Barriers to Black-Box Constructions of Traitor Tracing Systems

Bo Tang[1] and Jiapeng Zhang[2](✉)

[1] University of Oxford, Oxford, UK
tangbonk1@gmail.com
[2] University of California, San Diego, USA
jpeng.zhang@gmail.com

Abstract. Reducibility between different cryptographic primitives is a fundamental problem in modern cryptography. As one of the primitives, traitor tracing systems help content distributors recover the identities of users that collaborated in the pirate construction by tracing pirate decryption boxes. We present the first negative result on designing efficient traitor tracing systems via black-box constructions from symmetric cryptographic primitives, e.g. one-way functions. More specifically, we show that there is no secure traitor tracing scheme in the random oracle model, such that $\ell_k \cdot \ell_c^2 < \widetilde{\Omega}(n)$, where ℓ_k is the length of user key, ℓ_c is the length of ciphertext and n is the number of users, under the assumption that the scheme does not access the oracle to generate private user keys. To our best knowledge, all the existing cryptographic schemes (not limited to traitor tracing systems) via black-box constructions from one-way functions satisfy this assumption. Thus, our negative results indicate that most of the standard black-box reductions in cryptography cannot help construct a more efficient traitor tracing system.

We prove our results by extending the connection between traitor tracing systems and differentially private database sanitizers to the setting with random oracle access. After that, we prove the lower bound for traitor tracing schemes by constructing a differentially private sanitizer that only queries the random oracle polynomially many times. In order to reduce the query complexity of the sanitizer, we prove a large deviation bound for decision forests, which might be of independent interest.

1 Introduction

Traitor tracing systems, introduced by Chor et al. [11], are *broadcast encryption* schemes that are capable of tracing malicious “traitor” coalitions aiming at building pirate decryption devices. Such schemes are widely applicable to the distribution of digital commercial content (e.g. Pay-TV, news websites subscription, online stock quotes broadcast) for fighting against copyright infringement. In particular, consider a scenario where a distributor would like to send digital contents to n authorized users via a broadcast channel while users possess different secret keys that allow them to decrypt the broadcasts in a non-ambiguous

J. Zhang—Research supported by NSF CAREER award 1350481.

Y. Kalai and L. Reyzin (Eds.): TCC 2017, Part I, LNCS 10677, pp. 3–30, 2017.
https://doi.org/10.1007/978-3-319-70500-2_1

fashion. Clearly, a pirate decoder, built upon a set of leaked secret keys, could also extract the cleartext content illegally. To discourage such piracy in a traitor tracing system, once a pirate decoder is found, the distributor can run a *tracing* algorithm to recover the identity of at least one user that collaborated in the pirate construction.

As a cryptographic primitive, traitor tracing system, together with its various generalizations, has been studied extensively in the literature (e.g., [6,15,24,28, 31]). Considerable efforts have been made on the construction of more efficient traitor tracing scheme from other primitives, in terms of improving two decisive performance parameters – the length of user key and the length of ciphertext. To illustrate, we exhibit in Table 1 the relation between cryptographic assumptions and the performance of the *fully collusion resistant* traitor tracing systems where tracing succeeds no matter how many users keys the pirate has at his disposal.

Table 1. Some previous results on fully collusion resistant traitor tracing systems.

Hardness assumption	User key length	Ciphertext length	Reference
Existence of one-way functions	$\widetilde{O}(n^2)$	$O(1)^{a}$	[7]
Existence of one-way functions	$O(1)$	$\widetilde{O}(n)$	[11,32]
Bilinear group assumptions[b]	$O(1)$	$\widetilde{O}(\sqrt{n})$	[8]
Indistinguishability obfuscation	$O(1)$	$(\log n)^{O(1)}$	[9,16]

[a] All terms in the table including $O(1)$ terms should depend on the security parameters.

[b] Specifically, they need to assume the Decision 3-party Diffie-Hellman Assumption, the Subgroup Decision Assumption and the Bilinear Subgroup Decision Assumption.

Obviously, as we illustrate in Table 1, more efficient traitor tracing schemes can be constructed based on stronger assumptions. Nonetheless, it is natural to ask whether the known constructions can be made more efficient if we only rely on the most fundamental cryptographic assumption – the existence of one-way functions. Impagliazzo and Rudich [21] first studied this type of questions in the context of key agreement. They observed that for most constructions in cryptography, the starting primitive is treated as an oracle, or a “black-box” and the security of the constructed scheme is derived from the security of the primitive in a black-box sense. Based on this observation, they showed that a black-box construction of key agreement built upon one-way functions implies a proof that $P \neq NP$. This approach has been subsequently adopted in investigating the reducibility between other cryptographic primitives, such as one-way permutations [23], public-key encryption [18,19], universal one-way hash functions [25]. In particular, in the context of traitor tracing, the question is whether there exists a more efficient traitor tracing scheme via black-box constructions based on one-way functions. In this paper, we focus on this problem and provide a partial answer to it.

1.1 Our Results

We consider traitor tracing systems in the *random oracle model* [4], which is an ideal model using one way functions in the strongest sense. In this model, the constructed cryptographic scheme can access a random oracle O which can be viewed as a fully random function. In spite of the criticism on its unjustified idealization in practical implementations [10], the random oracle model seems to be an appropriate model and a clean way to establish lower bounds in cryptography (e.g. [1,21]). As there is no security measure defined on the oracle, one common way to prove security for oracle based constructions is to rely on the fully randomness of the oracle and the restriction on the number of queries the adversary (even computationally unbounded) can ask.

Our main result is a lower bound on the performance of traitor tracing systems satisfying a property we call IndKeys. Roughly speaking, a cryptographic scheme is said to be IndKeys if the scheme does not use the black-box hardness of the starting primitive to generate private keys. Here we give an informal definition of the IndKeys property for any cryptographic systems and defer the formal definition tailored for traitor tracing systems to Sect. 2.

Definition 1 (informal). *Let $\Pi^{(\cdot)}$ be a cryptographic scheme that takes other cryptographic primitives or ideal random functions as oracles. We say that $\Pi^{(\cdot)}$ is* IndKeys *if $\Pi^{(\cdot)}$ does not access the oracles while generating private keys.*

Remark 1. Considering all cryptographic primitives (not restriced in the private-key traitor tracing systems we study here), it should be mentioned that the IndKeys property does not require any independence between the public keys and the oracles. Indeed, some of the known black-box constructions of cryptographic primitives use the black-box hardness to generate public keys, (e.g. one time signature [26]), but the private keys are still generated independent of the oracles as requested in IndKeys. To our best knowledge, all the exisiting cryptographic schemes via black-box reductions from one-way functions are IndKeys. Thus, our negative result for IndKeys systems shows that most of the standard black-box reductions in cryptography cannot help to construct a more efficient traitor tracing system. At last, as the IndKeys property is defined on all cryptographic schemes, it might be helpful to investigate the technical limitations of known black-box reductions and derive more lower bounds for other primitives.

In this paper, we show a lower bound on the performance (or efficiency) of the IndKeys traitor tracing systems in terms of the lengths of user keys and ciphertexts. We summarize the main theorem informally as follows and defer the rigorous statement to Sect. 2.

Theorem 1 (informal). *Let $\Pi_{\mathrm{TT}}^{(\cdot)}$ be a secure traitor tracing system that is* IndKeys, *then*

$$\ell_k \cdot {\ell_c}^2 \geq \widetilde{\Omega}(n)$$

where ℓ_k is the length of user key, ℓ_c is the length of ciphertext and n is the number of users.

1.2 Our Approach

We prove our results by building on the connection between traitor tracing systems and *differentially private sanitizers* for counting queries discovered by Dwork et al. [13]. Informally, a database sanitizer is differentially private if its outputs on any two databases that only differ in one row, are almost the same. Dwork, Naor, Reingold, et al. showed that any differentially private and accurate sanitizer (with carefully calibrated parameters) can be used as a valid pirate decoder to break the security of traitor tracing systems. Intuitively, a pirate decoder can be viewed as a sanitizer of databases consist of leaked user keys.

Built upon this connection, we show the lower bound on traitor tracing systems by constructing a sanitizer in the random oracle model. We first build a natural extension of sanitizers and differential privacy in presence of random oracles in Sect. 3. The main difference from standard definitions is that we relax the accuracy requirement by asking sanitizer to be accurate with high probability w.r.t. the random oracle. That is, an accurate sanitizer under our definition might be (probabilistic) inaccurate for some oracle but must be accurate for most oracles. This relaxation allows us to derive a query-efficient sanitizer.

Our sanitizer is developed upon the *median mechanism* designed by Roth and Roughgarden [30], which maintains a set $\mathcal{D}$ of databases and for each counting query: (1) compute the results of the query for all databases in $\mathcal{D}$; (2) Use the median *med* of these results to answer the query if *med* is close to the answer a^* of the true database; (3) If not, output a^* added with a Laplacian noise and remove the databases in $\mathcal{D}$ whose result of the query is not close to a^*. Note that when computing *med*, the median mechanism need to query the oracle for all databases in $\mathcal{D}$ whose size might be exponential in ℓ_k. Thus, it will make exponentially many queries to the oracle.

We design a query-efficient implementation of the median mechanism by using the expectations of query results (taken over all oracles) to compute *med* without querying the real oracle. Our mechanism would be accurate if the answers are concentrated around their expectations taken over all random oracles. Unfortunately, such concentration property does not hold for arbitrary queries and databases. But fortunately, we can show that it holds if there is no "significant" variables in the decryption (or query answering). More specifically, we generalize the deviation bound proved in [2] where they required the size of the database (decision forest) to be relatively larger than the "significance" of the variable (see formal definitions in Sect. 6). Our bound does not make this requirement and is much more applicable. We prove this bound by generalizing two previous deviation bounds proved by Beck et al. [2] and Gavinsky et al. [17]. Note that the IndKeys property is essential in our proof since the deviation bound only holds for uniformly distributed oracles.

To put it together, our mechanism maintains a set of databases $\mathcal{D}$ and for each counting query: (a) remove the variables which are significant for most databases in $\mathcal{D}$; (b) privately check whether the decryption process corresponding to the true database has a significant variable; (c) if there is a significant variable x^*, output the noisy true answer and remove the databases that do not view x^* as a significant variable; (d) otherwise, compute the median *med* among all expected answers of databases in $\mathcal{D}$; (e) if *med* is close to true answer, use it to answer the query; (f) otherwise, output the noisy answer and remove databases in $\mathcal{D}$ whose expected answer is not close to the true answer.

1.3 Related Work

Starting with the seminal paper by Impagliazzo and Rudich [21], black-box reducibility between primitives has attracted a lot of attention in modern cryptography. Reingold et al. [29] revisited existing negative results and gave a more formal treatment of the notions of black-box reductions. In their notions, our results can be viewed as a refutation of the *fully black-box reduction* of IndKeys traitor tracing systems to one-way functions. Our usage of the random oracle model also follows the work by Barak and Mahmoody-Ghidary [1], where they proved lower bounds on the query complexity of every black-box construction of one-time signature schemes from symmetric cryptographic primitives as modeled by random oracles. To our best knowledge, there is no lower bound results on the performance of traitor tracing systems prior to our work.

Differential privacy, as a well studied notion of privacy tailored to private data analysis was first formalized by Dwork et al. [12]. They also gave an efficient sanitizer called *Laplace Mechanism* that is able to answer n^2 counting queries. A remarkable following result of Blum et al. [5] shows that the number of counting queries can be increased to sub-exponential in n by using the *exponential mechanism* of McSherry and Talwar [27]. Subsequently, interactive mechanisms, with running time in polynomial of n and universe size, are developed to answer sub-exponentially many queries adaptively by Roth and Roughgarden [30] (*median mechanism*) and Hardt and Rothblum [20] (*multiplicative weights mechanism*). On the other hand, based on the connection between traitor tracing systems and sanitizers, Ullman [32] proved that no differentially private sanitizer with running time in polynomial of n and the logarithm of the universe size can answer $\widetilde{\Theta}(n^2)$ queries accurately assuming one-way functions exist. Our sanitizer constructions are inspired by the above mechanisms and also rely on the composition theorem of differentially private mechanisms by Dwork et al. [14]. Thus, our results can be viewed as an application of advanced techniques of designing differentially private sanitizer in proving cryptographic lower bounds.

This paper is also technically related to previous deviation bounds on Boolean decision forests. Gavinsky et al. [17] showed that for any decision forest such that every input variable appears in few trees, the average of the decision trees' outputs should concentrate around its expectation when the input variables are distributed independently and uniformly. Similar bounds have also been proved by Beck et al. [2] for low depth decision tress but with a weaker "average" condition (see Sect. 6). As an application, they used this deviation bound to show that AC^0 circuits can not sample good codes uniformly. By a finer treatment on the conditions stated in the above two works, we are able to prove a more general deviation bounds for decision forests, which we believe should have other applications.

1.4 Organization

The rest of the paper is organized as follows. In Sect. 2, we review the formal definition of traitor tracing systems in the random oracle model and state our main theorem. Then we review the connection between traitor tracing systems and differentially private sanitizers in Sect. 3. In Sect. 4, we prove a weaker lower bound which is $\widetilde{\Omega}(n^{1/3})$ to illustrate the main ideas via using a general large deviation bound for decision forests. Then we improve the bound to $\widetilde{\Omega}(n)$ as stated in our main theorem in Sect. 5 by more elaborate arguments. Then, in Sect. 6, we exhibit the proof the large deviation bound for decision forests which is omitted in the proof in Sect. 2. Due to space limit, some proofs are deferred in Appendix A. Furthermore, in Appendix B, we show an oralce separation result between one-way functions and secure traitor tracing systems as a straight-forward implication of our main theorem.

2 Traitor Tracing Systems

In this section, we give a formal definition of traitor tracing systems in the random oracle model and state our main theorem. For any security parameter $\kappa \in \mathbb{N}$, an oracle can be viewed as a Boolean function $O : \{0,1\}^{\ell_o(\kappa)} :\rightarrow \{0,1\}$, where ℓ_o is a function from $\mathbb{N}$ to $\mathbb{N}$.

Definition 2. *Let n, m, ℓ_k, ℓ_c, and ℓ_o be functions $: \mathbb{N} \rightarrow \mathbb{N}$, a traitor tracing system in the random oracle model denoted by $\Pi_{\mathtt{TT}}$ with n users, user-key length ℓ_k, ciphertext length ℓ_c, m tracing rounds and access to an oracle with input length ℓ_o, also contains the following four algorithms. We allow all the algorithms to be randomized except* $\mathtt{Dec}$.

- $\mathtt{Gen}^O(1^\kappa)$, *the setup algorithm, takes a security parameter κ as input and a Boolean function $O : \{0,1\}^{\ell_o(\kappa)} \rightarrow \{0,1\}$ as an oracle, and outputs $n = n(\kappa)$ user-keys $k_1, \ldots, k_n \in \{0,1\}^{l_k(\kappa)}$. Formally, $\mathbf{k} = (k_1, \ldots, k_n) \leftarrow_R \mathtt{Gen}^O(1^\kappa)$.*
- $\mathtt{Enc}^O(\mathbf{k}, b)$, *the encrypt algorithm, takes n user-keys $\mathbf{k}$ and a message $b \in \{0,1\}$ as input, and outputs a ciphertext $c \in \{0,1\}^{l_c(\kappa)}$ via querying an oracle O. Formally, $c \leftarrow_R \mathtt{Enc}^O(\mathbf{k}, b)$.*
- $\mathtt{Dec}^O(k_i, c)$, *the decrypt algorithm takes a user-key k_i and a ciphertext c as input, and outputs a message $b \in \{0,1\}$ via querying an oracle O. Formally, $b = \mathtt{Dec}^O(k_i, c)$.*
- $\mathtt{Trace}^{O,\mathcal{P}^O}(\mathbf{k})$, *the tracing algorithm, takes n user-keys $\mathbf{k}$ as input, an oracle O and a pirate decoder $\mathcal{P}^O$ as oracles, and makes $m(\kappa)$ queries to $\mathcal{P}^O$, and outputs the name of a user $i \in [n]$. Formally, $i \leftarrow_R \mathtt{Trace}^{O,\mathcal{P}^O}(\mathbf{k})$.*

Formally, $\Pi_{\mathtt{TT}} = (n, m, \ell_k, \ell_c, \ell_o, \mathtt{Gen}^{(\cdot)}, \mathtt{Enc}^{(\cdot)}, \mathtt{Dec}^{(\cdot)}, \mathtt{Trace}^{(\cdot,\cdot)})$.

For simplicity, when we use the notation $\Pi_{\mathtt{TT}}$ without any specification, we also mean all these functions and algorithms are defined correspondingly. We also abuse the notations of functions of κ to denote the values of functions when κ is clear from the context, (e.g., n denotes $n(\kappa)$).

Intuitively, the pirate decoder $\mathcal{P}$ can be viewed as a randomized algorithm that holds a set of user-keys $\mathbf{k}_S = (k_i)_{i \in S}$ with $S \subseteq [n]$. The tracing algorithm $\texttt{Trace}$ is attempting to identify a user $i \in S$ by making queries to $\mathcal{P}$ interactively. In particular, in each round $j \in [m]$, $\texttt{Trace}$ submits a ciphertext c_j to $\mathcal{P}$ and then $\mathcal{P}$ answers a message $\widehat{b}_j \in \{0,1\}$ based on all the previous ciphertexts $c_1, \ldots, c_j$. Formally, $\widehat{b}_j \leftarrow_R \mathcal{P}^O(\mathbf{k}_S, c_1, \ldots, c_j)$. Note that we allow the tracing algorithm to be *stateful*. That our lower bounds apply to stateful Traitor Tracing Systems makes our results stronger. Given a function ℓ_o and a security parameter κ, let $\mathcal{O}_{\text{unif}}$ denote the uniform distribution over all oracles with size $\ell_o(\kappa)$, i.e. the uniform distribution for all Boolean functions with input $\{0,1\}^{\ell_o(\kappa)}$. We also abuse $\mathcal{O}_{\text{unif}}$ to denote the support of this distribution. As a pirate decoder, $\mathcal{P}$ should be capable of decrypting ciphertext with high probability as defined formally as follows.

Definition 3. *Let Π_{TT} be a traitor tracing system and $\mathcal{P}^{(\cdot)}$ be a pirate decoder, we say that $\mathcal{P}$ is m-*available *if for every $S \subseteq [n]$ s.t. $|S| \geq n-1$,*

$$\Pr_{\substack{O \sim \mathcal{O}_{\text{unif}}, \mathbf{k} \leftarrow_R \texttt{Gen}^O(1^\kappa) \\ c_j \leftarrow_R \texttt{Trace}^{O,\mathcal{P}}(\mathbf{k}, \widehat{b}_1, \ldots, \widehat{b}_{j-1}) \\ \widehat{b}_j \leftarrow_R \mathcal{P}^O(\mathbf{k}_S, c_1, \ldots, c_j)}} \left[\begin{array}{c} \exists j \in [m], b \in \{0,1\} \\ \big(\forall i \in S, \texttt{Dec}^O(k_i, c_j) = b\big) \wedge \big(\widehat{b}_j \neq b\big) \end{array} \right] \leq neg(n(\kappa))$$

Similarly, a traitor tracing system should decrypt the ciphertext correctly.

Definition 4. *A traitor tracing system Π_{TT} is said to be* correct *if for all oracle O, user $i \in [n]$ and message $b \in \{0,1\}$,*

$$\Pr_{\substack{\mathbf{k} \leftarrow_R \texttt{Gen}^O(1^\kappa) \\ c \leftarrow_R \texttt{Enc}^O(\mathbf{k}, b)}} [\texttt{Dec}^O(k_i, c) = b] = 1$$

In addition, we require the traitor tracing system to be efficient in terms of the number of queries it makes. In particular, we use $\text{QC}(\mathcal{A}^O)$ to denote the query complexity of $\mathcal{A}^O$, i.e. the number of queries $\mathcal{A}^O$ makes to O.

Definition 5. *A traitor tracing system Π_{TT} is said to be* efficient *if for any oracle O with input size $\ell_o(\kappa)$ and for any pirate decoder $\mathcal{P}$, the query complexity of $\texttt{Gen}^O, \texttt{Enc}^O, \texttt{Dec}^O, \texttt{Trace}^O$ are in polynomial of their input size respectively. Formally, $\text{QC}(\texttt{Gen}^O) = \texttt{poly}(\kappa)$, $\text{QC}(\texttt{Enc}^O) = \texttt{poly}(n, \ell_k)$, $\text{QC}(\texttt{Dec}^O) = \texttt{poly}(\ell_k, \ell_c)$ and $\text{QC}(\texttt{Trace}^{O,\mathcal{P}}) = \texttt{poly}(n, m, \ell_k)$.*

Note that we do not make any restriction on the computational power of the traitor tracing systems. Obviously, any computationally efficient Π_{TT} is also query efficient but the other direction does not hold. That our lower bounds apply to efficient Π_{TT} in the above definition makes our results apply to computational efficient Π_{TT} directly. Similarly, we say a pirate decoder $\mathcal{P}$ is *efficient* if $\text{QC}(\mathcal{P}^O) = \texttt{poly}(n, \ell_k, \ell_c)$ in each round of its interaction with $\texttt{Trace}$.

Definition 6. *A traitor tracing system Π_{TT} is said to be* secure *if for any efficient $m(\kappa)$-available pirate decoder $\mathcal{P}$ and $S \subseteq [n(\kappa)]$,*

$$\Pr_{\substack{O \sim \mathcal{O}_{\mathrm{unif}} \\ \mathbf{k} \leftarrow_R \mathtt{Gen}^O}} [\mathtt{Trace}^{O,\mathcal{P}^O(\mathbf{k}_S)}(\mathbf{k}) \notin S] \leq o\left(\frac{1}{n(\kappa)}\right)$$

Definition 7 (IndKeys). *A traitor tracing system Π_{TT} is said to be* IndKeys *if for all a security parameter $\kappa \in \mathbb{N}$ and any two oracles O and O', the distribution of $\mathbf{k}$ generated by $\mathtt{Gen}^O(1^\kappa)$ and $\mathtt{Gen}^{O'}(1^\kappa)$ are the same distribution. Equivalently, conditioned on any particular user-keys $\mathbf{k}$, the oracle O can still be viewed as a random variable drawn from $\mathcal{O}_{\mathrm{unif}}$.*

Remark 2. Note that all known traitor tracing systems via black-box hardness are IndKeys. The scheme designed by with $\ell_k = O(n^2\kappa)$ and $\ell_c = O(\kappa)$ does not require oracles and the one designed by Chor et al. [11] and modified by Ullman [32] with $\ell_k = O(\kappa)$ and $\ell_c = O(n\kappa)$ does not need the oracle to generate private keys.

The following theorem is our main theorem whose proof is deferred to Sects. 4 and 5.

Theorem 2. *In the random oracle model, for any $\theta > 0$, there is no query-efficient, correct and secure traitor tracing system $\Pi_{\mathrm{TT}}^{(\cdot)}$ which is* IndKeys, *such that for any security parameter $\kappa \in \mathbb{N}$,*

$$\ell_k(\kappa) \cdot \ell_c(\kappa)^2 \leq n(\kappa)^{1-\theta}.$$

3 Differentially Private Sanitizers in Random Oracle Model

In this section, we formally define differentially private sanitizers for counting queries in the random oracle model by extending the standard definitions. After that we show its connection with traitor tracing systems by slightly modifying the proofs in [13,32]. For ease of presentation, we reuse the notations used in Sect. 2, (e.g. $n, m, \ell_k, \ell_c, \ell_o$) to denote their counterparts in the context of private data analysis.

A counting query on $\{0,1\}^{\ell_k}$ is defined by a deterministic algorithm $q^{(\cdot)}$ where given any oracle $O : \{0,1\}^{\ell_o} \to \{0,1\}$, q^O is a Boolean function $\{0,1\}^{\ell_k} \to \{0,1\}$. Abusing notation, we define the evaluation of the query $q^{(\cdot)}$ on a database $D = (x_1, \ldots, x_n) \in (\{0,1\}^{\ell_k})^n$ with access to O to be $q^O(D) = \frac{1}{n}\sum_{i\in[n]} q^O(x_i)$. Let $\mathcal{Q}$ be a set of counting queries. A sanitizer $\mathcal{M}^{(\cdot)}$ for $\mathcal{Q}$ can be viewed as a randomized algorithm takes a database $D \in (\{0,1\}^{\ell_k})^n$ and a sequence of counting queries $\mathbf{q}^{(\cdot)} = (q_1^{(\cdot)}, \ldots, q_m^{(\cdot)}) \in \mathcal{Q}^m$ as input and outputs a sequence of answers $(a_1, \ldots, a_m) \in \mathbb{R}^m$ by accessing an oracle O. We consider interactive mechanisms, that means $\mathcal{M}^{(\cdot)}$ should answer each query without knowing subsequent queries. More specifically, the computation of a_i can only depends on the first i queries, i.e. $(q_1^{(\cdot)}, \ldots, q_i^{(\cdot)})$. One might note that our definition differs from the traditional definition of sanitizers by allowing both sanitizers and queries to access oracles. Actually, this kind of sanitizers are defined in such a specific way which makes them useful in proving the hardness for the traitor tracing systems defined in Sect. 2. It is also not clear for us if it has any real application in the context of privately data analysis. Here we use the term "query" in two ways, one referring to the query answered by the santizer and the other one meaning the query sent by algorithms to oracles. Without specification, only when we say "query complexity" or "query efficient", we are referring the oracle queries.

We say that two databases $D, D' \in (\{0,1\}^{\ell_k})^n$ are *adjacent* if they differ only on a single row. We use $\mathbf{q}^{(\cdot)} = (q_1^{(\cdot)}, \ldots, q_m^{(\cdot)})$ to denote a sequence of m queries. Next, we give a natural extension of *differential privacy* to the setting with oracle access.

Definition 8. *A sanitizer $\mathcal{M}^{(\cdot)}$ for a set of counting queries $\mathcal{Q}$ is said to be* (ε, δ)-differentially private *if for any two adjacent databases D and D', oracle O, query sequence $\mathbf{q}^{(\cdot)} \in \mathcal{Q}^m$ and any subset $S \subseteq \mathbb{R}^m$,*

$$\Pr[\mathcal{M}^O(D, \mathbf{q}^O) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}^O(D', \mathbf{q}^O) \in S] + \delta$$

If $\mathcal{M}^{(\cdot)}$ is (ε, δ)-differentially private for some constant $\varepsilon = O(1)$ and $\delta = o(1/n)$, we will drop the parameters ε and δ and just say that $\mathcal{M}^{(\cdot)}$ is differentially private.

Proposition 1 (Lemma 3.7 from [20]). *The following condition implies (ε, δ)-differential privacy. For any two adjacent databases D and D', oracle O and any query sequence $\mathbf{q}^{(\cdot)} \in \mathcal{Q}^m$,*

$$\Pr_{a \leftarrow_R \mathcal{M}^O(D, \mathbf{q}^O)} \left[\left| \log \left(\frac{\Pr[\mathcal{M}^O(D, \mathbf{q}^O) = a]}{\Pr[\mathcal{M}^O(D', \mathbf{q}^O) = a]} \right) \right| > \varepsilon \right] \leq \delta$$

Moreover, a sanitizer should answer any sequence of queries accurately with high probability.

Definition 9. *A sanitizer $\mathcal{M}^{(\cdot)}$ is said to be* (α, β)-accurate *for a set of counting queries $\mathcal{Q}$ if for any database D*

$$\Pr_{O \sim \mathcal{O}_{\text{unif}}} \left[\forall \mathbf{q}^{(\cdot)} \in \mathcal{Q}^m, \left\| \mathcal{M}^O(D, \mathbf{q}^O) - \mathbf{q}^O(D) \right\|_\infty \leq \alpha \right] \geq 1 - \beta$$

If $\mathcal{M}^{(\cdot)}$ is (α, β)-accurate for some constant $\alpha < 1/2$ and $\beta = o(1/n^{10})$, we will drop parameters α and β and just say that $\mathcal{M}^{(\cdot)}$ is accurate.

Finally, we consider the query complexity of sanitizers. Clearly, a sanitizer cannot be query efficient if the evaluation of some counting query $q^{(\cdot)}$ is not query efficient. Let $\mathcal{Q}_{\texttt{Enf}}$ be the set of all *efficient* queries, i.e. for any database $D = (\{0,1\}^{\ell_k})^n$ and any oracle O, any $q^O(D) \in \mathcal{Q}_{\texttt{Enf}}$ can be evaluated in $\texttt{poly}(n, \ell_k, \ell_c)$ number of queries to O. A sanitizer is said to be *efficient* if for any oracle O, database D and any query sequence $\mathbf{q}^{(\cdot)} \in \mathcal{Q}_{\texttt{Enf}}^m$, $\mathcal{M}^O(D, \mathbf{q}^O)$ can be computed in $\texttt{poly}(n, m, \ell_k)$ number of queries to O.

Theorem 3. *Given functions n, m, ℓ_k, ℓ_c and $\ell_o : \mathbb{N} \to \mathbb{N}$, if for any query set $\mathcal{Q} \subseteq \mathcal{Q}_{\mathtt{Enf}}$ with size $|\mathcal{Q}| \leq 2^{\ell_c(\kappa)}$, there exists an* efficient, differentially private *and* accurate *sanitizer for any database $D \in (\{0,1\}^{\ell_k(\kappa)})^{n(\kappa)}$ and any m-query sequence in $\mathcal{Q}^m$, then there exists no* efficient, correct *and* secure *traitor tracing system $\mathit{\Pi}_{\mathtt{TT}} = (n, m, \ell_k, \ell_c, \ell_o, \mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Trace})$.*

Remark 3. The proof idea is similar to [13,32], that is if there exist such a sanitizer and a traitor tracing system, we can slightly modify the sanitizer to be an available pirate decoder for the traitor tracing system. The only technical difference is that the traitor tracing system and the sanitizer defined here have access to a random oracle O. So we need to modify the proof in [32] to accommodate these oracle accesses and the definitions in Sects. 2 and 3.

4 Lower Bounds on Traitor Tracing Systems

In this section, we exhibit the proof of a weaker version of Theorem 2. That is, there is no efficient, correct and secure traitor tracing system such that $\ell_k(\kappa) \cdot \ell_c(\kappa)^2 \leq n(\kappa)^{\frac{1}{3}-\theta}$ for any $\theta > 0$. Assume to the contrary that there exists such a system $\mathit{\Pi}_{\mathtt{TT}}$, let q_π denote the maximum query complexity of $\mathtt{Dec}^O(\mathbf{k}, c)$ over all database $\mathbf{k}$, ciphertext c and oracle O. We will construct an efficient, differentially private and accurate sanitizer $\mathcal{M}$ for any m queries from the query set $\{\mathtt{Dec}^{(\cdot)}(\cdot, c) \,|\, c \in \{0,1\}^{\ell_c}\}$ and any database $D \in (\{0,1\}^{\ell_k})^n$ (inspired by [20,30]). In this section, we abuse the notation $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c)$ to denote the function $\frac{1}{n} \cdot \sum_{i \in [n]} \mathtt{Dec}^{(\cdot)}(k_i, c)$. Before describing the santizer, we first define significant variable for decryption.

Definition 10. *Given a database $\mathbf{k} \in (\{0,1\}^{\ell_k})^n$, a decrypt algorithm $\mathtt{Dec}^{(\cdot)}$ and a ciphertext c, we say a variable $x \in \{0,1\}^{\ell_o}$ is β-significant for $\mathtt{Dec}^{(\cdot)}(k_i, c)$ if*

$$\Pr_{O \sim \mathcal{O}_{\text{unif}}} \left[\mathtt{Dec}^O(k_i, c) \, queries \, x\right] \geq \beta$$

We say x is β-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c)$, if x is β-significant for at least one $k_i \in \mathbf{k}$. We say x is (α, β)-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c)$, if x is β-significant for at least αn entries of $\mathbf{k}$.

Our sanitizer is described as Algorithm 1 by setting the parameters σ, α, β to be

$$\sigma = n^{\theta/3}\sqrt{\frac{\ell_k}{n}}, \qquad \alpha = \frac{1}{\ell_c n^\theta}, \qquad \beta = \frac{1}{54 n^4 q_\pi^3}$$

The intuition behind the calibration of parameters is that we need the condition that α dominates $\sigma \ell_k$ which will be used in the later analysis. Since $\ell_k \cdot {\ell_c}^2 \leq n^{\frac{1}{3}-\theta}$, by simple calculation, we have $\alpha/(\sigma \ell_k) \geq n^{\theta/6}$.

The main idea is to maintain a set of potential databases denoted by $\mathcal{D}_j$ for each round j. Note that the IndKeys property of the system guarantees that conditioned on any particular database, the oracles are always distributed uniformly. This allows us to focus on the available databases not the database and oracle pairs. For each ciphertext c_j, the sanitizer consists of three phases. In phase 1, we examine all $x \in \{0,1\}^{\ell_o}$ and determine a set (denoted by W_j) of significant variables which is queried with probability at least $\beta/2$ over randomness over all $O \in \mathcal{O}_{\text{unif}}$ and $\mathbf{k} \in \mathcal{D}_{j-1}$. Roughly speaking, we pick all variables which are significant for most databases. It should be emphasized that even though some variables are not picked in this phase, they might be significant for some database. Then for each variable in W_j, we query O^* on it and simplify the decrypt algorithm by fixing the value of this specific variable. Note that, this phase does not depend on the true database $\mathbf{k}^*$ so it is clear that there is no privacy loss here. On the other hand, as we will show in Lemma 1, the total number of queries we ask to the oracle O^* in this phase is polynomial in n.

In phase 2, we check if the $\texttt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ has (α, β)-significant variables by using a variant of the exponential mechanism. If there is a significant variable, the santizer outputs $\widehat{a}_j$ the true answer with a noise and modifies $\mathcal{D}_j$. If there are no (α, β)-significant variables, the sanitizer runs phase 3, where it "guesses" the answer by using the median of database set $\mathcal{D}'_{j-1}$ which is the set of all databases in $\mathcal{D}_{j-1}$ which has no (α, β)-significant variables. The sanitizer outputs the guess med_j if it is close to the true answer. Otherwise, the sanitizer outputs $\widehat{a}_j$ and modify $\mathcal{D}_j$.

4.1 Efficiency Analysis

Lemma 1. *The query complexity of Algorithm 1 is $O(n\ell_k q_\pi/\beta)$ which is polynomial in n.*

Proof. Let $\mathbf{x} = (x_1, \ldots, x_{q_\pi})$ be a sequence of q_π oracle variables where $x_i \in \{0,1\}^{\ell_o}$ and $\mathbf{b} = (b_1, \ldots, b_{q_\pi})$ be a sequence of q_π bits where $b_i \in \{0,1\}$. We define an indicator function of $\mathbf{x}, \mathbf{b}, O$ and $\mathbf{k}$ as follows.

$$\mathbf{1}_{\mathbf{x},\mathbf{b}}(O, \mathbf{k}) = \begin{cases} 1 & \text{if } \texttt{Dec}^O(\mathbf{k}, c_j) \text{ queries } x_1, \ldots, x_{q_\pi} \text{ sequentially and } \mathbf{b} = O(\mathbf{x}) \\ 0 & \text{otherwise.} \end{cases}$$

Then we define a potential function $\Phi = \sum_{\mathbf{x},\mathbf{b}} \sum_{O \in \mathcal{O}_{\text{unif}}, k \in \mathcal{D}_{j-1}} \mathbf{1}_{\mathbf{x},\mathbf{b}}(O, \mathbf{k})$. Clearly, the value of Φ at the beginning of Phase 1 is at most $2^{n\ell_k q_\pi}$ since $|\mathcal{D}_{j-1}| \le 2^{n\ell_k}$ and for any particular $\mathbf{k}$ and c_j, the number of all possible query histogram of $\texttt{Dec}^{(\cdot)}(\mathbf{k}, c_j)$ is at most 2^{q_π}.

We will show that when fixing a variable $x \in W_j$ such that

$$\Pr_{\mathbf{k} \sim \text{Unif}(\mathcal{D}_{j-1}), O \sim \mathcal{O}_{\text{unif}}} [\texttt{Dec}^O(k_i, c_j) \text{ queries } x \text{ for some } k_i \in \mathbf{k}] \ge \beta/2$$

the value of Φ will decrease by a factor $(1 - \beta/4)$. This is because fixing the value of x will kill all pair of O and $\mathbf{k}$ such that $\texttt{Dec}^O(\mathbf{k}, c_j)$ queries x but O is not consistent to O^* on x. Since Φ can be less than 1, there are at most $O(n\ell_k q_\pi/\beta)$ elements in W_j. □

Algorithm 1. Sanitizer for Traitor Tracing Lower Bound

```
Input: $n, m$, an oracle $O^* : \{0,1\}^{\ell_o} \to \{0,1\}$, a database $k^* = \{k_1^*, \ldots, k_n^*\}$ with
    $k_i^* \in \{0,1\}^{\ell_k}$, a sequence of queries $\left(\mathtt{Dec}^{(\cdot)}(\cdot, c_1), \ldots, \mathtt{Dec}^{(\cdot)}(\cdot, c_m)\right)$ with
    $c_j \in \{0,1\}^{\ell_c}$
Output: A sequence of answers $ans_1, \ldots, ans_m$ with $a_j \in \mathbb{R}$ or a fail symbol
    FAIL
Initialize $\mathcal{D}_0 \leftarrow$ the set of all databases of size $n$ over $\{0,1\}^{\ell_k}$;
for each query $\mathtt{Dec}^{(\cdot)}(\cdot, c_j)$ where $j = 1, \ldots, m$ do
    Sample a noise $\Delta a_j \sim \mathtt{Lap}(\sigma)$;
    Compute the true answer $a_j \leftarrow \mathtt{Dec}^{O^*}(\mathbf{k}^*, c_j)$ and the noisy answer
      $\widehat{a}_j \leftarrow a_j + \Delta a_j$;
    /* Phase 1: Fix significant variables by querying $O^*$ */
    Initialize the set of significant variables $W_j \leftarrow \emptyset$;
    repeat foreach $x \in \{0,1\}^{\ell_o} \setminus W_j$ do
        if $\Pr_{\mathbf{k} \sim \mathrm{Unif}(\mathcal{D}_{j-1}), O \sim \mathcal{O}_{\mathrm{unif}}}[\mathtt{Dec}^{O}(k_i, c_j)$ queries $x$ for some $k_i \in \mathbf{k}] \geq \beta/2$
          then
            Query $O^*$ on $x$ and fix $x$ to be $O^*(x)$ in $\mathtt{Dec}^{(\cdot)}(\cdot, c_j)$;
            $W_j \leftarrow W_j \cup \{x\}$;
    until $W_j$ is not changed in the last iteration;
    /* Phase 2: Examine whether $k^*$ has $(\alpha, \beta)$-significant variables.
      */
    $\mathcal{U}_j \leftarrow \{x \notin W_j \mid \exists \mathbf{k} \in \mathcal{D}_{j-1}$ s.t. $x$ is $\beta$-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)\}$;
    foreach $x \in \mathcal{U}_j$ do
        $S_j(x) \leftarrow \{k_i^* \mid x$ is $\beta$-significant for $\mathtt{Dec}^{(\cdot)}(k_i^*, c_j)\}$;
        Sample $\Delta I_j(x) \sim \mathtt{Lap}(\sigma)$;
        $I_j(x) \leftarrow |S_j(x)|/n$;
        $\widehat{I}_j(x) \leftarrow I_j(x) + \Delta I_j(x)$;
    $x_j^* \leftarrow \mathrm{argmax}\{\widehat{I}_j(x)\}$;
    if $\widehat{I}_j(x_j^*) \geq \alpha/2$ then
        $u_j \leftarrow 1$; if $\sum_{t=1}^{j} u_t > n\ell_k$ then abort and output FAIL;
        $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1} \setminus \{\mathbf{k} \mid x_j^*$ is not $\beta$-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)\}$;
        Output $ans_j \leftarrow \widehat{a}_j$;
    else /* Phase 3: Check whether the median is a good estimation. */
        $\mathcal{D}'_{j-1} \leftarrow \mathcal{D}_{j-1} \setminus \{\mathbf{k} \mid \exists x \in$
          $\{0,1\}^{\ell_o} \setminus W_j, x$ is $(\alpha, \beta)$-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)\}$;
        $med_j \leftarrow$ the median value of $\mathbb{E}_{O \sim \mathcal{O}_{\mathrm{unif}}}[\mathtt{Dec}^{O}(\mathbf{k}, c_j)]$ among all $\mathbf{k} \in \mathcal{D}'_{j-1}$;
        if $|med_j - \widehat{a}_j| > 0.2$ then
            $u_j \leftarrow 1$; if $\sum_{t=1}^{j} u_t > n\ell_k$ then abort and output FAIL;
            $\mathcal{D}_j \leftarrow \mathcal{D}'_{j-1} \setminus \{\mathbf{k} \mid |\widehat{a}_j - \mathbb{E}_{O \sim \mathcal{O}_{\mathrm{unif}}}[\mathtt{Dec}^{O}(\mathbf{k}, c_j)]| > 0.2\}$;
            Output $ans_j \leftarrow \widehat{a}_j$;
        else $u_j \leftarrow 0$; $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1}$; Output $ans_j \leftarrow med_j$;
```

4.2 Utility Analysis

In this section, we show that the sanitizer is $(1/3, \mathit{neg}(n))$-accurate. We use $\mathbf{c} = (c_1, \ldots, c_m)$ to denote a sequence of m ciphertexts. Let $\mathcal{M}^O(\mathbf{k}, \mathbf{c})$ be the sanitizer described as Algorithm 1 running on database $\mathbf{k}$ and ciphertext sequence $\mathbf{c}$. We first show that with high probability, $\widehat{a}_j$ is close to a_j for all round j.

Lemma 2. *For any $O^* \in \mathcal{O}_{\text{unif}}$, any database $\mathbf{k}^* \in (\{0,1\}^{\ell_k})^n$ and any sequence of m ciphertexts $\mathbf{c} \in (\{0,1\}^{\ell_c})^m$,*

$$\Pr_{\widehat{\mathbf{a}} \leftarrow_R \mathcal{M}^{O*}(\mathbf{k}^*, \mathbf{c})} [\exists j \in [m],\, |\widehat{a}_j - a_j| > 0.1] \le \mathit{neg}(n)$$

Proof. Since Δa_j is drawn from $\mathtt{Lap}(\sigma)$, $\Pr[|\Delta a_j| > 0.1] \le e^{-0.1/\sigma} = \mathit{neg}(n)$. The lemmas follows by using union bound on all $j \in [m]$. □

Then we show that with high probability, the phase 2 can successfully detect the significant variable in $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ for all round j.

Lemma 3. *In the execution of Algorithm 1, for any round j where $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ has a (α, β)-significant variable after Phase 1,*

$$\Pr\left[\widehat{I}_j(x_j^*) < \alpha/2\right] < \mathit{neg}(n)$$

Proof. Let τ be $\max_x\{I_j(x)\}$. Note that $\tau \ge \alpha$ since $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ has a (α, β)-significant variable. So we have

$$\Pr[\tau + \mathtt{Lap}(\sigma) < \alpha/2] < \frac{1}{2} \cdot e^{-\frac{\alpha}{2\sigma}} = \mathit{neg}(n)$$

The lemma follows the fact that $\widehat{I}_j(x_j^*) < \alpha/2$ implies $\tau + \mathtt{Lap}(\sigma) < \alpha/2$. □

Before bounding the failure probability of the sanitizer, we first exhibit a large deviation bound for decision forest whose proof is deferred to Sect. 6.

Proposition 2. *For any $c_j \in \{0,1\}^{\ell_c}$ and $\mathbf{k} \in (\{0,1\}^{\ell_k})^n$, if there is no (α, β)-significant variable in $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)$ then for any $\delta_1 > 0$ and $\delta_2 > 0$,*

$$\Pr_{O^* \sim \mathcal{O}_{\text{unif}}}\left[\left|\mathtt{Dec}^{O^*}(\mathbf{k}, c_j) - \mathop{\mathbb{E}}_{O \sim \mathcal{O}_{\text{unif}}}\left[\mathtt{Dec}^{O^*}(\mathbf{k}, c_j)\right]\right| > \delta_1 + h\delta_2 + n^2 h\sqrt{\beta}\right] \le e^{-2\delta_1^2/\alpha} + h^8 e^{-\delta_2^2/\beta}$$

where h is the query complexity of $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)$.

Lemma 4. *For any database $\mathbf{k}^* \in (\{0,1\}^{\ell_k})^n$, if there is no (α, β)-significant variables in $\mathtt{Dec}^O(\mathbf{k}^*, c)$, then*

$$\Pr_{O^* \sim \mathcal{O}_{\text{unif}}}\left[\exists c \in \{0,1\}^{\ell_c},\, \left|\mathtt{Dec}^{O^*}(\mathbf{k}^*, c) - \mathop{\mathbb{E}}_{O \sim \mathcal{O}_{\text{unif}}}\left[\mathtt{Dec}^{O}(\mathbf{k}^*, c)\right]\right| > 0.1\right] \le \mathit{neg}(n)$$

Proof. Let $T = 0.1$, by Proposition 2 (setting $\delta_1 = T/3$, $\delta_2 = T/(3q_\pi)$, $h = q_\pi$) and noting that $\beta = T/(3n^4 q_\pi^3)$,

$$\Pr_{O^* \sim \mathcal{O}_{\text{unif}}} \left[\left| \mathtt{Dec}^{O^*}(\mathbf{k}^*, c) - \mathop{\mathbb{E}}_{O \sim \mathcal{O}_{\text{unif}}} \left[\mathtt{Dec}^{O}(\mathbf{k}^*, c) \right] \right| > T \right] \leq 2e^{-T^2/(9\alpha)} + 2q_\pi^8 e^{-2Tn^4 q_\pi/3}$$

By taking union bound over all $c \in \{0,1\}^{\ell_c}$, the lemma follows that $\alpha = 1/(\ell_c n^\theta)$. □

Remark 4. Note that the statement of Lemma 4 requires that, with high probability, for all ciphertext $c \in \{0,1\}^{\ell_c}$, $\mathtt{Dec}^{O^*}(k^*, c)$ should concentrate around the expectation. One might wonder whether this requirement is too stringent as the sanitizer only answers m (which may be far less than 2^{ℓ_c}) queries. Unfortunately, it seems that this condition cannot be relaxed because the m queries asked by the adversary might depend on the oracle O^*. So when considering all O^*, the number of possible queries can be much greater than m.

In order to bound the failure probability of the sanitizer, we divide all the query rounds $1, \ldots, m$ into three types.

- Type 1: $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ has a (α, β)-significant variable. So $\widehat{a}_j$ is used to answer the query.
- Type 2: The median med_j is not close to $\widehat{a}_j$. So $\widehat{a}_j$ is used to answer the query.
- Type 3: The mechanism use med_j to answer the query.

We say a round is *bad* if it is in Type 1 or 2 otherwise it is said to be *good*.

Lemma 5. *For any database* $\mathbf{k} \in (\{0,1\}^{\ell_k})^n$*,*

$$\Pr_{O^* \sim \mathcal{O}_{\text{unif}}} \left[\forall \mathbf{c} \in (\{0,1\}^{\ell_c})^m, \textit{the number of bad rounds in } \mathcal{M}^{O^*}(\mathbf{k}, \mathbf{c}) > n\ell_k \right] \leq neg(n)$$

Proof. We first show that, in any bad round j, the size of $\mathcal{D}_j$ will shrink by at least a factor of 2, i.e. $|\mathcal{D}_j| \leq |\mathcal{D}_{j-1}|/2$. Consider any Type 1 round j. Let x_j^* be the significant variable picked at this round. Since $x_j^* \notin W_j$,

$$\sum_{O \in \mathcal{O}_{\text{unif}}, k \in \mathcal{D}_{j-1}} \mathbf{1}_{\mathtt{Dec}^{O}(\mathbf{k}, c_j) \text{ queries } x_j^*} \leq |\mathcal{D}_{j-1}| \cdot |\mathcal{O}_{\text{unif}}| \cdot \beta/2$$

On the other hand, since $\mathcal{D}_j$ is obtained by removing all database $\mathbf{k}$ where x_j^* is not β-significant for $\mathtt{Dec}(\mathbf{k}, c_j)$, we have

$$\sum_{O \in \mathcal{O}_{\text{unif}}, k \in \mathcal{D}_{j-1}} \mathbf{1}_{\mathtt{Dec}^{O}(\mathbf{k}, c_j) \text{ queries } x_j^*} \geq |\mathcal{D}_j| \cdot |\mathcal{O}_{\text{unif}}| \cdot \beta$$

Combine above two inequalities, we have $|\mathcal{D}_j| \leq |\mathcal{D}_{j-1}|/2$. Consider any Type 2 round j. Suppose $|\mathcal{D}_j| > |\mathcal{D}_{j-1}|/2 \geq |\mathcal{D}'_{j-1}|/2$. By the definition of $\mathcal{D}_j$ and med_j, we have $|med_j - \widehat{a}_j| \leq T$ which contradicts the fact that j is a Type 2 round.

Next we show that $\mathbf{k}^* \in \mathcal{D}_m$ with probability $1 - neg(n)$ by induction on j. Clearly, $\mathbf{k}^* \in \mathcal{D}_0$. If j is Type 1, in order to show $\mathbf{k}^* \notin \mathcal{D}_{j-1} \setminus \mathcal{D}_j$, it suffices to show that x_j^* is β-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ with probability $1 - \mathit{neg}(n)$. For any x which is not β-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$n, we have $I_j(x) = 0$. Thus, a

$$\Pr[\widehat{I}_j(x) \geq \alpha/2] \leq \frac{1}{2}e^{-\alpha/2\sigma}$$

On the other hand, $|\mathcal{U}_j|$ is at most $2^{\ell_k}\beta/q_\pi$ since

$$|\mathcal{U}_j| \cdot |\mathcal{O}_{\text{unif}}| \cdot \beta \leq \sum_{O \in \mathcal{O}_{\text{unif}}, k \in \mathcal{D}_{j-1}, x \notin W_j} \mathbf{1}_{\mathtt{Dec}^O(\mathbf{k}, c_j) \text{ queries } x} \leq |\mathcal{D}_{j-1}| \cdot |\mathcal{O}_{\text{unif}}| \cdot q_\pi$$

By taking union bound over all $x \in \mathcal{U}_j$, we have the probability that x_j^* is not β-significant for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$ is at most $|\mathcal{U}_j| \cdot e^{-\alpha/2\sigma} \leq 2^{\ell_k}\beta/q_\pi \cdot e^{-\alpha/2\sigma}$. Since $\alpha/(\sigma\ell_k) \geq n^{\theta/6}$, this probability is negligible.

If j is Type 2, by Lemma 3, $\mathbf{k}^* \in \mathcal{D}'_{j-1}$ with probability at least $1 - \mathit{neg}(n)$. Then by Lemmas 2 and 4, with probability at least $1 - \mathit{neg}(n)$, $|\widehat{a}_j - a_j| \leq 0.1$ and $\left|a_j - \mathbb{E}_{O \sim \mathcal{O}_{\text{unif}}}\left[\mathtt{Dec}^O(\mathbf{k}^*, c_j)\right]\right| \leq 0.1$. Thus, $\mathbf{k}^* \notin \mathcal{D}'_{j-1} \setminus \mathcal{D}_j$ by triangle inequality. If j is Type 3, it is obvious since $\mathcal{D}_{j-1} = \mathcal{D}_j$.

Putting it all together, the lemma follows the facts that $|\mathcal{D}_0| = 2^{n\ell_k}$, $|\mathcal{D}_m| \geq 1$ with probability $1 - \mathit{neg}(n)$ and $|\mathcal{D}_j| \leq |\mathcal{D}_{j-1}|/2$ for all bad rounds. □

Lemma 6 *(Utility).* *Algorithm 1 is $(0.3, \mathit{neg}(n))$-accurate, i.e., for any database $\mathbf{k}^* \in (\{0,1\}^{\ell_k})^n$,*

$$\Pr_{O^* \sim \mathcal{O}_{\text{unif}}}\left[\forall \mathbf{c} \in (\{0,1\}^{\ell_c})^m, \forall j \in [m], |\mathit{ans}_j - a_j| < 0.3\right] \geq 1 - \mathit{neg}(n)$$

where ans_j is the answer output by $\mathcal{M}^{O^*}(\mathbf{k}^*, \mathbf{c})$ at round j and a_j is the true answer $\mathtt{Dec}^{O^*}(\mathbf{k}^*, c_j)$.

Remark 5. Actually, the outermost probability should also be taken over the random coins in $\mathcal{M}$, i.e. the randomness of the Laplace noises. We omit this for the ease of presentation since these random coins are independent from the choice of O^* and $\mathbf{c}$.

Proof. By the description of Algorithm 1, if the sanitizer succeeds, $|\mathit{ans}_j - \widehat{a}_j| \leq 0.2$ for all round j. Thus the lemma follows from Lemmas 2 and 5. □

4.3 Privacy Analysis

Our goal in this section is to demonstrate that, Algorithm 1 is $(\varepsilon, \mathit{neg}(n))$-differentially private. We first simplify the output of our sanitizer as a vector $\mathbf{v}$, which will be shown to determine the output transcript of the sanitizer.

$$v_j = \begin{cases} \widehat{a}_j, x_j^* & \text{if round } j \text{ is Type 1} \\ \widehat{a}_j, \perp & \text{if round } j \text{ is Type 2} \\ \perp, \perp & \text{if round } j \text{ is Type 3} \end{cases}$$

Lemma 7. *Given the oracle O^* and $\mathbf{v}$, the output of Algorithm 1 can be determined.*

Fix an oracle O^* and two adjacent databases $\mathbf{k}, \mathbf{k}' \in (\{0,1\}^{\ell_k})^n$. Let A and B denote the output distributions of our sanitizer when run on the input database $\mathbf{k}$ and $\mathbf{k}'$ respectively. We also use A and B to denote their probability density function dA and dB. The support of both distributions is denoted by $\mathcal{V} = (\{\perp\} \cup \mathbb{R}, \{\perp\} \cup \{0,1\}^{\ell_o})^n$. For any $\mathbf{v} \in \mathcal{V}$, we define the *loss* function $L : \mathcal{V} \to \mathbb{R}$ as

$$L(\mathbf{v}) = \log\left(\frac{A(\mathbf{v})}{B(\mathbf{v})}\right)$$

By Proposition 1, it suffices to show that

$$\Pr_{\mathbf{v} \sim A}[L(\mathbf{v}) > \varepsilon] < neg(n)$$

Given a transcript $\mathbf{v}$, by chain rule,

$$L(\mathbf{v}) = \log\left(\frac{A(\mathbf{v})}{B(\mathbf{v})}\right) = \sum_{j \in [m]} \log\left(\frac{A_j(v_j \mid \mathbf{v}_{<j})}{B_j(v_j \mid \mathbf{v}_{<j})}\right)$$

where $A_j(v_j \mid \mathbf{v}_{<j})$ is the probability density function of the conditional distribution of Algorithm 1 outputting v_j, conditioned on $\mathbf{v}_{<j} = (v_1, \ldots, v_{j-1})$.

Now fix a round $j \in [m]$ and $\mathbf{v}_{<j}$. We define two borderline events on the noise values $\Delta I_j(x)$ and Δa_j. Let $\mathcal{E}_1$ be the event that $\widehat{I}_j(x_j^*) > \alpha/2 - \sigma$ and $\mathcal{E}_2$ be the event that $|\widehat{a}_j - med_j| > T - \sigma$. It should be emphasized that given $\mathbf{v}_{<j}$, both $\mathcal{E}_1$ and $\mathcal{E}_2$ are events only depends on the Laplacian noises $\{\Delta I_j(x)\}_{x \in \mathcal{U}_j}$ and Δa_j. Equivalently, $\mathcal{E}_1$ is the event that $\{\Delta I_j(x)\}_{x \in \mathcal{U}_j}$ is in the set of noises such that $\widehat{I}_j(x_j^*) > \alpha/2 - \sigma$ and $\mathcal{E}_2$ is the event that $\Delta a_j > T - \sigma + med_j - a_j$ or $\Delta a_j < med_j - a_j - T + \sigma$. In the following lemma, we show that conditioned on $\mathcal{E}_1 \vee \mathcal{E}_2$, with probability at least $1/e$, a round j is a bad round.

Lemma 8. $\Pr[j$ *is of* Type 1 $\mid \mathcal{E}_1] \geq 1/e$ *and* $\Pr\left[j\right.$ *is of* Type 2 $\left.\mid \overline{\mathcal{E}}_1, \mathcal{E}_2\right] \geq 1/e$.

Then we show upper bounds on the privacy loss for three cases $\overline{\mathcal{E}}_1 \wedge \overline{\mathcal{E}}_2$, $\overline{\mathcal{E}}_1 \wedge \mathcal{E}_2$ and $\mathcal{E}_1$. By combining all these three cases, we are able to show the following lemma. Due to space limit, we defer all the proofs in Appendix A.

Lemma 9. *Algorithm 1 is* $(\varepsilon, neg(n))$*-differently private.*

5 Improved Lower Bound

In this section, we show how to improve the bound proved in Sect. 4 to $\widetilde{\Omega}(n)$ by modifying the sanitizer and the proof a bit. Suppose $\ell_k \cdot {\ell_c}^2 \leq n^{1-\theta}$. Set parameters σ, α, β to be

$$\sigma = n^{\theta/3}\sqrt{\frac{\ell_k}{n}}, \qquad \alpha = \frac{1}{\ell_c n^{\theta}}, \qquad \beta = \frac{1}{54 n^4 q_\pi^3}$$

Since $\ell_k \cdot {\ell_c}^2 \leq n^{1-\theta}$, by simple calculation, we have $\alpha/\sigma \geq n^{\theta/6}$.

We modify the definition of $\mathcal{U}_j$ in the line 10 of Algorithm 1 as follows.

$$\begin{aligned} \text{Algorithm 1}: &\quad \mathcal{U}_j \leftarrow \{x \notin W_j \mid \exists \mathbf{k} \in \mathcal{D}_{j-1} \text{ s.t. } x \text{ is } \beta\text{-significant for } \mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)\} \\ \text{New Algorithm}: &\quad \mathcal{U}_j \leftarrow \{x \notin W_j \mid x \text{ is } \beta\text{-significant for } \mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)\} \end{aligned}$$

The efficiency of the new sanitizer follows Lemma 1. The only difference in the utility analysis is in the proof of Lemma 5 where we show $k^* \in \mathcal{D}_m$ if j is $\mathtt{Type\ 1}$. In the new algorithm, this is straight forward since $x_j^* \in \mathcal{U}_j$ must be a β-significant variable for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}^*, c_j)$.

In the privacy analysis, the only difference is that the new definition of $\mathcal{U}_j$ does depend on the true database. Given any adjacent databases $\mathbf{k}, \mathbf{k}'$, we fix a round j and $\mathbf{v}_{<j}$. Let $\mathcal{U}$ and $\mathcal{U}'$ denote the set $\mathcal{U}_j$ when the sanitizer running on $\mathbf{k}$ and $\mathbf{k}'$ respectively. We also use x^* and $x^{*\prime}$ to denote the variable $x_j^* = \operatorname{argmax}_x\{\widehat{I}_j(x)\}$ for $\mathbf{k}$ and $\mathbf{k}'$ respectively. Let $\mathcal{H}_j$ be the event that there exists $x \in \mathcal{U} \setminus \mathcal{U}'$ such that $\Delta I_j(x) \geq \alpha/2 - \sigma - 1/n$ or there exists $x \in \mathcal{U}' \setminus \mathcal{U}$ such that $\Delta I_j'(x) \geq \alpha/2 - \sigma - 1/n$.

Lemma 10. $\Pr[\mathcal{H}_j | \mathbf{v}_{<j}] \leq \mathit{neg}(n)$.

Proof. First, note that $|\mathcal{U}| \leq q_\pi/\beta$ since

$$|\mathcal{U}| \cdot |\mathcal{O}_{\text{unif}}| \cdot \beta \leq \sum_{O \in \mathcal{O}_{\text{unif}}, x \notin W_j} \mathbf{1}_{\mathtt{Dec}^O(\mathbf{k}, c_j) \text{ queries } x} \leq |\mathcal{O}_{\text{unif}}| \cdot q_\pi$$

On the other hand, since $\Delta I_j(x)$ is drawn from $\mathtt{Lap}(\sigma)$ and $\alpha/\sigma \geq n^{\theta/6}$,

$$\Pr[\Delta I_j(x) \geq \alpha/2 - \sigma - 1/n] \leq \frac{1}{2} \cdot e^{-(\alpha/2-\sigma)/\sigma} = \mathit{neg}(n)$$

The lemma follows by taking union bound over all $x \in \mathcal{U} \setminus \mathcal{U}'$ and applying similar arguments for $x \in \mathcal{U}' \setminus \mathcal{U}$. □

We define another random variable A_j' such that $d_{tv}(A_j, A_j') \leq \mathit{neg}(n)$ and $\mathcal{H}_j$ never occurs with respect to A_j' (similar ideas has been also used in proving Theorem 3.5 of [14]). Observe that, conditioned on $\overline{\mathcal{H}}_j$, $\mathcal{E}_1$ implies $x^*, x^{*\prime} \in \mathcal{U} \cap \mathcal{U}'$ and $\overline{\mathcal{E}}_1$ implies the round j is not $\mathtt{Type\ 1}$ for both $\mathbf{k}$ and $\mathbf{k}'$. Let $L'(\mathbf{v})$ be the analogues of $L(\mathbf{v})$ by replacing A_j by A_j' for all $j \in [m]$. Clearly $d_{tv}(L, L') \leq m \cdot \mathit{neg}(n) = \mathit{neg}(n)$. Following the proof of Lemma 9, we can show $\Pr[L'(\mathbf{v}) \geq \varepsilon] \leq \mathit{neg}(n)$ for any $\varepsilon = \Omega(1)$. Thus $\Pr[L(\mathbf{v}) \geq \varepsilon] \leq \mathit{neg}(n)$ follows.

6 Large Deviation Bound for Decision Forests

In this section, we show the large deviation bound for $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)$ for any given $\mathbf{k} \in (\{0,1\}^{\ell_k})^n$ and $c_j \in \{0,1\}^{\ell_c}$. Intuitively, a decrypt algorithm $\mathtt{Dec}^{(\cdot)}(k_i, c_j)$ can be viewed as a decision tree and similarly, $\mathtt{Dec}^{(\cdot)}(\mathbf{k}, c_j)$ represents a decision forest (see formal definition below). So throughout this section, we will use the terms like decision trees/forest instead of decrypt algorithms to present our result on large deviation bound for decision forest.

A *decision tree* D is a binary tree whose internal nodes are labeled with Boolean variables while leaves labeled with 0 or 1. Given an input assignment $\mathbf{a} = (a_1, \ldots, a_m) \in \{0,1\}^n$ to the variables $x_1, \ldots, x_m$, the value computed by D on this input $\mathbf{a}$ is denoted by $D(\mathbf{a})$. This value $D(\mathbf{a})$ is the value of the leaf at a path on D determined in the following way. The path starts from the root of D and then moves to the left child if the current internal node is assigned 0 and to right otherwise. A variable x_i is said to be queried by $D(\mathbf{a})$ if the corresponding path passes through a node labeled x_i. Clearly, every x_i can only be queried by $D(\mathbf{a})$ at most once.

A *decision forest* $\mathcal{F}$ is a collection of $|\mathcal{F}|$ decision trees. For any assignment $\mathbf{a}$ of $\mathbf{x}$, $\mathcal{F}(\mathbf{a})$ denotes the $|\mathcal{F}|$-dimensional vector computed by $\mathcal{F}$ on $\mathbf{a}$, whose ith component is the value computed by the ith tree. We use $w(\mathcal{F}(\mathbf{a}))$ to denote the fractional hamming weight of $\mathcal{F}(\mathbf{a})$, i.e.,

$$w(\mathcal{F}(\mathbf{a})) = \frac{\sum_{D_j \in \mathcal{F}} D_j(\mathbf{a})}{|\mathcal{F}|}.$$

In most cases, we assume the assignment $\mathbf{a}$ are drawn from the uniform distribution on $\{0,1\}^m$. We also use the shorthand notations $\Pr_{\mathbf{a}}$ and $\mathbb{E}_{\mathbf{a}}$ to denote the probability and expectation when $\mathbf{a}$ are uniformly distributed when it is clear from the context. We may also abuse the $\Pr_{\mathbf{a}}$ or $\mathbb{E}_{\mathbf{a}}$ inside another $\Pr_{\mathbf{a}}$ or $\mathbb{E}_{\mathbf{a}}$ to denote the probability or expectation corresponding to another random variable when it is not ambiguous, e.g. $\Pr_{\mathbf{a}}\big[w(\mathcal{F}(\mathbf{a})) > \mathbb{E}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))]\big]$.

Definition 11 ((α,β)-significant). *For a decision forest $\mathcal{F}$ and an input $\mathbf{x}$, a Boolean variable x_i is said to be (α,β)-significant if at least α fraction of trees D in $\mathcal{F}$ satisfy $\Pr_{\mathbf{a}}\big[D(\mathbf{a})$ queries $x_i\big] \geq \beta$.*

For comparison, we discuss the difference between the above definition and the notion called "average significance" used in [2]. Recall that the average significance of x_i on $\mathcal{F}$ is defined as

$$\frac{1}{|\mathcal{F}|} \cdot \sum_{D \in \mathcal{F}} \Pr_{\mathbf{a}}\big[D(\mathbf{a}) \text{ queries } x_i\big].$$

Obviously, if x_i is (α,β)-significant, the average significance of x_i is at least $\alpha \cdot \beta$. On the other hand, if x_i is not (α,β)-significant, it can be shown that the average significance of x_i is at most $\alpha + \beta$. To see this, let $\mathcal{F}_1 \subseteq \mathcal{F}$ be the set of trees D such that $\Pr_{\mathbf{a}}[D(\mathbf{a})$ queries $x_i] \geq \beta$.

$$\begin{aligned}
&\frac{1}{|\mathcal{F}|} \cdot \sum_{D \in \mathcal{F}} \Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i] \\
\leq &\frac{1}{|\mathcal{F}|} \left(\sum_{D \in \mathcal{F}_1} \Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i] + \sum_{D \in \mathcal{F} \setminus \mathcal{F}_1} \Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i] \right) \\
\leq &\frac{1}{|\mathcal{F}|} \left(|\mathcal{F}_1| + \sum_{D \in \mathcal{F} \setminus \mathcal{F}_1} \beta \right) \leq \alpha + \beta
\end{aligned}$$

We restate two theorems from [2,17] in our terms.

Theorem 4 (Theorem 1.1. in [17]). *Let $\mathcal{F}$ be a decision forest that has no $(\alpha, 0)$-significant variable and n be $|\mathcal{F}|$. Then for any $\delta > 0$,*

$$\Pr_{\mathbf{a}} \left[\left| w(\mathcal{F}(\mathbf{a})) - \mathop{\mathbb{E}}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] \right| \geq \delta \right] \leq e^{-2\delta^2/\alpha}$$

Theorem 5 ([2]). *Let $\mathcal{F}$ be a decision forest of height at most h that has no (β, β)-significant variable. Then for any $\delta > 0$,*

$$\Pr_{\mathbf{a}} \left[\left| w(\mathcal{F}(\mathbf{a})) - \mathop{\mathbb{E}}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] \right| \geq h\delta \right] \leq h^8 e^{-\delta^2/\beta}$$

We state the main theorem that we will prove in this section.

Theorem 6. *Let $\mathcal{F}$ be a decision forest of height at most h that has no (α, β)-significant variable. Then for any $\delta_1 > 0$ and $\delta_2 > 0$,*

$$\Pr_{\mathbf{a}} \left[\left| w(\mathcal{F}(\mathbf{a})) - \mathop{\mathbb{E}}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] \right| > \delta_1 + h\delta_2 + n^2 h\sqrt{\beta} \right] \leq e^{-2\delta_1^2/\alpha} + h^8 e^{-\delta_2^2/\beta}$$

For the rest of this section, we fix $\mathcal{F}$ to be a decision forest of size n and height h, which has no (α, β)-significant variables. Let S denote the set of all variables x_i such that there exists $D \in \mathcal{F}$, $\Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i] \geq \sqrt{\beta}$. Clearly, $|S| \leq nh/\sqrt{\beta}$. We use $\bar{S}$ to denote the complement set of S and $\mathbf{a}_S$ to denote the partial assignment truncated on S.

Definition 12 (pruning). *Let $\mathcal{F}_{\mathcal{P}}$ be the pruned forest of $\mathcal{F}$ defined as follows. For each variable $x_i \in S$ and $D \in \mathcal{F}$, if $\Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x] \leq \beta$, we deleted x_i from the corresponding tree in $\mathcal{F}_{\mathcal{P}}$ and instead replaced with leaves assigning the value 0.*

We only show one side of the Theorem 6, i.e.

$$\Pr_{\mathbf{a}} \left[w(\mathcal{F}(\mathbf{a})) < \mathop{\mathbb{E}}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] - \delta_1 - h\delta_2 - n^2 h\sqrt{\beta} \right] \leq e^{-2\delta_1^2/\alpha} + h^8 e^{-\delta_2^2/\beta}$$

The proof of the other side is symmetric by changing the definition of pruning to replacing x_i by 1.

The proof sketch of Theorem 6 can be described as follows. Note that for any assignment $\mathbf{a}$, $w(\mathcal{F}(\mathbf{a})) \geq w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))$. On the other hand, $\mathbb{E}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))] \geq \mathbb{E}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] - n\beta \cdot nh/\sqrt{\beta}$ since pruning each variable in $|S|$ decreases the expectation value at most βn and $|S| \leq nh/\sqrt{\beta}$. Hence, to prove Theorem 6, it suffices to prove that $w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))$ is close to $\mathbb{E}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))]$ with high probability, which can be established in two steps. We first show that, in Lemma 11, for any partial assignment $\mathbf{a}_{\bar{S}}$, $w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))$ is close to $\mathbb{E}_{\mathbf{a}_S}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))]$ with high probability (w.r.t. the randomness of $\mathbf{a}_S$). Then in Lemma 12, we prove that with respect to the randomness of $\mathbf{a}_{\bar{S}}$, $\mathbb{E}_{\mathbf{a}_S}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))]$ is close to $\mathbb{E}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))]$ with high probability. Therefore, Theorem 6 follows union bound.

Lemma 11. *For any partial assignment $\mathbf{a}_{\bar{S}}$ and $\delta > 0$,*

$$\Pr_{\mathbf{a}_S}\left[\left|w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}})) - \mathop{\mathbb{E}}_{\mathbf{a}_S}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))]\right| \geq \delta\right] \leq e^{-2\delta_1^2/\alpha}$$

Proof. Given an assignment $\mathbf{a}_{\bar{S}}$, it is not hard to see that the decision forest $\mathcal{F}_{\mathcal{P}}(\mathbf{x}_S, \mathbf{a}_{\bar{S}})$, which only takes $\mathbf{x}_S$ as input, has no $(\alpha, 0)$-significant variable. Otherwise, such variable must be (α, β)-significant in $\mathcal{F}$. Hence the lemma follows Theorem 4. □

Lemma 12. *For any $\delta > 0$,*

$$\Pr_{\mathbf{a}_{\bar{S}}}\left[\left|\mathop{\mathbb{E}}_{\mathbf{a}_S}\left[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))\right] - \mathop{\mathbb{E}}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))]\right| \geq h\delta\right] \leq h^8 e^{-\delta^2/\beta}$$

Before proving Lemma 12, we define an operation on $\mathcal{F}_{\mathcal{P}}$.

Definition 13 (truncating). *Let $\mathcal{F}_{\mathcal{T}}$ be a truncated forest of $\mathcal{F}_{\mathcal{P}}$ with size $2^{|S|} \cdot |\mathcal{F}_{\mathcal{P}}|$. For each tree $D \in \mathcal{F}_{\mathcal{P}}$, there are $2^{|S|}$ trees in $\mathcal{F}_{\mathcal{T}}$ that corresponds to all possible assignments of $\mathbf{x}_S$.*

Proof. We first show that there is no $(\sqrt{\beta}, \sqrt{\beta})$-significant variables in $\mathcal{F}_{\mathcal{T}}$. Note that all the variables in $\mathcal{F}_{\mathcal{T}}$ are in $\bar{S}$. Assume to the contrary that there exists $x_i \in \bar{S}$ that is $(\sqrt{\beta}, \sqrt{\beta})$-significant. Then

$$\sum_{D \in \mathcal{F}_{\mathcal{P}}} \Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i]/n = \sum_{D_{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}} \Pr_{\mathbf{a}_{\bar{S}}}[D_{\mathcal{T}}(\mathbf{a}_{\bar{S}}) \text{ queries } x_i]/(n \cdot 2^{|S|}) \geq \sqrt{\beta} \cdot \sqrt{\beta} = \beta$$

which implies there is a $D \in \mathcal{F}_{\mathcal{P}}$ such that $\Pr_{\mathbf{a}}[D(\mathbf{a}) \text{ queries } x_i] \geq \beta$. This is a contradiction with the definition of $\bar{S}$.

Thus, by Theorem 5,

$$\Pr_{\mathbf{a}_{\bar{S}}}\left[\left|w(\mathcal{F}_{\mathcal{T}}(\mathbf{a}_{\bar{S}})) - \mathop{\mathbb{E}}_{\mathbf{a}_{\bar{S}}}[w(\mathcal{F}_{\mathcal{T}}(\mathbf{a}_{\bar{S}}))]\right| \leq h\delta\right] \leq h^8 e^{-\delta^2/\beta}$$

Therefore, the lemma follows the fact that $w(\mathcal{F}_{\mathcal{T}}(\mathbf{a}_{\bar{S}})) = \mathbb{E}_{\mathbf{a}_S}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}_S, \mathbf{a}_{\bar{S}}))]$. □

Proof (Proof of Theorem 6). Combining Lemmas 11 and 12, with probability at least $1 - e^{-2\delta_1^2/\alpha} - h^8 e^{-\delta_2^2/\beta}$, we have $w(\mathcal{F}_{\mathcal{P}}(\mathbf{a})) \geq \mathbb{E}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))] - \delta_1 - h\delta_2$. Then the theorem follows that $w(\mathcal{F}(\mathbf{a})) \geq w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))$ and $\mathbb{E}_{\mathbf{a}}[w(\mathcal{F}_{\mathcal{P}}(\mathbf{a}))] \geq \mathbb{E}_{\mathbf{a}}[w(\mathcal{F}(\mathbf{a}))] - n^2 h\sqrt{\beta}$. □

Acknowledgement. We would like to thank Shachar Lovett, Jon Ullman and Salil Vadhan for helpful discussion. Jiapeng Zhang would also like to thank his wife, Yingcong Li.

A Missing Proofs

Proof (of Theorem 3). Assume there exist such a traitor tracing system Π_{TT} and a sanitizer $\mathcal{M}$. We define the pirate decoder $\mathcal{P}$ as follows. The database of $\mathcal{M}$ is the set of user keys hold by the pirate decoder. For each ciphertext sent from `Trace` to $\mathcal{P}$, we use $\mathcal{M}$ to answer it and then return 1 is the answer is at least 1/2 and return 0 otherwise.

Clearly, the $\mathcal{P}$ is efficient and available since $\mathcal{M}$ is efficient and accurate. Let $S = \{k_i\}_{i\in[n]}$. Now consider two experiments: in the first one, we run `Trace` on $\mathcal{P}^{(\cdot)}(S,\cdot)$. Since `Trace` is secure, there must exist a user i^* such that

$$\Pr_{\substack{O\sim\mathcal{O}_{\mathrm{unif}}\\ \mathbf{k}\leftarrow_R \mathtt{Gen}^O}}[\mathtt{Trace}^{O,\mathcal{P}^O(S)}(\mathbf{k}) = i^*] \geq \frac{1}{n(\kappa)} - o\left(\frac{1}{n(\kappa)}\right)$$

Let $S' = S\setminus\{i^*\}$. We run the second experiments on `Trace` and $\mathcal{P}^{(\cdot)}([n]\setminus\{i^*\},\cdot)$. Since $\mathcal{M}$ is differentially private for any $O \in \mathcal{O}_{\mathrm{unif}}$, we have

$$\Pr_{\substack{O\sim\mathcal{O}_{\mathrm{unif}}\\ \mathbf{k}\leftarrow_R \mathtt{Gen}^O}}[\mathtt{Trace}^{O,\mathcal{P}^O(S')}(\mathbf{k}) = i^*] \geq \Omega\left(\frac{1}{n(\kappa)}\right)$$

To complete the proof, notice that since $i^* \notin S'$, a secure Π_{TT} can only output i^* with probability $o(1/n(\kappa))$, a contradiction. □

Proof of Lemma 7). By the description of Algorithm 1, the only variable (or information) passed from round $j-1$ to round j is $\mathcal{D}_{j-1}$. So it suffices to show that given $\mathbf{v}$, the adversary can recover $\mathcal{D}_j$ for all $j \in [m]$. We prove this by induction on j. Clearly, it holds when $j = 0$. Given $\mathcal{D}_{j-1}$, $\mathcal{D}_j$ can be construct as follows. Since Phase 1 does not use any information about $\mathbf{k}*$, the adversary first simulate it by querying O^* on significant variables and simplifying $\mathtt{Dec}^O(\cdot,c_j)$. Next, if $\mathbf{v} = (\widehat{a}_j, x_j^*)$, $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1}\setminus\{\mathbf{k}\,|\, x_j^* \text{ is not } \beta\text{-significant for } \mathtt{Dec}^{(\cdot)}(\mathbf{k},c_j)\}$. If $\mathbf{v} = (\widehat{a}_j, \perp)$, $\mathcal{D}_j \leftarrow \mathcal{D}'_{j-1} \setminus \{\mathbf{k}\,|\,|\widehat{a}_j - \mathbb{E}_{O\sim\mathcal{O}_{\mathrm{unif}}}[\mathtt{Dec}^O(\mathbf{k},c_j)]| > 0.2\}$. If $\mathbf{v} = (\perp,\perp)$, $\mathcal{D}_j \leftarrow \mathcal{D}_{j-1}$. Obviously, this $\mathcal{D}_j$ is exactly is the same one used in Algorithm 1. Finally, we need to argue the case where the sanitizer outputs `FAIL`. It is not hard to see, the santizer fails only if $v_j \neq (\perp,\perp)$ for more than $n\ell_k$ number of rounds. So the adversary can recognize the failure of the sanitizer. □

Before proceeding, we first state two obvious probability facts.

Lemma 13. *For any* $\mu \in \mathbb{R}$ *and* $\sigma > 0$, $\Pr[\mathtt{Lap}(\sigma) > \mu \mid \mathtt{Lap}(\sigma) > \mu - \sigma] \geq 1/e$ *and* $\Pr[\mathtt{Lap}(\sigma) < \mu \mid \mathtt{Lap}(\sigma) < \mu + \sigma] \geq 1/e$.

Proof. We only prove the first inequality. The second follows similar arguments. If $\mu \geq \sigma$, the probability is

$$\frac{\frac{1}{2}e^{-\mu/\sigma}}{\frac{1}{2}e^{-(\mu-\sigma)/\sigma}} = 1/e$$

If $\mu \in (0, \sigma)$, the probability is

$$\frac{\frac{1}{2}e^{-\mu/\sigma}}{1 - \frac{1}{2}e^{-(\mu-\sigma)/\sigma}} \geq \frac{\frac{1}{2e}}{\frac{1}{2}} = 1/e$$

If $\mu \leq 0$, the probability is

$$\frac{1 - \frac{1}{2}e^{\mu/\sigma}}{1 - \frac{1}{2}e^{(\mu-\sigma)/\sigma}} \geq \frac{1}{2}$$

□

Lemma 14. *Let* A, B, C, D *be four random events such that* $\Pr[A \wedge B] = 0$. *Then*

$$\Pr[A \vee B \mid C \vee D] \geq \min\{\Pr[A \mid C], \Pr[B \mid D]\}$$

Proof.

$$\begin{aligned}
\Pr[A \vee B \mid C \vee D] &= \Pr[A \mid C \vee D] + \Pr[B \mid C \vee D] \\
&\geq \Pr[A \mid C] \Pr[C \mid C \vee D] + \Pr[B \mid D] \Pr[D \mid C \vee D] \\
&\geq \min\{\Pr[A \mid C], \Pr[B \mid D]\} \cdot (\Pr[C \mid C \vee D] + \Pr[D \mid C \vee D]) \\
&\geq \min\{\Pr[A \mid C], \Pr[B \mid D]\}
\end{aligned}$$

□

Proof (of Lemma 8). Note that j is of Type 1 iff $\widehat{I}_j(x_j^*) \geq \alpha/2$. So by Lemma 13,

$$\begin{aligned}
&\Pr\left[\widehat{I}_j(x_j^*) \geq \alpha/2 \mid \widehat{I}_j(x_j^*) \geq \alpha/2 - \sigma\right] \\
&\quad = \Pr[\mathsf{Lap}(\sigma) \geq \alpha/2 - S_j(x_j^*)/n \mid \mathsf{Lap}(\sigma) \geq \alpha/2 - S_j(x_j^*)/n - \sigma] \geq 1/e
\end{aligned}$$

Similarly, conditioned on $\overline{\mathcal{E}}_1$, j is of Type 2 iff $\widehat{a}_j - med_j > T$ or $\widehat{a}_j - med_j < -T$. By Lemma 13,

$$\begin{aligned}
\Pr\left[\widehat{a}_j - med_j \leq -T \mid \widehat{a}_j - med_j \leq -(T - \sigma)\right] \geq 1/e \\
\Pr\left[\widehat{a}_j - med_j \geq T \mid \widehat{a}_j - med_j \geq T - \sigma\right] \geq 1/e
\end{aligned}$$

Since $T \geq \sigma$, the second part of the lemma follows by combining the above two inequalities. □

Then in the following three lemmas, we show upper bounds on the privacy loss for three cases $\overline{\mathcal{E}}_1 \wedge \overline{\mathcal{E}}_2$, $\overline{\mathcal{E}}_1 \wedge \mathcal{E}_2$ and $\mathcal{E}_1$.

Lemma 15. *For every $v_j \in \mathcal{V}$,*

$$\log\left(\frac{A_j(v_j \mid \overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \mathbf{v}_{<j})}{B_j(v_j \mid \overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \mathbf{v}_{<j})}\right) = 0$$

Proof. Conditioned on $\overline{\mathcal{E}}_1$ and $\overline{\mathcal{E}}_2$, we have $\widehat{I}_j(x_j^*) \leq \alpha/2 - \sigma$ and $|\widehat{a}_j - med_j| \leq T - \sigma$. Then the round j must be of Type 3 for both $\mathbf{k}$ and $\mathbf{k}'$ since $|a_j - a_j'| \leq 1/n$ and $|I_j(x) - I_j'(x)| \leq 1/n$. □

Lemma 16. *For every $v_j \in \mathcal{V}$,*

$$\log\left(\frac{A_j(v_j \mid \overline{\mathcal{E}}_1, \mathcal{E}_2, \mathbf{v}_{<j})}{B_j(v_j \mid \overline{\mathcal{E}}_1, \mathcal{E}_2, \mathbf{v}_{<j})}\right) \leq \frac{1}{\sigma n}$$

Proof. Following similar argument in Lemma 15, the round j cannot be Type 1 for $\mathbf{k}$ and $\mathbf{k}'$. For any $v_j \in (\mathbb{R}, \perp)$, the sanitizer outputs v_j is either with probability 0 for both $\mathbf{k}, \mathbf{k}'$ or with probabilities differing by an $e^{1/\sigma n}$ ratio. Similarly, for $v_j = (\perp, \perp)$, the probabilities by $\mathbf{k}$ and $\mathbf{k}'$ differ by an $e^{1/\sigma n}$ ratio since $|a_j - a_j'| \leq 1/n$. □

Lemma 17. *For every $v_j \in \mathcal{V}$,*

$$\log\left(\frac{A_j(v_j \mid \mathcal{E}_1, \mathbf{v}_{<j})}{B_j(v_j \mid \mathcal{E}_1, \mathbf{v}_{<j})}\right) \leq \frac{3}{\sigma n}$$

Proof. If $v_j \in (\mathbb{R}, \{0,1\}^{\ell_o})$, let $v_j = (a^*, z)$. We couple the random noise $\Delta I_j(x)$ and $\Delta I_j'(x)$ for all $x \in \mathcal{U}_j \setminus \{z\}$. Let h and h' denote $\max_{x \in \mathcal{U}_j \setminus \{z\}}\{\widehat{I}_j(x_j)\}$ and $\max_{x \in \mathcal{U}_j \setminus \{z\}}\{\widehat{I}_j'(x_j)\}$ respectively. Then we have,

$$A_j(v_j \mid \mathcal{E}_1, \mathbf{v}_{<j}) = \Pr[a_j + \Delta a_j = a^* \wedge \Delta I_j(z) \geq \max\{\alpha/2, h\} - I_j(z) \mid \mathcal{E}_1, \mathbf{v}_{<j}]$$
$$B_j(v_j \mid \mathcal{E}_1, \mathbf{v}_{<j}) = \Pr[a_j' + \Delta a_j' = a^* \wedge \Delta I_j'(z) \geq \max\{\alpha/2, h'\} - I_j'(z) \mid \mathcal{E}_1, \mathbf{v}_{<j}]$$

Thus the ratio between the above two probabilities is at most $e^{\frac{3}{\sigma n}}$ since $|a_j - a_j'| \leq 1/n$, $|I_j(z) - I_j'(z)| \leq 1/n$ and $|h - h'| \leq 1/n$.

If $v_j \in (\perp \cup \mathbb{R}, \perp)$, the santizer outputs v_j only if the round j is not of Type 1. Similarly to the above argument, it is not hard to see that the probabilities that the round j is not of Type 1 for $\mathbf{k}$ and $\mathbf{k}'$ differ at a $e^{2/\sigma n}$ ratio. Then the lemma follows the similar arguments in Lemmas 15 and 16. □

Combining all the above three cases, we are able to bound the expected privacy loss for each round j by using the following two propositions.

Proposition 3 (Lemma 3.2 in [14]). *For any two distributions A, B on a common support $\mathcal{V}$, if*

$$\sup_{v \in \mathcal{V}} \left|\log\left(\frac{A(v)}{B(v)}\right)\right| \leq \varepsilon$$

then

$$\mathop{\mathbb{E}}_{v \sim A}\left[\log\left(\frac{A(v)}{B(v)}\right)\right] \leq 2\varepsilon^2$$

Proposition 4 (Convexity of KL Divergence). *Let A, B, A_1, B_1, A_2, B_2 be distributions over a common probability space such that for some $\lambda \in [0, 1]$, $A = \lambda A_1 + (1-\lambda)A_2$ and $B = \lambda B_1 + (1-\lambda)B_2$. Then*

$$\mathop{\mathbb{E}}_{v\sim A}\left[\log\left(\frac{A(v)}{B(v)}\right)\right] \leq \lambda \mathop{\mathbb{E}}_{v\sim A_1}\left[\log\left(\frac{A_1(v)}{B_1(v)}\right)\right] + (1-\lambda)\mathop{\mathbb{E}}_{v\sim A_2}\left[\log\left(\frac{A_2(v)}{B_2(v)}\right)\right]$$

Lemma 18. *For all $j \in [m]$,*

$$\mathbb{E}\left[\log\left(\frac{A_j(v_j \mid \mathbf{v}_{<j})}{B_j(v_j \mid \mathbf{v}_{<j})}\right)\right] \leq \frac{9}{(\sigma n)^2}$$

Proof. Applying Proposition 3 to Lemmas 15, 16 and 17, we have

$$\begin{aligned}
&\mathbb{E}\left[\log\left(\frac{A_j(v_j \mid \overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \mathbf{v}_{<j})}{B_j(v_j \mid \overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \mathbf{v}_{<j})}\right)\right] = 0 \\
\text{and } &\mathbb{E}\left[\log\left(\frac{A_j(v_j \mid \overline{\mathcal{E}}_1, \mathcal{E}_2, \mathbf{v}_{<j})}{B_j(v_j \mid \overline{\mathcal{E}}_1, \mathcal{E}_2, \mathbf{v}_{<j})}\right)\right] \leq \frac{1}{(\sigma n)^2} \\
\text{and } &\mathbb{E}\left[\log\left(\frac{A_j(v_j \mid \mathcal{E}_1, \mathcal{E}_2, \mathbf{v}_{<j})}{B_j(v_j \mid \mathcal{E}_1, \mathcal{E}_2, \mathbf{v}_{<j})}\right)\right] \leq \frac{9}{(\sigma n)^2}
\end{aligned}$$

Then we can express $A_j(v_j \mid \mathbf{v}_{<j})$ as a convex combination in the form

$$\begin{aligned}
&\Pr[\overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2 \mid \mathbf{v}_{<j}]A_j(v_j \mid \overline{\mathcal{E}}_1, \overline{\mathcal{E}}_2, \mathbf{v}_{<j}) + \Pr[\overline{\mathcal{E}}_1, \mathcal{E}_2 \mid \mathbf{v}_{<j}]A_j(v_j \mid \overline{\mathcal{E}}_1, \mathcal{E}_2, \mathbf{v}_{<j}) \\
&\quad + \Pr[\mathcal{E}_1 \mid \mathbf{v}_{<j}]A_j(v_j \mid \mathcal{E}_1, \mathbf{v}_{<j})
\end{aligned}$$

and express $B_j(v_j \mid \mathbf{v}_{<j})$ similarly. By Proposition 4,

$$\mathbb{E}\left[\log\left(\frac{A_j(v_j \mid \mathbf{v}_{<j})}{B_j(v_j \mid \mathbf{v}_{<j})}\right)\right] \leq \frac{9}{(\sigma n)^2} \cdot \Pr[\mathcal{E}_1 \vee \mathcal{E}_2 \mid \mathbf{v}_{<j}]$$

The lemma follows the fact that any probability is at most 1. □

We say a round j is a borderline round if in this round, either $\mathcal{E}_1$ or $\mathcal{E}_2$ occurs. The following lemma gives a bound on the number of borderline round.

Lemma 19. *Let m' be the number of borderline rounds in Algorithm 1.*

$$\Pr[m' > n^{1+\theta/3}\ell_k] \leq neg(n)$$

Proof. By Lemmas 8 and 14,

$$\Pr\left[j \text{ is a borderline round} \mid j \text{ is } \mathsf{Type\ 1} \text{ or } \mathsf{Type\ 2}\right] \geq 1/e$$

Thus, $\mathbb{E}[m'] \leq e \cdot n\ell_k$. Note that the noises added in each round are independent from other rounds. Hence, by Hoeffding's bound, the lemma follows. □

Proposition 5 (Azuma's Inequality). *Let $A_1, \ldots, A_m$ be real-valued random variables such that for every $i \in [m]$,*

1. $\Pr[|A_1| \leq \alpha] = 1$, *and*
2. *for every* $(a_1, \ldots, a_n) \in \mathrm{Supp}(A_1, \ldots, A_m)$,

$$\mathbb{E}[A_i | A_1 = a_1, \ldots, A_{i-1} = a_{i-1}] \leq \beta.$$

Then for any $z > 0$, *we have*

$$\Pr\left[\sum_{i=1}^{m} A_i > m\beta + z\sqrt{m} \cdot \alpha\right] \leq e^{-z^2/2}$$

Proof (of Lemma 9). We apply Proposition 5 the set of m' borderline rounds. Let $J \subset [m]$ be the set of borderline rounds. For each $j \in J$, let

$$X_j = \log\left(\frac{A_j(v_j \mid \mathbf{v}_{<j})}{B_j(v_j \mid \mathbf{v}_{<j})}\right).$$

Note that $\mathbb{E}[X_j | \mathbf{v}_{<j}] \leq 9/(\sigma n)^2$, $|X_j| \leq 3/(\sigma n)$ and $L(\mathbf{v}) = \sum_{j \in J} X_j$. By Proposition 5 (setting $\alpha = 3/(\sigma n)$, $\beta = 9/(\sigma n)^2$ and $z = n^{\theta/7} n$),

$$\Pr[L(\mathbf{v}) > 9m'/(\sigma n)^2 + 3n^{\theta/7}\sqrt{m'}/(\sigma n)] < neg(n)$$

Since $m' \leq n^{1+\theta/6}\ell_k$ with probability $1 - neg(n)$, we have

$$9m'/(\sigma n)^2 + 3n^{\theta/7}\sqrt{m'}/(\sigma n) \leq \frac{9\ell_k n^{1+\theta/3}}{\ell_k n^{1+2\theta/3}} + \frac{3n^{\theta/7+\theta/6}\sqrt{n\ell_k}}{n^{\theta/3}\sqrt{n\ell_k}} = o(1) \qquad \square$$

B Oracle Separation

In this section, we prove that there exists an oracle such that relative to this oracle, there exist one-way functions but no secure traitor tracing systems. Indeed, we show that given an NP-oracle and a random oracle, one can implement the sanitizer designed in Sects. 4 and 5 computationally efficiently (instead of query efficiently as required before). Recall that the sanitizer in Sect. 4 (with modification described in Sect. 5) need to take exponential time to compute the median value. To make it run in polynomial time, we use the NP-oracle to uniformly sample an NP-set by adopting the algorithms in [3,22].

Proposition 6 ([3]). *Let R be an NP-relation. Then there is a uniform generator for R which is implementable in probabilistic polynomial time with an NP-oracle.*

By using the above proposition we can prove the following theorem.

Theorem 7. *Given an NP-oracle and a random oracle, there is a computationally efficient, accurate and differentially private sanitizer.*

Proof. We prove this theorem by implementing the sanitizer designed in Sect. 4 (with modification described in Sect. 5) efficiently. We first show that given an uniform generator for all $\mathcal{D}_j$, one can implement the sanitizer in polynomial time. Then we show how to construct the desired uniform generator by using Proposition 6.

First, we modify the Phase 3 of the sanitizer such that the estimation can be computed in polynomial time. The idea is that, instead of recording all the $\mathcal{D}_j$ and $\mathcal{D}'_j$, we use the uniform generator to sample databases from them uniformly. Indeed, we sample n times from the uniform generator of $\mathcal{D}_j$ and compute $\mathbb{E}_{O\sim\mathcal{O}_{\text{unif}}}[\mathtt{Dec}^O(\mathbf{k}, c_j)]$ where $\mathbf{k}$ is sampled from generator. Note that $\mathbb{E}_{O\sim\mathcal{O}_{\text{unif}}}[\mathtt{Dec}^O(\mathbf{k}, c_j)]$ can be approximately computed efficiently by sampling $O \sim \mathcal{O}_{\text{unif}}$. Let avg_j be the average value of these samples. By Chernoff bound, we have

$$\Pr\left[\left|avg_j - \mathop{\mathbb{E}}_{\mathbf{k}\sim\mathcal{D}_j, O\sim\mathcal{O}_{\text{unif}}}[\mathtt{Dec}^O(\mathbf{k}, c_j)]\right| > 0.01\right] \leq neg(n)$$

Then we replace med_j by avg_j in the sanitizer. To prove the correctness of the sanitizer, it suffices to show that if $|avg_j - \widehat{a}_j| > 0.2$, the size of $\mathcal{D}_j$ is at most $0.9 \cdot |\mathcal{D}_{j-1}|$. Suppose not. We have

$$\left|\mathop{\mathbb{E}}_{\mathbf{k}\sim\mathcal{D}_j, O\sim\mathcal{O}_{\text{unif}}}[\mathtt{Dec}^O(\mathbf{k}, c_j)] - \widehat{a}_j\right| \leq \frac{|\mathcal{D}_{j-1} \setminus \mathcal{D}_j|}{|\mathcal{D}_{j-1}|} \cdot 1 + \frac{|\mathcal{D}_j|}{|\mathcal{D}_{j-1}|} \cdot 0.2 = 0.19$$

that contradicts the triangle inequality. Similar modifications can be made in other phases of the sanitizer to remove the explicit use of $\mathcal{D}_j$.

Finally, we show how to construct such uniform generators for $\mathcal{D}_j$ and $\mathcal{D}'_j$. By Proposition 6, it suffices to define the corresponding NP-relations. We prove this by induction on the round number j. For the base case where $j = 1$, we have $\mathcal{D}_{j-1} = \mathcal{D}_0$ is the uniform distribution over all databases. Clearly, this can be sampled without using the NP-oracle. For the inductive step, we define the NP-relation between the databases and the algorithm running histories (including the first j queries and all the random coins used). A database is in the NP-relation if and only if it is in the set $\mathcal{D}_{j-1}$ that is consistent with algorithm running history. In other words, the databases are the witness of histories in the relation. It is easy to see that this relation can be verified in polynomial time. Therefore we get uniform generators for them by Proposition 6. □

References

1. Barak, B., Mahmoody-Ghidary, M.: Lower bounds on signatures from symmetric primitives. In: FOCS 2007, pp. 680–688. IEEE (2007)
2. Beck, C., Impagliazzo, R., Lovett, S.: Large deviation bounds for decision trees and sampling lower bounds for AC_0-circuits. In: FOCS 2012, pp. 101–110. IEEE (2012)
3. Bellare, M., Goldreich, O., Petrank, E.: Uniform generation of NP-witnesses using an NP-oracle. Inf. Comput. **163**(2), 510–526 (2000)

4. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: CCS 1993. ACM Request Permissions, December 1993
5. Blum, A., Ligett, K., Roth, A.: A learning theory approach to non-interactive database privacy. In: STOC 2008, New York, USA, p. 609. ACM Request Permissions, New York, May 2008
6. Boneh, D., Franklin, M.: An efficient public key traitor tracing scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 338–353. Springer, Heidelberg (1999). doi:10.1007/3-540-48405-1_22
7. Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: CCS 2008, pp. 501–510. ACM (2008)
8. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006). doi:10.1007/11761679_34
9. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 480–499. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_27
10. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: STOC 1998, pp. 209–218. ACM (1998)
11. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 257–270. Springer, Heidelberg (1994). doi:10.1007/3-540-48658-5_25
12. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). doi:10.1007/11681878_14
13. Dwork, C., Naor, M., Reingold, O., Rothblum, G.N., Vadhan, S.: On the complexity of differentially private data release. In: STOC 2009, pp. 381–390. ACM Press, New York (2009)
14. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: FOCS 2010, pp. 51–60. IEEE (2010)
15. Fiat, A., Tassa, T.: Dynamic traitor tracing. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 354–371. Springer, Heidelberg (1999). doi:10.1007/3-540-48405-1_23
16. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS 2013, pp. 40–49. IEEE (2013)
17. Gavinsky, D., Lovett, S., Saks, M., Srinivasan, S.: A tail bound for read-k families of functions. Random Struct. Algorithms **47**(1), 99–108 (2015)
18. Gennaro, R., Gertner, Y., Katz, J., Trevisan, L.: Bounds on the efficiency of generic cryptographic constructions. SIAM J. Comput. **35**(1), 217–246 (2005)
19. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: FOCS 2000, pp. 325–335. IEEE (2000)
20. Hardt, M., Rothblum, G.N.: A multiplicative weights mechanism for privacy-preserving data analysis. In: FOCS 2010, pp. 61–70. IEEE (2010)
21. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: STOC 1989, pp. 44–61. ACM Request Permissions, New York, February 1989
22. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. **43**, 169–188 (1986)

23. Kahn, J., Saks, M., Smyth, C.: A dual version of Reimer's inequality and a proof of Rudich's conjecture. In: CCC 2000, pp. 98–103. IEEE (2000)
24. Kiayias, A., Yung, M.: On crafty pirates and foxy tracers. In: Sander, T. (ed.) DRM 2001. LNCS, vol. 2320, pp. 22–39. Springer, Heidelberg (2002). doi:10.1007/3-540-47870-1_3
25. Kim, J.H., Simon, D.R., Tetali, P.: Limits on the efficiency of one-way permutation-based hash functions. In: FOCS 1999, p. 535. IEEE Computer Society, Washington, D.C. (1999)
26. Lamport, L.: Constructing digital signatures from a one-way function. Technical report, October 1979
27. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS 2007, pp. 94–103. IEEE (2007)
28. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_3
29. Reingold, O., Trevisan, L., Vadhan, S.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 1–20. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24638-1_1
30. Roth, A., Roughgarden, T.: Interactive privacy via the median mechanism. In: STOC 2010, pp. 765–774. ACM Request Permissions, New York, June 2010
31. Safavi-Naini, R., Wang, Y.: Sequential traitor tracing. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 316–332. Springer, Heidelberg (2000). doi:10.1007/3-540-44598-6_20
32. Ullman, J.: Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In: STOC 2013. ACM Request Permissions, June 2013