ equal to 0. Let X be the random variable with $X = \kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k$.*

1. If $X_k \sim \mathcal{N}(0, \sigma^2)$ then

$$\mathbb{P}\left[|X| \geqslant t\right] \leqslant 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa}{2 \cdot \sigma^2}\right);$$

2. if $X_k \sim \mathrm{Lap}(0, b)$ then

$$\mathbb{P}\left[|X| \geqslant t\right] \leqslant \begin{cases} 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa}{8 \cdot b^2}\right), & \text{if } 0 \leqslant t \leqslant \frac{2 \cdot \sqrt{2} \cdot b}{\kappa}; \\ 2 \cdot \exp\left(-\frac{t \cdot \kappa}{2 \cdot \sqrt{2} \cdot b}\right), & \text{for } t \geqslant \frac{2 \cdot \sqrt{2} \cdot b}{\kappa}. \end{cases}$$

Example 1. Let $X_k \sim \mathcal{N}(0, 1)$ and the desired maximal error be smaller than t equal to 1, then after $\kappa = 10$ queries, the probability for an error of at least 1 is at most $2 \cdot \exp(-5) < 0.014$, and thus party P_1 observes an error smaller than 1 with probability at least 0.986. For an additive noise $X_k \sim \mathrm{Lap}(0, b)$, we have to be a bit more careful as to which bound applies. For instance, with b equal to 1 and the same maximal error bound t after κ equal to 10 queries, we have to apply the second bound, which yields a less powerful probability bound of $2 \cdot \exp(-5 \cdot (\sqrt{2})^{-1}) < 0.059$. However, after κ equal to 15 queries, the probability for an error of at least 1 is smaller than 0.01, and thus party P_1 observes an error smaller than 1 with probability at least 0.99.

General Practical Scenario: Bounded Noise. The probabilities computed in Corollary 2 and in Proposition 1 depend on explicit characteristics of the used probability distribution that the noise is drawn from. There is a property shared by every random noise applied in any practical scenario: The noise X lies in a finite interval $[a; b]$ in $\mathbb{R}$. Boundedness is a strong enough property to obtain concentration bounds. The reason is that a bounded random variable is sub-Gaussian, a result known as *Hoeffding's lemma* (see e.g. [29]).

From Hoeffding's lemma, Corollary 2, and a centralization $X' = X - \mu$, we obtain the following result, which is referred to as *Hoeffding's inequality.*

Corollary 3 (Hoeffding's Inequality). *Let $X_1, \ldots, X_\kappa$ be a sequence of identically distributed independent random variables with finite range $[a; b]$ contained in $\mathbb{R}$ and expected value μ. Let X be the random variable $X = \kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k$, then for all t larger than 0*

$$\mathbb{P}\left[|X| \geqslant (\mu + t)\right] \leqslant 2 \cdot \exp\left(-\frac{2 \cdot t^2 \cdot \kappa}{(b - a)^2}\right).$$

For instance, taking random noise X_k uniformly at random in $[0; 2^\ell]$ party P_1 needs κ equal to $\frac{5}{2} \cdot 2^{2 \cdot \ell}$ queries in order to observe the same error of at least t equal to 1 with the same probability bound of at most $2 \cdot \mathrm{e}^{-5}$ as it did already

after κ equal to 10 queries when $X_k \sim \mathcal{N}(0,1)$. The Laplace and the normal distribution are usually much stronger concentrated around their expected value. Nonetheless, asymptotically the behaviour is the same: The probability that the average is in a given confidence interval around the mean grows exponentially fast in κ.

The following example illustrates how a scenario of uniform noise translates to the setting studied.

Example 2. Assume a protocol hides a function value $f(\bar{x})$ with a multiplicative noise X, where X is drawn uniformly at random from $[0; 2^\ell]$ with fixed non-negative integer ℓ. Moreover, assume party P_1 has knowledge of the range $[a; b] \subsetneq \mathbb{R}_{\geqslant 0}$ of the function $f(\cdot)$ for any possible input. Then, the output $Y = f(\bar{x}) \cdot X$ is a bounded random variable with range $[0; 2^\ell \cdot b]$ and expected value $f(\bar{x}) \cdot \mu$, where μ is the expected value of X.

Combination of Noise. In order to blind a function value $f(\bar{x})$, a protocol generates two independently generated random variables X and X' and outputs the random variable Y, where $Y = X \cdot f(\bar{x}) + X'$. Due to the additive noise X', the attacks studied in Sect. 3.3 are invalid. However, in any practical application both X and X' have to be bounded random variables and thus Y is also a bounded random variable. With knowledge of the possible range of $f(\bar{x})$, Corollary 3 is applicable similar to Example 2. However, this line of attack is very suboptimal as it does not exploit the fact that X and X' are independent from each other.

Let $Y_1, \ldots, Y_\kappa$ be multiple outputs of the protocol hiding the same function value $f(\bar{x})$, where for $k \in [\kappa]$, X_k and X'_k are two independent random variables and $Y_k = X_k \cdot f(\bar{x}) + X'_k$. We apply the same attack method of observing the average $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} Y_k$ and its concentration around the expected value. By the independence of X_k and X'_k and by

$$\frac{1}{\kappa} \cdot \sum_{k=1}^{\kappa} Y_k = f(\bar{x}) \cdot \frac{1}{\kappa} \cdot \sum_{k=1}^{\kappa} X_k + \frac{1}{\kappa} \cdot \sum_{k=1}^{\kappa} X'_k$$

we are allowed to observe the partial sums using the same method. Doing so, the partial errors will add up.

Compared to the straightforward application of Corollary 3 on Y this has the benefit of allowing to use concentration results specific to the individual distributions that X and X' are drawn from.

Example 3. Let the range of the function value $f(\bar{x})$ be $[0; 1]$ and $Y_1, \ldots, Y_\kappa$ be multiple outputs of the same protocol, where X_k and X'_k are two independent random variables and $Y_k = X_k \cdot f(\bar{x}) + X'_k$. Moreover, let X_k be taken uniformly at random from $[0; 2^4]$ and $X'_k \sim \mathcal{N}(0,1)$.

As in Example 2, the function value $f(\bar{x})$ can be neglected. Observing $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} X_k$ with Corollary 3, party P_1 observes after κ equal to $40 \cdot 2^4$ queries an error of less than t equal to 1 with probability at least $1 - 2 \cdot \mathrm{e}^{-5} > 0.9865$. Due to the different distribution of the X'_k and Corollary 2, we obtain a far better

concentration of $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} X'_k$. Party P_1 observes after κ equal to $40 \cdot 2^4$ queries an error of less than t' equal to 10^{-4} with probability larger than 0.99999. Summing up, party P_1 observes for $\kappa^{-1} \cdot \sum_{k=1}^{\kappa} Y_k$ an error of less than $t + t' = 1.0001$ with probability larger than $(1 - 2 \cdot \mathrm{e}^{-5}) \cdot 0.99999 > 0.9865$.

3.5 General Case

In the general case, the distribution of the random noise X_k may change in k. As long as the random variables are drawn independently, the same attack as employed before will leak information. Assuming every X_k is a bounded random variable in $[a; b] \subset \mathbb{R}$, we still have a sequence of independent random variables, which are by Hoeffding's lemma sub-Gaussian albeit with possibly different parameters. The general form of Hoeffding's inequality [35] (see also Appendix A.3) also gives in this case the same type of concentration result: The probability that the average is in a given confidence interval around the mean grows exponentially fast in κ.

An increasing number n of input-producing parties reduces the individual contribution and, thus, it becomes increasingly difficult for party P_1 to infer information about the input of any party P_j. Hence, generalizing this attack to scenarios with $n > 2$ parties, an adversary that colludes with less than $n - 1$ parties is not able to infer precise knowledge but will learn relations of private inputs, clearly reducing entropy.

4 Data-Dependent Deterministic Obfuscation

We showed in Sect. 3 that the standard blinding functions reveal information exponentially fast in the number of queries on the same value to be blinded. The used concentration bounds are applicable as long as the noise is drawn independently. Therefore, blinding values based on input-specific knowledge are an important measure against the described class of attacks.

We propose to use *data-dependent* blinding values.To mitigate the described class of attacks, we suggest to always use the same, i.e., *deterministic*, blinding values to blind the same confidential data. This ensures that adversaries do not gain additional knowledge through repeated queries. However, different confidential inputs should always be blinded with different blinding values. Furthermore, blinding values should be impossible to guess or compute given only public knowledge, except with negligible probability. The paradigm of data-dependent deterministic obfuscation ($\mathrm{D}^3\mathrm{O}$), given by Definition 1, incorporates these requirements.

4.1 A $\mathrm{D}^3\mathrm{O}$ Instantiation

We focus on obfuscation functions that involve multiplicative and additive blinding, respectively. Without loss of generality, we assume an obfuscation function

$y = x \cdot r_1 + r_2$. The core of our D^{3}O protocol is the construction of deterministic random values r_i with $i \in \{1, 2\}$ that can be used for multiplicatively and additively blinding x. Hence, it works similarly for other obfuscation functions. Following the analysis for multiplicative blinding in [24], we further require the r_i to have random, but fixed, magnitude l_i to hide the length of the confidential value x. Both r_i are constructed in the same way.

We follow the general definitions for cryptographic, i.e., collision-resistant, hash functions and cryptographically secure pseudorandom generators (PRG). Let $H(\cdot)$ be a cryptographic hash function, x be the secret value that is supposed to be obfuscated, and K_i be a uniformly chosen but fixed secret key known only to an obfuscating party R. We first compute a hash h_i as follows.

$$h_i = H(K_i, x) \tag{5}$$

As we will demonstrate in Sect. 4.2, the hash computation can incorporate further details like query attributes. If we model $H(\cdot)$ as a random oracle and require K_i to be known only to R, we can assume h_i to be distributed uniformly at random and hard to guess [22].

Given a cryptographically secure PRG $G(\cdot)$ with outputs that are normally distributed, have (fixed) expected value μ_i, and standard deviation σ_i, we generate the length l_i by

$$l_i = G(\mu_i, \sigma_i; h_i), \tag{6}$$

where h_i acts as the seed. As h_i was generated uniformly at random and is known only to R, l_i is pseudorandom [22].

We assume that the length of h_i is reasonably larger than μ_i and define r_i as the first l_i bits of h_i, denoted by

$$r_i = [h_i]_0^{l_i - 1}. \tag{7}$$

The resulting r_i is a pseudorandom value of normally distributed length consisting of uniformly distributed bits. h_i is computed deterministically because all inputs of $H(\cdot)$ in Eq. (5) are fixed for the secret value x. Hence, for the fixed μ_i and σ_i, Eq. (6) outputs a deterministic l_i.

Consequently, each r_i is a deterministic value with uniformly distributed bits and depends on x and a secret key K_i. The obfuscation of x such that $y = x \cdot r_1 + r_2$ meets Definition 1. Since the lengths of the r_i are normally distributed (but deterministic), y does not leak the length of x.

4.2 mRVP - Modified Ratio Verification Protocol with D^{3}O

Recall the cobalt ratio verification function from Eq. (1) as used in the RVP protocol (see Sect. 2.3). For ratio verification of a transaction θ, RVP computes the (approximate) ratio ρ for any party P_j's amounts $x_{j_{ASM}}$ of artisanally mined (ASM) and $x_{j_{LSM}}$ of industrially mined (LSM) cobalt used to manufacture a particular product. The amounts x_j are stored in a distributed ledger DL and can be encrypted under different keys, denoted by $E(x_j, pk_j)$. The central party

R first re-encrypts them under a common key, homomorphically computes the two encrypted aggregates of ASM and ASM+LSM amounts, obfuscates them homomorphically with two random blinding values, and forwards the resulting ciphertexts to a decryption party D. D decrypts them and returns the resulting blinded aggregates to the consumer C. C divides the two values to obtain a close approximation ρ of the ratio. An additional outer layer of additive blinding is used to prevent D from learning the ratio. The required blinding values are samples from D's plaintext space $\mathcal{M}_D$. This form of output obfuscation hides the aggregates but preserves their ratio.

We now show how our D^{3}O instantiation of Sect. 4.1 can be used to defend RVP against averaging of attacks. Additionally, we modify the additive blinding for the divisor as suggested in Sect. 3.2 to mitigate attacks that remove the additive blinding values that prevent factorization. The result is our modified Ratio Verification Protocol mRVP, given in Algorithm 1.

Algorithm 1: mRVP - Modified Ratio Verification Protocol

Data: $\theta, ..., rk_{j\to D}, ..., pk_D, sk_D$
Result: ρ

C sends to R:
  θ
  $r_{s_1}, r_{s_2} \xleftarrow{U} \mathcal{M}_D$
R reads from DL and re-encrypts:
  $(..., E_D(x_j) = RE(E_j(x_j), rk_{j\to D}), ...)$
R computes:
  $E_D(S_{ASM}) = E_D\left(\sum_{j=1}^{m} x_{j_{ASM}}\right) = \bigoplus_{j=1}^{m} E_D(x_{j_{ASM}})$
  $E_D(S_{Total}) = E_D\left(\sum_{j=1}^{m} x_j\right) = \bigoplus_{j=1}^{m} E_D(x_j)$
  $\forall i \in \{1,2,3\} : h_i = H(K_i, \theta || ... || E_j(x_j) || ...)$
    $l_i = G(\mu_i, \sigma_i; h_i)$
    $r_i = [h_i]_0^{l_i - 1}$
  $E_D(S'_{ASM}) = E_D(S_{ASM} \cdot r_1 + r_2) = (E_D(S_{ASM}) \odot E_D(r_1)) \oplus E_D(r_2)$
  $E_D(S'_{Total}) = E_D(S_{Total} \cdot r_1 + r_3) = (E_D(S_{Total}) \odot E_D(r_1)) \oplus E_D(r_3)$
R computes and sends to D:
  $E_D(S''_{ASM}) = E_D(S'_{ASM} + r_{s_1}) = E_D(S'_{ASM}) \oplus E_D(r_{s_1})$
  $E_D(S''_{Total}) = E_D(S'_{Total} + r_{s_2}) = E_D(S'_{Total}) \oplus E_D(r_{s_2})$
D computes and sends to C:
  $S''_{ASM} = D_D(E_D(S''_{ASM}))$
  $S''_{Total} = D_D(E_D(S''_{Total}))$
C computes:
  $\rho = \frac{S'_{ASM}}{S'_{Total}} = \frac{S''_{ASM} - r_{s_1}}{S''_{Total} - r_{s_2}}$

Let "||" denote concatenation, "$\oplus$" and "$\odot$" denote homomorphic addition and multiplication, and $a \xleftarrow{U} \mathcal{D}$ denote sampling some a from a uniform distri-

bution $\mathcal{D}$. For the sake of readability, we further denote encryption of m with pk_j by $c = E_j(m)$ and decryption of c with sk_j by $m = D_j(c)$.

Upon a consumer's request for ratio verification of a transaction θ, the obfuscating party R proceeds as follows. Each r_i with $i \in \{1, 2, 3\}$ is computed separately. Given fixed, secret keys K_1, K_2, K_3, the party R computes $h_i = H(K_i, \theta||...||E(x_j, pk_j)||...)$ for all encrypted amounts that are incorporated in the aggregates as well as the transaction index θ (see Eq. (5)). Then, given fixed μ_i, σ_i, the obfuscating party computes $l_i = G(\mu_i, \sigma_i; h_i)$ (see Eq. (6)) and truncates the h_i accordingly to obtain r_1, r_2, r_3 (see Eq. (7)). Using different μ_i and σ_i for each r_i ensures that r_1, r_2, r_3 satisfy $0 < r_2, r_3 \ll r_1$. Blinding the aggregates results in two deterministically obfuscated aggregates that are not subject to averaging attacks.

Making use of hardware acceleration for $H(\cdot)$ and given the fact that R only needs to store three secret keys K_1, K_2, K_3 as well as six distribution parameters $\mu_1, \mu_2, \mu_3, \sigma_1, \sigma_2, \sigma_3$, which can be public, our D^{3}O instantiation ensures both computational and storage efficiency. We demonstrate the performance and applicability in the following evaluation.

5 Evaluation of Performance and Applicability

To evaluate practicality and applicability of D^{3}O as our mitigation against averaging attacks in mRVP, we chose cobalt ratio verification as a real-life scenario as in [5]. We require the solution to leak as little information about confidential transactions as possible. We first measure the performance of our D^{3}O instantiation in terms of obfuscation runtime. This allows us to quantify the computational overhead that it adds compared to obfuscation with independent random numbers. We then investigate how it affects the accuracy of the protocol outputs and quantify accuracy in terms of deviation from the actual cobalt ratio.

5.1 mRVP and Baselines

Our implementation of mRVP generates values with uniformly distributed bits and a length which follows a Gaussian distribution by incorporating the product's transaction ID as data-dependent input. Similar to the original construction of RVP, mRVP uses a distributed ledger DL to store public supply-chain data.

The plain RVP protocol as proposed in [5] acts as our first baseline. Additionally, we implemented another and somewhat extreme alternative protocol that relies on local differential privacy (LDP) using Laplace noise to protect the confidential data (see Appendix A.1), instead of homomorphic encryption. In this protocol, the supply-chain data is published via a distributed ledger DL in the form of noisy plaintexts rather than ciphertexts. Upon request by a consumer C, the central party R aggregates the noisy data and returns the aggregates to the consumer C, who then computes ρ. As no data encryption is involved, this protocol does not involve a decryption party D. We refer to this protocol as the LDP protocol. Using LDP as a second baseline allows us to investigate

the overhead caused by the homomorphic encryption and proxy re-encryption operations. Furthermore, it allows us to include an additional perspective: the privacy-utility trade-off, with the utility being ratio computation. LDP seems intuitive for settings where participants can add noise locally before publishing the data. Differential privacy gives quantifiable privacy guarantees and allows to compute the risk of leaking information based on the ϵ parameter. One can reasonably assume the LDP protocol to have better performance than RVP and mRVP due to the absence of expensive homomorphic operations.

5.2 Experimental Setup

The three protocols were implemented in Go and C++. We used a permissionless Multichain as our distributed ledger and the PALISADE [1] implementation of the fully homomorphic BFV scheme [8,16].

We deployed the central party R in an industry-scale cloud instance with 488 GiB of memory and 64 vCPUs and used a moderately sized machine with 16 GiB of memory and 4 vCPUs for the requesting consumer. For RVP and mRVP, we deployed the additional decryption party D in a cloud instance with 16 GiB of memory and 4 vCPUs. All three machines were distributed with a distance of several hundred kilometers for a life-like communication scenario.

For our runtime and accuracy comparison, we take the effect of different parameters into account. The first parameter is the number of inputs n, which we chose to be $n \in \{100, 250, 400, 550, 700, 850, 1000\}$. We assume the runtime to increase for larger n. For the LDP protocol, we set $\epsilon \in \{0.1, 1, 2, 3\}$, as common in the literature, and expect the accuracy of the LDP protocol to increase with growing ϵ. Furthermore, we investigated what effect different distributions of the confidential input data might have. For this, we used inputs that follow (a) **U**niform distributions as a baseline, (b) **G**aussian distributions to model mine outputs dominated by large mines, (c) **G**aussian **M**ixtures to model mine outputs dominated by both very small and very large mines, and (d) **P**ower-**L**aw distributions, with the latter being the closest to the actual distribution of sourced cobalt amounts, as the mine outputs from [34] indicate. Each of these distributions was parameterized to meet a life-like ASM-to-LSM ratio in the pool of inputs. For each combination of the four input distributions, seven input sizes, and four ϵ values (for LDP), we ran each protocol 100 times.

5.3 Runtime Evaluation

We start by investigating the computational overhead of our D^3O instantiation as part of output obfuscation. We consider the time required for obfuscation with data-dependent deterministic values in mRVP and compare it to the effort of obfuscation with randomly sampled blinding values in RVP. Even though we assume the blinding-value computation to take longer in mRVP due to the more complex construction compared to random number sampling, we expect the D^3O procedure to have small effect on the overall output obfuscation, which is dominated by homomorphic multiplication and addition.

Table 1. Total obfuscation runtime for RVP and mRVP

Protocol	Obfuscation runtime in ms	Confidence interval in ms
RVP	25.893	[25.764, 26.021]
mRVP	26.297	[26.139, 26.456]

Table 1 shows the total obfuscation runtime for both protocols in milliseconds. The 95% confidence intervals in the rightmost column show that the overhead of D^3O computation is in the order of microseconds and therefore hardly measurable. Hence, the protocols perform equally well, which meets our expectations.

Next, we investigate the total runtime of all three protocols relatively to the number of confidential inputs. Due to the previous runtime comparison, we expect RVP and mRVP to perform similarly. We assume the LDP protocol to perform and scale better due to the absence of expensive homomorphic operations, especially re-encryptions, which mostly leaves plaintext summations.

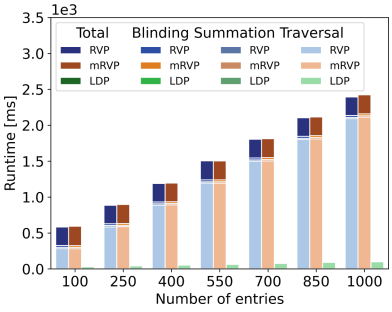

Fig. 1. Total runtimes for RVP, mRVP, and LDP

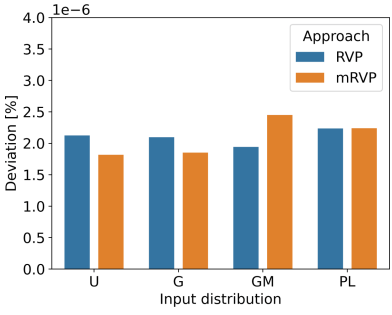

Fig. 2. Accuracy of RVP and mRVP for different input distributions

In fact, Fig. 1 shows that mRVP exhibits a performance that is similar to that of RVP while the LDP protocol performs substantially better. It further reveals the proportion between different parts of the ratio computation. Those parts are the ledger traversal for reading the encrypted or noisy input data, the summation of that data, and the blinding computation, i.e., output obfuscation, bottom to top. Summation and blinding, however, are hardly visible as their impact is negligible. For RVP and mRVP, the total runtime additionally contains communication with the decryption party D. In this scenario, ledger traversal remains the main bottleneck, whereas summation and blinding are computed highly efficiently. The effect of the D^3O computation in this scenario is negligible.

The runtime of summation in RVP and mRVP, including re-encryptions, appears to be constant. We presume that this is caused by the high parallelizability of homomorphic additions and re-encryptions used for aggregation.

5.4 Accuracy Evaluation

We investigate the accuracy of the three protocols by comparing the computed ratio to the actual ratio as reflected in the supply-chain transactions (see Eq. (1)). Given $0 < r_2, r_3 \ll r_1$, we expect the blinding values to almost perfectly cancel out and cause very high accuracy in RVP and mRVP. We expect less accurate results for the LDP protocol due to the effect of the Laplace noise used to perturb the confidential inputs.

Figure 2 shows that, with an average deviation of approximately $2 \cdot 10^{-6}\%$, RVP and mRVP achieve very high accuracy with negligible deviation from the actual cobalt ratio. Furthermore, the different input distributions appear to have negligible effect on the accuracy.

The results for the LDP protocol are depicted in Fig. 3. We observe that the overall deviation is orders of magnitude higher than for the other two protocols, being in the range of several percent. We observe that the input distribution heavily affects the accuracy. For uniformly distributed inputs, we observed a deviation of approximately 6% as well as 10% for normally distributed inputs, 116% for Gaussian mixtures, and 21% for inputs that follow a power-law distribution. Furthermore, Fig. 4 depicts the deviation relatively to the LDP parameter ϵ. It demonstrates that if ϵ increases, so does the accuracy.

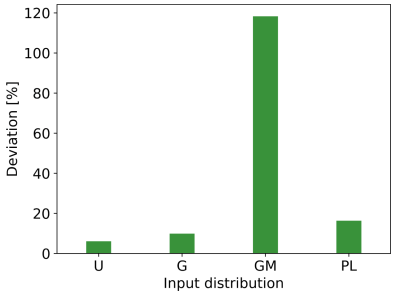

Fig. 3. Accuracy of LDP for different input distributions

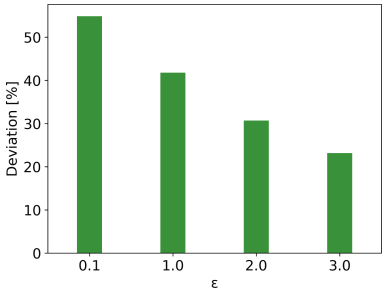

Fig. 4. Accuracy of LDP for different LDP-ϵ values

5.5 Summary

We found that our D^{3}O instantiation adds negligible runtime overhead and does not affect the accuracy of ratio computation, with accuracy negligibly close to 100%. Even though the LDP-based protocol has better performance, its accuracy is substantially worse than that of our mRVP protocol. Hence, in the scenario of cobalt ratio verification, mRVP is preferable. However, in scenarios with loose accuracy requirements where inputs follow a uniform distribution, the LDP-based protocol might be a valid choice.

6 Conclusion and Future Work

Analysis of confidential data is gaining importance, with public verification and traceability of supply-chain data being a prominent example. In this work, we investigated the security of the Ratio Verification Protocol (RVP) proposed in [5]. RVP stands out due to its efficient approach to dividing homomorphically encrypted data. We were able to show that its probabilistic output obfuscation renders RVP vulnerable to averaging attacks. We proved rigorously that such attacks against RVP perform exceptionally well if obfuscation is based on random values sampled independently for every query. We generalized our formal attack analysis to all protocols that employ probabilistic output obfuscation based on additive or multiplicative blinding of aggregates. A clear picture on the performance of such attacks was missing before.

We introduced the paradigm of *data-dependent deterministic obfuscation (*D^3O*)* to prevent averaging attacks and proposed an instantiation of D^3O. We presented mRVP, a D^3O-based modified version of RVP, and demonstrated its practicality and effectiveness in the real-world scenario of cobalt supply-chain verification. We were able to show that D^3O adds negligible runtime overhead. It does not affect the accuracy of outputs and effectively prevents averaging attacks.

Our mRVP protocol allows one single form of queries and thus achieves minimal expressiveness. Hence, it is not subject to typical query-system attacks that try to tailor queries which are semantically equal but not syntactically [17]. However, we note that protocols that use our D^3O instantiation might still be subject to more sophisticated attacks similar to the class of attacks described in [18]. An adversary who requests verifications that involve the same or almost the same aggregates but based on different inputs would learn obfuscated outputs based on fresh randomness for each request. This makes the system vulnerable similar to the analysis in Sect. 3. We emphasize that adding the same noise also for nearly identical confidential data leaks more information than adding different noise. Furthermore, these attacks are scenario-specific. An adversary needs to corrupt almost all producers contributing to the product of interest, or otherwise gain a deep understanding of a major part of the encrypted supply-chain details (to gather a sufficiently large amount of similar query combinations leading to similar outputs). For the considered supply-chain scenario, we deem this class of attacks highly unlikely.

Some of the defenses suggested in [3,18], e.g., query throttling and auditing, can help reduce the risk of sophisticated attacks. However, we note that in general, query auditing is NP-hard [25].

Acknowledgements. We dedicate this work to our late colleague, mentor, and friend Axel Schröpfer, who raised the founding question of this contribution and enriched our work through numerous discussions. Javier Parra-Arnau is the recipient of a "Ramón y Cajal" fellowship (ref. RYC2021-034256-I) funded by the Spanish Ministry of Science and Innovation and the European Union – "NextGenerationEU"/PRTR (Plan de Recuperación, Transformación y Resiliencia). This work also received support from the

Government of Spain under the projects "COMPROMISE" (PID2020-113795RB-C31/AEI/10.13039/501100011033) and "MOBILYTICS" (TED2021-129782B-I00), the latter funded also by the European Union "NextGenerationEU"/PRTR. The authors at KIT are supported by KASTEL Security Research Labs (Topic 46.23 of the Helmholtz Association) and Germany's Excellence Strategy (EXC 2050/1 'CeTI'; ID 390696704).

A Appendix

A.1 Local Differential Privacy

Local differential privacy (LDP) [14] is an adaptation of differential privacy (DP) to a local anonymization scenario, where individuals do not fully trust the data controller and, therefore, anonymize their data *locally*, on their own, before handing it to the controller. The setting is as follows. Consider a set of data providers each wishing to protect private data $X_i \in \mathscr{X}$ on their own. A randomized anonymization mechanism $\mathcal{A}$ is a mechanism that maps X_i randomly to Y_i, where the Y_i's are the anonymized versions of X_i's, the data each individual will send to the controller. For any pair of input values $x, x' \in \mathscr{X}$, and for all $\mathcal{O} \subseteq \mathit{range}(\mathcal{A})$, we say $\mathcal{A}$ satisfies ε-LDP with $\varepsilon > 0$, if

$$\mathbb{P}\left[\mathcal{A}(x) \in \mathcal{O}\right] \leqslant \exp(\varepsilon)\, \mathbb{P}\left[\mathcal{A}(x') \in \mathcal{O}\right]. \tag{8}$$

One of the primary approaches to designing LDP mechanisms is through the addition of Laplace noise [15]. The Laplace mechanism $\mathcal{A}_{\mathcal{L}}$ masks the actual private data x by adding noise L distributed according to a Laplace distribution, and then it returns the randomized response $\mathcal{A}_{\mathcal{L}}(x) = x + L$.

A.2 Averaging Attacks Against Obfuscation Protocols: Formalization of the Special Case

Recall that the outputs of RVP are two values x_1, x_2 that are multiplicatively blinded with r_1 and additively blinded with r_2, i.e.,

$$x_1 \cdot r_1 + r_2$$
$$x_2 \cdot r_1 + r_2$$

They are used for computing the target function

$$\rho = \frac{x_1 \cdot r_1 + r_2}{x_2 \cdot r_1 + r_2}$$

where $0 < r_2 \ll r_1$ ensures

$$\rho \approx \frac{x_1}{x_2}.$$

Given that both the dividend and the divisor are additively blinded with the same r_2, an adversary can subtract one from the other in order for r_2 to cancel out, as follows.

$$\begin{aligned}&(x_1 \cdot r_1 + r_2) - (x_2 \cdot r_1 + r_2)\\ &= x_1 \cdot r_1 + r_2 - x_2 \cdot r_1 - r_2\\ &= x_1 \cdot r_1 - x_2 \cdot r_1\\ &= r_1(x_1 - x_2)\end{aligned}$$

This causes a reduction to multiplicative blinding and renders the quotient subject to factorization in order to obtain

$$\delta = x_1 - x_2.$$

Consequently, $\delta = x_1 - x_2$ and $\rho \approx \frac{x_1}{x_2}$ together yield x_1 and x_2 as follows.

$$\begin{aligned}x_1 &\approx \rho \cdot x_2\\ \delta &\approx \rho \cdot x_2 - x_2\\ \delta &\approx x_2 \cdot (\rho - 1)\\ x_2 &\approx \frac{\delta}{\rho - 1}\end{aligned}$$

Given that r_1 is exponentially larger than r_2, the computed values are close approximations with negligible deviation.

A.3 Averaging Attacks Against Obfuscation Protocols: Omitted Statement

The precise concentration bounds are given below. The proof can be found in [35].

Theorem 3. *Let $X_1, \ldots, X_\kappa$ be a sequence of independent random variables with, for $k \in [\kappa]$, expected value $\mathbb{E}[X_k]$ equal to μ_k. Further, let X be the random variable $X = \kappa^{-1} \cdot \sum_{k=1}^{\kappa}(X_k - \mu_k)$.*

1. *If for every $k \in [\kappa]$ the random variable X_k is sub-Gaussian with parameter σ_k, then*
$$\mathbb{P}[|X| \geqslant t] \leqslant 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa^2}{2 \cdot \sum_{k=1}^{\kappa} \sigma_k^2}\right).$$
2. *If for every $k \in [\kappa]$ the random variable X_k is sub-exponential with parameters (ν_k, b_k), then with*
$$\nu_* = \sqrt{\sum_{k=1}^{\kappa} \frac{\nu_k^2}{\kappa}} \quad \text{and} \quad b_* = \max_{k \in [\kappa]} b_k$$
it holds
$$\mathbb{P}[|X| \geqslant t] \leqslant \begin{cases} 2 \cdot \exp\left(-\frac{t^2 \cdot \kappa}{2 \cdot \nu_*^2}\right), & \text{if } 0 \leqslant t \leqslant \frac{\nu_*^2}{b_*};\\ 2 \cdot \exp\left(-\frac{t \cdot \kappa}{2 \cdot b_*}\right), & \text{for } t \geqslant \frac{\nu_*^2}{b_*}. \end{cases}$$

References

1. PALISADE Lattice Cryptography Library (release 1.11.5), September 2021. https://palisade-crypto.org/
2. Agrawal, T.K.: Contribution to development of a secured traceability system for textile and clothing supply chain. Ph.D. thesis, University of Borås (2019)
3. Asghar, H.J., Kaafar, D.: Averaging attacks on bounded noise-based disclosure control algorithms. Proc. Priv. Enhancing Technol. **2020**(2), 358–378 (2020)
4. Becher, K., Beck, M., Strufe, T.: An enhanced approach to cloud-based privacy-preserving benchmarking. In: Proceedings of NetSys (2019)
5. Becher, K., Lagodzinski, J.A.G., Strufe, T.: Privacy-preserving public verification of ethical cobalt sourcing. In: Proceedings of TrustCom (2020)
6. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054122
7. Boenisch, F., Munz, R., Tiepelt, M., Hanisch, S., Kuhn, C., Francis, P.: Side-channel attacks on query-based data anonymization. In: Proceedings of ACM CCS (2021)
8. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50
9. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory **6**(3), 1–36 (2014)
10. Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, R.: Blockchain-based traceability in Agri-Food supply chain management: a practical implementation. In: Proceedings of IOT Tuscany (2018)
11. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
12. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley Series in Telecommunications and Signal Processing, Wiley, Hoboken (2006)
13. Denning, D.E.: Secure statistical databases with random sample queries. ACM Trans. Database Syst. **5**(3), 291–315 (1980)
14. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: Proceedings of FOCS (2013)
15. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
16. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012). https://eprint.iacr.org/2012/144
17. Francis, P., Probst Eide, S., Munz, R.: Diffix: high-utility database anonymization. In: Schweighofer, E., Leitold, H., Mitrakas, A., Rannenberg, K. (eds.) APF 2017. LNCS, vol. 10518, pp. 141–158. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67280-9_8
18. Gadotti, A., Houssiau, F., Rocher, L., Livshits, B., de Montjoye, Y.-A.: When the signal is in the noise: exploiting Diffix's sticky noise. In: Proceedings of USENIX Security (2019)

19. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
20. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_5
21. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers, 6th edn. Oxford University Press, Oxford (2008)
22. Katz, J., Lindell, Y.: Introduction to Modern Cryptography, 2nd edn. Chapman & Hall/CRC (2014)
23. Kellaris, G., Papadopoulos, S.: Practical differential privacy via grouping and smoothing. Proc. VLDB Endow. **6**(5), 301–312 (2013)
24. Kerschbaum, F.: A privacy-preserving benchmarking platform. Ph.D. thesis, Karlsruhe Institute of Technology (2010)
25. Kleinberg, J., Papadimitriou, C., Raghavan, P.: Auditing Boolean attributes. In: Proceedings of ACM PODS (2000)
26. Kotz, S., Kozubowski, T.J., Podgórski, K.: The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance. Birkhäuser, Boston (2001)
27. Lindell, Y.: Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, 1st edn. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57048-8
28. Malik, S., Kanhere, S., Jurdak, R.: ProductChain: scalable blockchain framework to support provenance in supply chains. In: Proceedings of NCA (2018)
29. Massart, P.: Concentration Inequalities and Model Selection: Ecole d'Eté de Probabilités de Saint-Flour XXXIII - 2003. Lecture Notes in Mathematics, Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-48503-2
30. Mertens, F.: Ueber einige asymptotische gesetze der zahlentheorie. J. für die reine und angewandte Mathematik (1874)
31. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
32. Pibernik, R., Zhang, Y., Kerschbaum, F., Schröpfer, A.: Secure collaborative supply chain planning and inverse optimization - the JELS model. Eur. J. Oper. Res. **208**(1), 75–85 (2011)
33. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
34. van den Brink, S., Kleijn, R., Sprecher, B., Tukker, A.: Identifying supply risks by mapping the cobalt supply chain. Resour. Conserv. Recycl. **156**, 104743 (2020)
35. Wainwright, M.J.: High-Dimensional Statistics: A Non-Asymptotic Viewpoint. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, Cambridge (2019)
36. Westerkamp, M., Victor, F., Küpper, A.: Blockchain-based supply chain traceability: token recipes model manufacturing processes. In: Proceedings of the 2018 IEEE International Conference on Blockchain (2018)