Differentially Private Algorithms for Statistical Verification of Cyber-Physical Systems

YU WANG †, HUSSINE SIBAI ‡, MARK YEN †, SAYAN MITRA ‡, AND GEIR E. DULLERUD ‡ (Fellow, IEEE)

†Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA
‡Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA
§Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61820 USA

ABSTRACT

Statistical model checking is a class of sequential algorithms that can verify specifications of interest on an ensemble of cyber-physical systems (e.g., whether 99% of cars from a batch meet a requirement on their functionality). These algorithms infer the probability that given specifications are satisfied by the systems with provable statistical guarantees by drawing sufficient numbers of independent and identically distributed samples. During the process of statistical model checking, the values of the samples (e.g., a user’s car trajectory) may be inferred by intruders, causing privacy concerns in consumer-level applications (e.g., automobiles and medical devices). This paper addresses the privacy of statistical model checking algorithms from the point of view of differential privacy. These algorithms are sequential, drawing samples until a condition on their values is met. We show that revealing the number of samples drawn can violate privacy. We also show that the standard exponential mechanism that randomizes the output of an algorithm to achieve differential privacy fails to do so in the context of sequential algorithms. Instead, we relax the conservative requirement in differential privacy that the sensitivity of the output of the algorithm should be bounded to any perturbation for any data set. We propose a new notion of differential privacy which we call expected differential privacy (EDP). Then, we propose a novel expected sensitivity analysis for the sequential algorithm and propose a corresponding exponential mechanism that randomizes the termination time to achieve the EDP. We apply the proposed exponential mechanism to statistical model checking algorithms to preserve the privacy of the samples they draw. The utility of the proposed algorithm is demonstrated in a case study.

INDEX TERMS

Cyber-physical systems, formal verification, privacy, statistical model checking, stochastic systems.

I. INTRODUCTION

Cyber-Physical systems appear naturally when physical processes are controlled by computer-based algorithms [1] in applications such as automobiles [2], smart grids [3], and medical/health devices [4]. These systems are typically designed in a compositional fashion by interconnecting numerous cyber and physical components. In operation, they are subject to variability from different sources, e.g., control algorithm updates and physical wear and tear. In these contexts, besides pursuing correct-by-construction, verification plays an important role in assuring the systems’ functionality in critical applications [1].

Statistical model checking is a commonly-used approach to verify general specifications for (an ensemble of) cyber-physical systems. These specifications are formally expressed using temporal logic, which is composed of a simple set of syntactic rules. These syntactic rules augment the standard propositional logic with a few temporal operators.

For any temporal logic specification of interest, the statistical model checker can automatically parse it by the semantic rules of the logic and divide it into several sub-specifications, which are verifiable by basic statistical inference techniques. Although the verification results are subject to statistical errors, these are usually tolerable for most
applications [5]. Compared to other model-based verification methods (e.g., [6], [7], [8], [9]), statistical model checking is more scalable and can handle black-box systems. They have been successfully used to verify various specifications of real-world cyber-physical systems from automobiles [2], [10], [11] and system biology [12].

Statistical model checking can be applied to verify specifications for a group of cyber-physical systems. For example, with the advances in connected vehicles, there is a growing interest in monitoring their average performance in various driving scenarios using real-time vehicle data [13], [14], [15]. In such problems, the difference between agents (e.g., vehicles) can be captured by probabilistic models and the various specifications of interest (e.g., fuel efficiency in various driving scenarios) can be captured by temporal logic. Statistical model checking can check whether these specifications hold on the group of systems by sampling.

This work is motivated by the privacy issue in applying statistical model checking for groups of cyber-physical systems, especially with human-on-the-loop. For example, in verifying connected vehicles, the sampled performance may include personal data (e.g., daily itinerary) which is not intended to be revealed to the public [16]. When statistical model checking algorithms infer specifications for a group of cyber-physical systems, the values of the samples they draw may be compromised by intruders by observing their outputs and termination times, as is the case of other statistical inference algorithms [17]. Such an issue causes privacy concerns of statistical model checking in consumer-level applications (e.g., automobiles and health/medical devices) where the samples are related to sensitive personal information.

To prevent such privacy violations, the concept of differential privacy has been proposed and adopted by various industries, e.g., Google [18], Apple [19], and the U.S. Census Bureau [20]. Differential privacy requires the (random) observations of an algorithm not to change significantly in the probabilistic sense when small changes are made to its sample values. If such a bound on the change exists, it will subsequently provide privacy guarantees on how hard it is to infer the values of the samples from the observations.

Achieving standard differential privacy is challenging (if not impossible) for statistical model checking. Existing literature focuses on algorithms that use a fixed number of data entries in their inference [21]. For these algorithms, protecting the data privacy when publicly releasing their outputs can be done by analyzing their outputs’ sensitivity to changes in their input data and accordingly randomizing the latter in what is called the exponential mechanism [22]. However, statistical model checking algorithms are sequential. They draw a varying number of samples depending on a given termination condition. And releasing their verification results and the number of samples they draw may endanger privacy. We will show in Section III-B that the sensitivity of the number of samples, i.e., the maximal difference caused by changing one sample, can be arbitrarily large. It implies that the standard exponential mechanism, which applies to algorithms with finite sensitivity to changes in the input samples for any possible values of these samples, fails to achieve standard differential privacy. Although there have been previous works on differential privacy for sequential algorithms [23], [24], [25], they usually assume that the number of samples used is not observable.

This work shows that achieving a new notion of expected differential privacy (EDP), which is weaker than the standard one, is possible for statistical model checking. This is based on the observation that the data used by statistical model checking algorithms are independently and identically distributed (i.i.d.) samples from a probabilistic model instead of taking arbitrary values. Such a condition has not been considered in previous work. Consequently, for the sequence of samples used for statistical model checking, their empirical distribution converges to the underline distribution of the probabilistic model. This new definition takes into account the distribution of the data sampled by the statistical model checking algorithm instead of the standard worst-case assumption of arbitrary valued databases in standard differential privacy. The idea of utilizing the distribution of input data in differential privacy has been considered in [26]. However, that work only utilizes the distribution of part of the data in a fixed-size database.

The difference between the proposed EDP and the standard differential privacy is as follows. Consider a sequential algorithm $\mathcal{A}$ that takes an infinite sequences of samples $\sigma = (\sigma_1, \sigma_{\ldots})$ as input and generates an output $o = \mathcal{A}(\sigma)$, where the latter is publicly released. Here, $\sigma_i$ denotes an arbitrary $i$th sample and $\sigma_{\ldots}$ denotes the rest of the samples. If $\mathcal{A}$ achieves the standard differential privacy, then changing the value of $\sigma_i$ should only change the output $o$ slightly for any possible values of the samples in $\sigma_{\ldots}$. Instead, if $\mathcal{A}$ achieves EDP, then changing the value of $\sigma_i$ should only slightly change the average of the output $o$ over the distribution of the samples in $\sigma_{\ldots}$ (i.e., the average case).

This paper constructively shows that achieving EDP for statistical model checking algorithms is feasible. Statistical inference in statistical model checking is performed by sequential probability ratio tests [27]. We propose a new exponential mechanism that randomizes the ratio tests using a novel type of sensitivity analysis which we call expected sensitivity analysis. Then, we develop a new statistical model checking algorithm for general signal temporal logic specifications with EDP guarantees. We demonstrate the scalability and applicability of the modified algorithm in a case study on the Toyota powertrain system.

The rest of the paper is organized as follows. Section II provides some preliminaries on statistical model checking. Section III explains why achieving standard differential privacy for sequential algorithms is difficult. Section IV introduces the definition of EDP. Section V develops an expectedly differentially private statistical model checking algorithm. Section VI provides a case study on the Toyota powertrain system. Finally, Section VII concludes this work.
I. NOTATIONS
We denote the set of natural, real numbers, and non-negative real numbers by \( \mathbb{N} \), \( \mathbb{R} \) and \( \mathbb{R}_{\geq 0} \), respectively. For \( n \in \mathbb{N} \), let \([n] = \{1, \ldots, n\}\). The binomial distribution has a probability mass function in the following form
\[
f(K; N, p) = \frac{N!}{K!(N-K)!} p^K (1-p)^{N-K},
\]
and we denote it by Binom\((N, p)\), where \( K, N \in \mathbb{N} \) and \( p \) is a real value in the interval \([0,1]\). The exponential distribution has a probability density function of the form
\[
f(x) = \begin{cases} e^{-x}, & x \geq 0 \\ 0, & x < 0 \end{cases}
\]
and we denote it by Exp\((\epsilon)\), where \( \epsilon \in \mathbb{R}_{\geq 0} \) and \( x \in \mathbb{R} \).

II. PRELIMINARIES ON STATISTICAL MODEL CHECKING
Real-world cyber-physical systems are typically subject to uncertainty in their parameters from different sources, e.g., thermal fluctuations in sensors/actuators. The probabilistic uncertainty is expressible by random variables drawn from probabilistic distributions, which may be unknown. For example, the uncertainty in the sensor readings can be expressed by the addition of a random noise. Such uncertainties can be captured by a probabilistic model. A key problem for the probabilistic model is whether a specification holds on a random system path with probability above some given threshold. Such a problem can be solved with provable probabilistic guarantees by statistical model checking. Below, we summarize previous work on statistical model checking algorithms based on sequentially probability ratio tests [5], [28].

Consider a random signal (e.g., paths/trajectories) \( \sigma \) from a probabilistic model \( S \). The model \( S \) can be either known or unknown and of general form (e.g., discrete, continuous, or hybrid). The goal of statistical model checking is to check the satisfaction probability of a specification \( \varphi \). To start with, we check whether the probability of \( \sigma \) satisfying \( \varphi \) is greater than some given probability threshold \( p \in [0, 1] \), i.e.,
\[
P_{\sigma \models \varphi} = p_\varphi > p,
\]
where \( \models \) means “to satisfy” and \( p_\varphi \) is the satisfaction probability of \( \varphi \) on \( S \).

Formally, the specifications are typically expressed by signal temporal logic (STL) [29], which contains a simple set of syntactic rules to describe the change of real-valued system variables in continuous time. The syntax of STL specifications is defined recursively by:
\[
\varphi := f(\sigma) \geq 0 \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \cup_{[t_1, t_2]} \psi,
\]
where \( f: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n \) is a signal (i.e., a vector of real-valued system variables that changes over time), \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) is a real-valued function of the signal, and \([t_1, t_2] \) is a time interval with \( t_2 > t_1 \geq 0 \). Strictly speaking, \( t_1 \) and \( t_2 \) should be rational-valued or infinity. By recursively applying these syntactic rules, we can write arbitrarily complex STL specifications to express general specifications of interest.

Given any STL specification \( \varphi \), we can define whether a signal \( \sigma \) satisfies it or not, written as \( \sigma \models \varphi \) or \( \sigma \not\models \varphi \), using the following semantic rules:
\[
\begin{align*}
\sigma &\models f(\sigma) \geq 0 \iff f(\sigma(0)) \geq 0 \\
\sigma &\models \neg \varphi \iff \sigma \not\models \varphi \\
\sigma &\models \varphi \land \psi \iff \sigma \models \varphi \land \sigma \models \psi \\
\sigma &\models \varphi \cup_{[t_1, t_2]} \psi \iff \exists \epsilon \in [t_1, t_2] \text{ such that } \sigma^{(\epsilon)} \models \psi
\end{align*}
\]
where \( \sigma^{(\epsilon)} \) denotes the \( \epsilon \)-shift of \( \sigma \), defined by \( \sigma^{(\epsilon)}(t') = \sigma(t + \epsilon) \) for any \( t' \in \mathbb{R}_{\geq 0} \). The first rule means that \( \sigma \) satisfies \( f(\sigma) \geq 0 \) if and only if the initial value of \( \sigma \) at time \( t = 0 \) satisfies \( f(\sigma(0)) \geq 0 \). The fourth rule defined the “until” temporal operator; it means that \( \sigma \) satisfies \( \varphi \cup_{[t_1, t_2]} \psi \) if and only if \( \sigma \) satisfies \( \varphi \) at all time instants before \( t \) until it satisfies \( \psi \) exactly at time \( t \) for some \( t \in [t_1, t_2] \).

Using the STL syntax and semantics, we can define other temporal operators such as “finally” (or “eventually”) and “always,” written as \( \Diamond \) and \( [\square] \). Specifically, \( \Diamond_{[t_1, t_2]} \varphi \) means the property \( \varphi \) finally holds; and \( [\square_{[t_1, t_2]} \varphi \) means the property \( \varphi \) always holds. In addition, although only the inequality symbol “\( > \)” is included in the STL syntax, we can express other inequalities symbols via the semantics. For example, \( f(\sigma) > 0 \) can be expressed by \( \neg (-f(\sigma) \geq 0) \). Below, we will also use these operators or symbols in STL specifications.

Example 1: Suppose \( \sigma = (\sigma_1, \sigma_2) \) where \( \sigma_1 \) and \( \sigma_2 \) are the velocity and battery level of an electric vehicle. Then the specification
\[
(\sigma_1 > 5) \cup_{[0,10]} (\sigma_2 < 0.2)
\]
means the car velocity should be greater than 5 before the battery level drops below 0.2 within 10 time units.

II. SEQUENTIAL PROBABILITY RATIO TEST
Statistical model checking handles the statement (1) by formulating it as a hypothesis testing problem
\[
H_{\text{null}} : p_\varphi > p, \quad H_{\text{alt}} : p_\varphi \leq p,
\]
where \( H_{\text{null}} \) and \( H_{\text{alt}} \) are the null and alternative hypotheses. Then, it infers whether \( H_{\text{null}} \) or \( H_{\text{alt}} \) holds by drawing sample signals \( \sigma_1, \sigma_2, \ldots \) from the probabilistic model \( S \). As with previous work [5], [28], we focus on samples that are drawn independently.

Remark 1: It has been shown that correlated samples can improve the efficiency of statistical model checking [30], [31]. However, using correlated samples requires at least partial knowledge of the system dynamics to implement. On the other hand, independent sampling can apply to general black-box systems.

The statistical model checking algorithm examines the correctness of \( \varphi \) on each sample signals \( \sigma_t \). This process can be performed automatically by existing model checking algorithms from [29], [32]. With a slight abuse of notation, we represent “True” and “False” by 1 and 0 and define the truth...
values by
\[ \varphi(\sigma_i) = \begin{cases} 1, & \sigma_i \models \varphi, \\ 0, & \text{otherwise.} \end{cases} \] (3)

Each \( \varphi(\sigma_i) \) is a binary random variable that is equal to 1 with probability \( p_\varphi \). Since the sample signals are independent, the sum
\[ K = \sum_{i \in [N]} \varphi(\sigma_i) \] (4)
is a random variable following the binomial distribution \( \text{Binom}(N, p_\varphi) \). Thus, the satisfaction probability \( p_\varphi \) can be approximated by the sample average \( K/N \).

Our goal is to find a statistical assertion \( \mathcal{A}(\sigma_1, \ldots, \sigma_N) \rightarrow \{ H_{\text{null}}, H_{\text{alt}} \} \) that claims either \( H_{\text{null}} \) or \( H_{\text{alt}} \) holds based on the observed samples. Moreover, since \( K \) is the sufficient statistic, the statistical assertion can be written as
\[ \mathcal{A}(K, N) \rightarrow \{ H_{\text{null}}, H_{\text{alt}} \}. \]

Due to the randomness of \( \sigma_1, \ldots, \sigma_N \), the value of the statistical assertion \( \mathcal{A} \) does not always agree with the truth value of \( \mathbb{P}_{\sigma \sim S}(\sigma \models \varphi) < p \). To capture these probabilistic errors, we define the false positive/false negative (FP/FN) ratios as
\[ \alpha_{\text{FP}} = \mathbb{P}_{\sigma_1, \ldots, \sigma_N \sim S}(\mathcal{A} = H_{\text{null}} \mid \mathcal{A} = H_{\text{alt}} \models \varphi), \] (5)
\[ \alpha_{\text{FN}} = \mathbb{P}_{\sigma_1, \ldots, \sigma_N \sim S}(\mathcal{A} = H_{\text{alt}} \mid \mathcal{A} = H_{\text{null}} \models \varphi). \] (6)
The FP ratio is the error probability of mistakenly rejecting the null hypotheses \( H_{\text{null}} \) while (1) holds. The FN ratio is the error probability of mistakenly accepting the null hypotheses \( H_{\text{null}} \) while (1) does not hold.

Intuitively, the assertion \( \mathcal{A} \) becomes more accurate in terms of decreasing \( \alpha_{\text{FP}} \) and \( \alpha_{\text{FN}} \) when the number of samples \( N \) increases. For any \( N \), quantitative bounds of \( \alpha_{\text{FP}} \) and \( \alpha_{\text{FN}} \) can be derived by using either the confidence interval method [33] or the sequential probability ratio test method [34]. The former only assumes that \( p_\varphi \neq p \). The latter requires the following stronger assumption but is more efficient. This work focuses on the latter method since the indifference parameter assumption holds in most applications (e.g., in [35]).

**Assumption 1:** There exists a known indifference parameter \( \delta > 0 \) such that \(|p_\varphi - p| > \delta \) in (1).

With Assumption 1, it suffices to consider the two extreme cases in hypothesis (2) according to [34], i.e.,
\[ H_{\text{null}} : p_\varphi = p + \delta, \quad H_{\text{alt}} : p_\varphi = p - \delta. \] (7)
To distinguish between \( H_{\text{null}} \) or \( H_{\text{alt}} \), we consider the likelihood ratio
\[ \lambda(K, N) = \frac{(p + \delta)^K (1 - p - \delta)^{N-K}}{(p - \delta)^K (1 - p + \delta)^{N-K}}. \] (8)

Our goal is to ensure that the FP/FN ratios \( \alpha_{\text{FP}}, \alpha_{\text{FN}} \leq \alpha \) for a given threshold \( \alpha > 0 \), which is called the desired significance level.\(^1\) By sequential probability ratio tests [36], it suffices to keep on drawing samples and stop to make a statistical assertion when either of the conditions on the right-hand-side of the following equation is satisfied:
\[ \mathcal{A}(K, N) = \begin{cases} H_{\text{null}}, & \text{if } \lambda(K, N) \geq \frac{1-\alpha}{\alpha}, \\ H_{\text{alt}}, & \text{if } \lambda(K, N) \leq \frac{\alpha}{1-\alpha}. \end{cases} \] (9)
As \( N \rightarrow \infty \), by the binomial distribution, we have \( \lambda(K, N) \rightarrow 0 \) if \( H_{\text{null}} \) holds or \( \lambda(K, N) \rightarrow \infty \) if \( H_{\text{alt}} \) holds, so the above procedure stops with probability 1. This procedure can be implemented incrementally by Algorithm 1.

**Remark 2:** By Assumption 1, we do not distinguish between checking \( p_\varphi > p \) or checking \( p_\varphi \geq p \) in (1). In addition, by the semantics of STL in Section II, checking \( p_\varphi < p \) or \( p_\varphi \leq p \) is equivalent to checking \( p_{\varphi'} < 1 - p \) or \( p_{\varphi'} \leq 1 - p \). Thus, the method for checking \( p_\varphi > p \) applies to the other three cases.

### III. DIFFERENTIAL PRIVACY IN STATISTICAL MODEL CHECKING

Statistical model checking of the probabilistic model \( S \) of an ensemble of cyber-physical systems, e.g., autonomous cars, service robots, and wearable devices, requires analyzing their sequential decision-making behavior.

![Image](https://example.com/image.png)

**Algorithm 1:** Statistical Model Checking of \( \mathcal{A} \)

**Require:** Probabilistic model \( S \), desired significance level \( \alpha \), and indifference parameter \( \delta \).

1. \( N \leftarrow 0, K \leftarrow 0, \lambda \leftarrow 1 \).
2. **while** True **do**
   3. Draw a new sample signal \( \sigma \) from \( S \).
   4. \( K \leftarrow K + \varphi(\sigma) \), \( N \leftarrow N + 1 \).
   5. \( \lambda \leftarrow \frac{(p + \delta)^K (1 - p - \delta)^{N-K}}{(p - \delta)^K (1 - p + \delta)^{N-K}} \).
   6. **if** \( \lambda \geq \frac{1-\alpha}{\alpha} \) **then** Return \( H_{\text{null}} \)
   7. **else** if \( \lambda \leq \frac{\alpha}{1-\alpha} \) **then** Return \( H_{\text{alt}} \)
   8. **else** Continue
   9. **end if**
10. **end while**

---

\(^1\) We choose the same threshold for \( \alpha_{\text{FP}} \) and \( \alpha_{\text{FN}} \) for simplicity. The method also applies to the case where two different thresholds for \( \alpha_{\text{FP}} \) and \( \alpha_{\text{FN}} \) are required (see [34]).
a slight abuse of notation, for an input sequence of sampled signals $\sigma_{1:}\infty = (\sigma_1, \sigma_2, \ldots)$, we write the termination step and output as functions of $\sigma_{1:}\infty$ by $\tau_{\mathcal{A}}(\sigma_{1:}\infty)$ and $o_{\mathcal{A}}(\sigma_{1:}\infty)$, respectively.

Differential privacy aims to prevent malicious inference on $\sigma_{1:}\infty$. Roughly, Algorithm $\mathcal{A}$ is differentially private if an attacker cannot infer the value of $\sigma_i$ even if they can observe $(\tau_{\mathcal{A}}(\sigma_{1:}\infty), o_{\mathcal{A}}(\sigma_{1:}\infty))$ and the values of the rest of the entries $\sigma_{-i}$ of $\sigma_{1:}\infty$, for any sequence $\sigma_{1:}\infty$ and entry $\sigma_i$ in $\sigma_{1:}\infty$. [39]. Clearly, for a deterministic Algorithm $\mathcal{A}$, the value of $\sigma_i$ can be inferred when the inverse of $\tau_{\mathcal{A}}(\cdot)$ and $o_{\mathcal{A}}(\cdot)$ is unique after knowing the value of $\sigma_{-i}$.

A common approach to achieve differential privacy is to randomize Algorithm $\mathcal{A}$, such that even for the same input sequence $\sigma_{1:}\infty$, the algorithm is randomly executed in slightly different ways and yields different observations. For a randomized algorithm $\mathcal{B}$ of $\mathcal{A}$, although the randomization prevents the intruder to take the inverse of $\tau_{\mathcal{A}}(\cdot)$ and $o_{\mathcal{A}}(\cdot)$, the intruder can still infer the value of $\sigma_i$ by observing the difference in $\tau_{\mathcal{B}}(\sigma_{1:}\infty)$ and $o_{\mathcal{B}}(\sigma_{1:}\infty)$ when the value of $\sigma_{-i}$ is fixed and only the value of $\sigma_i$ changes. To measure the distance between two sequences, we recall the definition of Hamming distance.

**Definition 1:** The Hamming distance between two input sequences $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$ is $D_H(\sigma_{1:}\infty, \sigma'_{1:}\infty) = n$ if and only if they are different in only $n \in \mathbb{N}$ entries. Namely, there exists indexes $1 \leq i_1 < \ldots < i_n \in \mathbb{N}$ such that

$$\sigma_j = \sigma'_j, \quad \text{if } j \in \{i_1, \ldots, i_n\}$$

$$\sigma_j \neq \sigma'_j, \quad \text{otherwise.}$$

We call two input sequences $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$ adjacent if $D_H(\sigma_{1:}\infty, \sigma'_{1:}\infty) = 1$.

Differential privacy requires a bound on the worst case change in the probabilities of the random observations for two adjacent input sequences. If the change in distribution is smaller, then it is harder for the intruder to infer the change in the input. We apply the standard notion of differential privacy in [39] to Algorithm 1 of Section II in the following definition.

**Definition 2:** A randomized sequential algorithm $\mathcal{B}$ is $\varepsilon$-differentially private, if it holds that $$P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma_{1:}\infty), o_{\mathcal{A}}(\sigma_{1:}\infty)) \in \mathcal{O}) \leq e^\varepsilon P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma'_{1:}\infty), o_{\mathcal{A}}(\sigma'_{1:}\infty)) \in \mathcal{O}) \quad (10)$$ for any two adjacent input sequences $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$ and any $\mathcal{O} \subseteq \mathbb{N} \times \{H_{\text{full}}, H_{\text{null}}\}$.

The notation $P_{\mathcal{B}}$ indicates that the probability is taken from the randomized algorithm $\mathcal{B}$. Similar to the standard definition of differential privacy, Definition 2 has the following statistical properties. First, if Algorithm $\mathcal{B}$ is differentially private, we can prove the more general statement: for any two non-adjacent input sequences $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$, $$P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma_{1:}\infty), o_{\mathcal{A}}(\sigma_{1:}\infty)) \in \mathcal{O}) \leq e^{D_H(\sigma_{1:}\infty, \sigma'_{1:}\infty)} P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma'_{1:}\infty), o_{\mathcal{A}}(\sigma'_{1:}\infty)) \in \mathcal{O}). \quad (11)$$ In addition, differential privacy bounds the difference in probabilities via their probability ratios, which are commonly used statistics for probabilistic inference. This bound implies that any two input sequences give exactly the same set of observations with non-zero probabilities. Formally, the condition (11) implies a necessary condition for differential privacy:

$$P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma_{1:}\infty), o_{\mathcal{A}}(\sigma_{1:}\infty)) \in \mathcal{O}) \neq 0$$

$$\iff P_{\mathcal{B}}(\tau_{\mathcal{A}}(\sigma'_{1:}\infty), o_{\mathcal{A}}(\sigma'_{1:}\infty)) \in \mathcal{O}) \neq 0. \quad (12)$$

Since most statistical inference methods depend on probability ratios, the condition (11) of differential privacy implies that statistical inference for the data is not easy (e.g., lower bounds on the variance of unbiased estimators [40]).

By Definition 2, the sequential statistical model checking Algorithm 1 is not differentially private, since for any fixed input sequence $\sigma_{1:}\infty$, its termination time and output $(\tau_{\mathcal{A}}(\sigma_{1:}\infty), o_{\mathcal{A}}(\sigma_{1:}\infty))$ are deterministic. That does not satisfy the necessary condition of differential privacy (12).

**B. CHALLENGES IN ACHIEVING STANDARD DIFFERENTIAL PRIVACY BY EXPONENTIAL MECHANISM**

The standard approach to make deterministic algorithms differentially private is to randomize their outputs by the exponential algorithm [22]. However, we show below that the exponential algorithm cannot achieve the differential privacy from Definition 2 for the statistical model checking method shown in Algorithm 1.

For simplicity, consider only the observation of the termination time $\tau_{\mathcal{A}}(\sigma_{1:}\infty)$. We can define the sensitivity of termination time to changes in any single entry by

$$\delta_{\tau_{\mathcal{A}}} := \max_{\sigma_{-i}} |\tau_{\mathcal{A}}(\sigma_{1:}\infty) - \tau_{\mathcal{A}}(\sigma'_{1:}\infty)|, \quad (13)$$

i.e., the maximal difference in termination time for any two adjacent input sequences of sample signals $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$.

If the sensitivity $\delta_{\tau_{\mathcal{A}}}$ is finite, then the following exponential algorithm achieves differential privacy. The exponential mechanism states that randomizing algorithm $\mathcal{A}$ to $\mathcal{B}$ such that the probability of terminating at step $k$ is

$$P_{\mathcal{B}}(\tau_{\mathcal{B}} = k) = e^{-\varepsilon |\tau_{\mathcal{A}}(\sigma_{1:}\infty) - k|/\delta_{\tau_{\mathcal{A}}}} \sum_{h \in \mathcal{H}} e^{-\varepsilon |\tau_{\mathcal{A}}(\sigma'_{1:}\infty) - h|/\delta_{\tau_{\mathcal{A}}}}. \quad (14)$$

where $\mathcal{H} \subseteq \mathbb{N}$ is the set of possible values of $\tau_{\mathcal{A}}$, and $P_{\mathcal{B}}$ means that the randomness should come from the algorithm $\mathcal{B}$ itself. The exponential mechanism (14) is $2\varepsilon$-differentially private [22], since for any two adjacent $\sigma_{1:}\infty$ and $\sigma'_{1:}\infty$, it holds by the definition of sensitivity that for any $k \in \mathcal{N}$,

$$e^{-\varepsilon |\tau_{\mathcal{A}}(\sigma_{1:}\infty) - k|/\delta_{\tau_{\mathcal{A}}}} \leq e^{-\varepsilon |\tau_{\mathcal{A}}(\sigma'_{1:}\infty) - k|/\delta_{\tau_{\mathcal{A}}}} \leq e^\varepsilon, \quad (15)$$

where the first inequality holds by applying the triangle inequality. Since (15) holds for any $k$, we have

$$\sum_{h \in \mathcal{H}} e^{-\varepsilon |\tau_{\mathcal{A}}(\sigma_{1:}\infty) - h|/\delta_{\tau_{\mathcal{A}}}} \leq e^\varepsilon. \quad (16)$$

Thus, the probability ratio satisfies

$$P_{\mathcal{B}}(\tau_{\mathcal{B}}(\sigma_{1:}\infty) = k) \leq e^{2\varepsilon}. \quad (17)$$
The above technique depends critically on the boundedness of the sensitivity $\delta_{\tau_{\varphi}}$, which is generally true for non-sequential algorithms [22]. However, this condition is violated for Algorithm 1, as shown in Example 2.

**Example 2:** Consider two adjacent input sequences of sample signals with

$$\varphi(\sigma_{1:\infty}) = (1, \ldots, 1, [0, 1, 0, \ldots])$$

and

$$\varphi(\sigma'_{1:\infty}) = (1, \ldots, 1, [0, 1, 0, \ldots])$$

in Algorithm 1, where $K_1$ satisfies

$$\frac{(p + \delta)K_1 + 1}{(p - \delta)K_1 + 1} \geq 1 - \alpha, \quad \frac{(p + \delta)K_1}{(p - \delta)K_1} < \frac{1 - \alpha}{\alpha}. \quad (18)$$

Recall from equation (3) that $\varphi(\sigma) = 1$ if $\sigma_i \neq \varphi$ and $\varphi(\sigma_i) = 0$ if $\sigma_i \neq \varphi$. The definition of $\lambda$ in equation (8), Algorithm 1 stops at $\tau_{\varphi}(\sigma) = K_1 + 1$ for the input sequence of sample signals $\sigma_{1:\infty}$, and it stops at $\tau_{\varphi}(\sigma') = \infty$ for the other input sequence of sample signals $\sigma'_{1:\infty}$. This shows that the sensitivity $\delta_{\tau_{\varphi}} = \infty$ and the exponential mechanism cannot achieve $\varepsilon$-differential privacy for any finite $\varepsilon$. Finally, we note that the case of $\tau_{\varphi}(\sigma') = \infty$ happens with probability 0, thus it does not violate the fact that the algorithm $A$ stops almost surely with probability 1.

**IV. EXPECTED DIFFERENTIAL PRIVACY**

Section III-B has shown that the exponential mechanism cannot achieve the standard differential privacy (Definition 2) for the statistical model checking Algorithm 1. In this section, we proposed a new notion called expected differential privacy (EDP), which is weaker than the standard differential privacy. We will show that EDP is achievable by a new type of exponential mechanism in Section V.

One assumption that we make in this paper that has not been considered in standard differential privacy (Definition 2) is that the sequences of signals that the algorithm processes are independently drawn from an underlying probability distribution. This is the probability model $\mathcal{S}$. This is a stronger assumption than the arbitrarily-valued static databases assumed in the literature of differential privacy [39]. Such a stronger assumption helps in defining a relaxed notion of privacy that is based on limiting the change of the average of the algorithm’s output over the distribution of input sequences with respect to arbitrary changes of an arbitrary entry. In contrast, standard differential privacy is based on limiting the change of the algorithm’s output for any input with respect to arbitrary changes of any of the input’s entries. In the case of Example 2, as we will show in this section, the average termination time of Algorithm 1 over the distribution of input sequences is bounded. Thus, the probability that an input leads to an infinite termination time as the one shown in Example 2 is negligible.

By Definition 2, if a randomized sequential algorithm $B$ is differentially private, the difference in the probability of the observations $(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty}))$ should be bounded under the change of a single entry $\sigma_n$ for any input sequence $\sigma_{1:\infty}$. However, under the assumption in this paper, all the sample signals in $\sigma_{-n}$ are drawn independently from the probabilistic model $\mathcal{S}$. We build on this assumption and propose an expected version of the standard differential privacy (Definition 2) that requires the boundedness of the sensitivity of the average output and termination time.

**Definition 3:** Denote $[\mathcal{H}_{null}, \mathcal{H}_{alt}]$ by $\{1, 0\}$. A randomized sequential algorithm $B$ is $\varepsilon$-expectedly differentially private if

$$P_B(\mathcal{E}_{\sigma_{-n}}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O) \leq e^\varepsilon P_B(\mathcal{E}_{\sigma'_{-n}}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O), \quad (19)$$

for any $n \in \mathbb{N}$, any $O \subseteq \mathbb{N} \times \{1, 0\}$, and any pair of signals $\sigma_n$ and $\sigma'_n$, where $\sigma_{-n}$ and $\sigma'_{-n}$ are random sequences with i.i.d. entries following the probabilistic model $\mathcal{S}$. We say that $B$ is $\varepsilon$-expectedly differentially private in termination time if

$$P_B(\mathcal{E}_{\sigma_{-n}}(\tau(\sigma_{1:\infty})) \in O) \leq e^\varepsilon P_B(\mathcal{E}_{\sigma'_{-n}}(\tau(\sigma'_{1:\infty})) \in O), \quad (20)$$

for any $O \subseteq \mathbb{N}$, $\sigma_n$, and $\sigma'_{n}$.

The EDP (Definition 3) is weaker than the standard differential privacy (Definition 2) since the condition of the latter (10) implies the former (19) by taking the expected values $\mathbb{E}_{\sigma_{-n}}\mathbb{E}_{\sigma'_{-n}}$ on both sides of the equation, i.e.,

$$\mathbb{E}_{\sigma_{-n}}\mathbb{E}_{\sigma'_{-n}} P_B(\mathcal{E}_{\sigma}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$\leq e^\varepsilon \mathbb{E}_{\sigma_{-n}}\mathbb{E}_{\sigma'_{-n}} P_B(\mathcal{E}_{\sigma'(\sigma'_{1:\infty}), o_B(\sigma'_{1:\infty})) \in O),$$

for any $O \subseteq \mathbb{N}$, $\sigma_n$, and $\sigma'_{n}$. This implies that

$$\mathbb{E}_{\sigma_{-n}} P_B(\mathcal{E}_{\sigma}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$\leq e^\varepsilon \mathbb{E}_{\sigma_{-n}} P_B(\mathcal{E}_{\sigma'}(\tau_B(\sigma'_{1:\infty}), o_B(\sigma'_{1:\infty})) \in O).$$

Note that

$$\mathbb{E}_{\sigma_{-n}} P_B(\mathcal{E}_{\sigma}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$= \mathbb{E}_{\sigma_{-n}} \mathbb{E}_\sigma \mathbb{I}(\mathcal{E}_{\sigma}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$= \mathbb{E}_\sigma \mathbb{E}_{\sigma_{-n}} \mathbb{I}(\mathcal{E}_{\sigma}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$\leq e^\varepsilon \mathbb{E}_{\sigma_{-n}} P_B(\mathcal{E}_{\sigma'}(\tau_B(\sigma'_{1:\infty}), o_B(\sigma'_{1:\infty})) \in O),$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\mathbb{E}_{\sigma_{-n}}$ and $\mathbb{E}_\sigma$ commute by Fubini’s Theorem. Thus, we can flip $\mathbb{E}_{\sigma_{-n}}$ and $P_B$ (or $\mathbb{E}_{\sigma'_{-n}}$ and $P_B$) to derive the following inequality:

$$P_B(\mathcal{E}_{\sigma_{-n}}(\tau_B(\sigma_{1:\infty}), o_B(\sigma_{1:\infty})) \in O)$$

$$\leq e^\varepsilon P_B(\mathcal{E}_{\sigma'_{-n}}(\tau_B(\sigma'_{1:\infty}), o_B(\sigma'_{1:\infty})) \in O).$$

**V. EXPECTED DIFFERENTIAL PRIVACY FOR STATISTICAL MODEL CHECKING**

In this section, we show that the termination time and output of Algorithm 1 can be randomized, to achieve the EDP. Our approach is based on a new sensitivity analysis and a novel exponential mechanism for randomizing the termination step.
A. ANALYSIS OF TERMINATION TIME FOR STATISTICAL MODEL CHECKING

The termination time and the output $(\tau_{\psi}, o_{\psi})$ of Algorithm 1 depend on the log-likelihood ratio $\lambda(N, K)$ in equation (8). We consider the log-likelihood ratio as an easier variable for analysis:

$$\Lambda_N = \ln \lambda(K, N) = Ks_+ - (N - K)s_-,$$  \hspace{1cm} (21)

$$s_+ := \ln \frac{p + \delta}{p - \delta} > 0, \quad s_- := \ln \frac{1 - p + \delta}{1 - p - \delta} > 0,$$  

and

$$K = \sum_{i \in [N]} \varphi(\sigma_i)$$

is a random variable that follows the binomial distribution $\text{Binom}(N, p_{\psi})$.

Lemma 1: The log-likelihood ratio $\Lambda_N$ forms an asymmetric random walk with probabilities and step sizes being $(p_{\psi}, s_+)$ and $(1 - p_{\psi}, -s_-)$.

Proof: By equation (21), we have $\Lambda_0 = 0$ and

$$\Lambda_{N+1} = \begin{cases} \Lambda_N + s_+, & \text{with probability } p_{\psi} \\ \Lambda_N - s_-, & \text{with probability } 1 - p_{\psi}. \end{cases}$$

By Lemma 1, Algorithm 1 can be interpreted as a random walk that stops upon hitting the upper or lower bounds:

$$B_+ := \ln \frac{1 - \alpha}{\alpha} > 0, \quad B_- := -\ln \frac{1 - \alpha}{\alpha} < 0.$$  \hspace{1cm} (22)

From Lemma 1, the average step size of the random walk $\Lambda_N$ is

$$D := \mathbb{E}_{\sigma_1} \Lambda_1 = p_{\psi}s_+ - (1 - p_{\psi})s_- = p_{\psi} \ln \frac{p + \delta}{p - \delta} - (1 - p_{\psi}) \ln \frac{1 - p + \delta}{1 - p - \delta}.$$  \hspace{1cm} (23)

When $B_+ \gg 1$ (equivalently, $B_- = -B_+ \ll -1$), we can prove that if $p_{\psi} > p - \delta$, then the average step size is positive and the asymmetric random walk $\Lambda_N$ will hit the upper bound $B_+$ with probability $1 - \alpha$ and the lower bound $B_-$ with probability $\alpha$. Similarly, if $p_{\psi} < p - \delta$, then the average step size is negative and the asymmetric random walk $\Lambda_N$ will hit the lower bound $B_-$ with probability $1 - \alpha$ and the upper bound $B_+$ with probability $\alpha$.

The average termination time $\tau_{\psi}(\sigma_1:\infty)$ satisfies the stopping time property of random processes [36], i.e.,

$$\mathbb{E}_{\sigma_1\infty}[\Lambda_{\tau_{\psi}(\sigma_1:\infty)}] = \mathbb{E}_{\sigma_1\infty}[\lambda_{\psi}(\sigma_1:\infty)] = D \mathbb{E}_{\sigma_1\infty}[\tau_{\psi}(\sigma_1:\infty)].$$  \hspace{1cm} (24)

B. EXPECTED SENSITIVITY ANALYSIS FOR STATISTICAL MODEL CHECKING

Now we define the sensitivity of the average termination time $\tau_{\psi}(\sigma_1:\infty)$ to single entry perturbations in the input sequence in the following equation:

$$\Delta_{\tau_{\psi}} := \max_{n \in \mathbb{N}, \sigma_n, \sigma_n'} |\mathbb{E}_{\sigma_n \psi}(\tau_{\psi}(\sigma_1:\infty)) - \mathbb{E}_{\sigma_n' \psi}(\tau_{\psi}(\sigma_1:\infty))|.$$  \hspace{1cm} (25)

where $\sigma_n$ and $\sigma_n'$ are two random sequences of signals obeying the distribution of the probabilistic model $S$. We call $\Delta_{\tau_{\psi}}$ the expected sensitivity of termination time. Expected sensitivity $\Delta_{\tau_{\psi}}$ represents the maximal change in the average termination time over input sequences sampled from the probabilistic model $S$ when arbitrarily changing the value of an arbitrary $n$th entry.

For Algorithm 1, we can use equation (24) to compute the expected sensitivity as shown in the following lemmas.

Lemma 2: For $D > 0$ in equation (23), consider the random walk $\Lambda_N$ from Lemma 1 for hitting any two absorbing bounds $A \gg \max\{s_+, s_-\}$ and $-B \ll -\max\{s_+, s_-\}$, the probability of hitting $B$ is $\approx e^{-B}$ and the expected termination time is given by

$$\mathbb{E}_{\sigma_1\infty}[\tau_{\psi}(\sigma_1:1)] \approx \frac{A(1 - e^{-B}) - Be^{-B}}{D}.$$  \hspace{1cm} (26)

Proof: Let $\alpha$ and $\beta$ be the probabilities of hitting the bounds $A$ and $B$ respectively. We have $\alpha + \beta = 1$, since the random walk terminates with probability 1. For $D > 0$, when $A \gg \max\{s_+, s_-\}$ and $-B \ll -\max\{s_+, s_-\}$, the stopping time property (24), we have the lemma holds.

Lemma 3: The expected sensitivity of the termination time $\tau_{\psi}$ satisfies

$$\Delta_{\tau_{\psi}} = \frac{s_+ + s_-}{|D|} = \frac{\ln \frac{p + \delta}{p - \delta} + \ln \frac{1 - p + \delta}{1 - p - \delta}}{|p_{\psi} \ln \frac{p + \delta}{p - \delta} - (1 - p_{\psi}) \ln \frac{1 - p + \delta}{1 - p - \delta}|}.$$  \hspace{1cm} (26)

where $s_+$ and $s_-$ are from (21) and $D$ is from (23).

Proof: Without loss of generality, consider the case of $D > 0$ in (23). The case of $D < 0$ can be proved similarly. Consider any two input sequences $\sigma_1:1$ and $\sigma_1:2$ for Algorithm 1, where $\sigma_1$ and $\sigma_1'$ are chosen such that $\sigma_1 \equiv \varphi$ and $\sigma_1' \not\equiv \varphi$, and $\sigma_2:1$ and $\sigma_2:2$ are drawn in an i.i.d. manner from the probabilistic model $S$. Then, for the input sequence $\sigma_1:1$, the log-likelihood ratio $\Lambda_0$ follows a random walk that starts at $s_+$ after the first step. To derive the termination time to hit either $B_+$ or $B_-$ from (22), we set $A := B_+ - s_-$ and $B := B_- - s_-$ in Lemma 2 to obtain

$$\mathbb{E}_{\sigma_2:1\infty}[\tau_{\psi}(\sigma_1:1)] = 1 + \frac{(B_+ - s_-)(1 - e^{B_+ - s_-}) + (B_- - s_+)e^{B_- - s_+}}{D}.$$  \hspace{1cm} (26)

Similarly, by setting $A = B_+ + s_-$ and $B = B_- + s_-$ in Lemma 2 as the bounds for the random walk of the log-likelihood ratio $\Lambda_0'$ of $\sigma_2:2$, we have

$$\mathbb{E}_{\sigma_2:2\infty}[\tau_{\psi}(\sigma_1:1)] = 1 + \frac{(B_+ + s_-)(1 - e^{B_+ + s_-}) + (B_- + s_+)e^{B_- + s_-}}{D}.$$  \hspace{1cm} (26)
Considering $s_+,-s_- \ll B_+, -B_-$, the subtraction of the two equations shown above gives
\[
\mathbb{E}_{\sigma_2} \tau_{\sigma_1}(\sigma_1) = \mathbb{E}_{\sigma_2'} \tau_{\sigma_1'}(\sigma_1') \approx \frac{s_+ + s_-}{D},
\]
considering that $e^{B_-} \ll 1$. This proves the bound of expected sensitivity (25) stated in the lemma for alternating the first entry. The case of other entries can be proved similarly. ■

When the expected sensitivity $\Delta_{\tau_{\sigma}} < \infty$ is finite, we can apply a modified version of the exponential mechanism described in Section III-B to Algorithm 1 to achieve the EDP for the termination time. This is shown in the following lemma.

**Lemma 4**: For a sequential algorithm $\mathcal{A}$ with a deterministic termination time $\tau_{\mathcal{A}}$, consider another sequential algorithm $\mathcal{B}$ with the same input and output spaces as $\mathcal{A}$ but with a random termination time $\tau_{\mathcal{B}}$ that satisfies
\[
\mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_1}^\dagger \tau_{\mathcal{B}}(\sigma_1) = k) = \frac{e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}}{\sum_{h \in \mathcal{H}} e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}},
\]
where $\sigma_1$ is a random sequence of signals obeying the distribution of the probabilistic model $S$, and $\mathcal{H} \subseteq \mathbb{N}$ is the set of possible values of $\mathcal{A}$. Then, it is $e$-expectedly differentially private in term of time.

**Proof**: Without loss of generality, assuming $D > 0$ in equation (23), we will prove the condition (20) of Definition 3 for $n = 1$, i.e.,
\[
\mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_1} \tau_{\mathcal{B}}(\sigma_1) = k) \leq e^2,
\]
for any pair of signals $\sigma_1$ and $\sigma_1'$, where $\sigma_2$ and $\sigma_2'$ are random sequences of i.i.d. signals obeying the distribution of the probabilistic model $S$. For simplicity, we consider the case of $\varphi(\sigma_1) = 1$ and $\varphi(\sigma_1') = 0$. The case of $\varphi(\sigma_1) = 0$ and $\varphi(\sigma_1') = 1$ can be handled similarly. The cases of $\varphi(\sigma_1) = \varphi(\sigma_1') = 0$ and $\varphi(\sigma_1) = \varphi(\sigma_1') = 1$ are trivial.

When $\varphi(\sigma_1) = 1$, by the stopping time property (24), we have
\[
\mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_2} \tau_{\mathcal{B}}(\sigma_1) = k) = \mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_2} \tau_{\mathcal{B}}(\sigma_2) = k - \frac{s_+}{D})
\]
where $s_+$ is given by (21) with $\mathbb{E}_{\sigma_2} \Delta_{\tau_{\mathcal{B}}}(\sigma_1) - \mathbb{E}_{\sigma_2'} \Delta_{\tau_{\mathcal{B}}}(\sigma_1) = s_+$.

By viewing $\sigma_2$ as a new random walk, we can apply (27) and derive
\[
\mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_2} \tau_{\mathcal{B}}(\sigma_2) = k - \frac{s_+}{D})
\]
\[
= \frac{e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}}{\sum_{h \in \mathcal{H}} e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}.
\]
Similarly, we have
\[
\mathbb{P}_{\mathcal{B}}(\mathbb{E}_{\sigma_2'} \tau_{\mathcal{B}}(\sigma_1') = k)
\]
\[
= \frac{e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}}{\sum_{h \in \mathcal{H}} e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}.
\]

By the triangular inequality and using the definition of $\Delta_{\tau_{\sigma}}$ from (26), the ratio of the numerators of equations (29) and (30) satisfies
\[
e^{-\epsilon} \leq \frac{e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}}{\sum_{h \in \mathcal{H}} e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}} \leq e^\epsilon.
\]

Since (31) holds for any $k$, we have
\[
e^{-\epsilon} \leq \frac{e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}}}{\sum_{h \in \mathcal{H}} e^{-e^{-\frac{1}{\Delta_{\tau_{\sigma}}}}} \leq e^\epsilon.
\]

Thus, by taking the ratios of (31) and (32) and applying (29) and (30) results in that for any $k \in \mathbb{N}$, we can derive (28).

The condition (20) of Definition 3 for $n \neq 1$ can be proved similarly. ■

**C. CAUSAL RANDOMIZATION FOR DIFFERENTIAL PRIVACY OF TERMINATION TIME**

In this section, we propose a causal randomization mechanism of the termination time of Algorithm 1 that achieves the probability distribution in (4).

A trivial addition of random noise to the deterministic termination time is not causal. If the deterministic termination time plus the random noise is $n \leq k = \tau_{\mathcal{A}}(\sigma_1)$, the randomized algorithm can draw additional $n - \tau_{\mathcal{A}}$ samples and ignore their values. However, if $n < k = \tau_{\mathcal{A}}(\sigma_1)$, then the randomized algorithm $\mathcal{B}$ will not be able to causally predict that it should stop before reaching the deterministic termination time of $\mathcal{A}$. This shows the difficulty in directly randomizing the termination time of Algorithm 1 to achieve the EDP.

Instead, we propose to randomize the stopping condition of Algorithm 1. Recall from (22) that Algorithm 1 stops upon reaching the upper or lower bounds $B_+$ and $B_-$, respectively. We modify these bounds as follows:
\[
B_+ = \ln \frac{1 - \alpha}{\alpha} + L \quad \text{and} \quad B_- = \ln \frac{1 - \alpha}{\alpha} - L,
\]
where
\[
L \sim \exp\left(\frac{\epsilon}{\ln \frac{p+\epsilon}{\epsilon + 1} - \frac{\epsilon}{\epsilon + 1}}\right).
\]

The modified version of Algorithm 1 is shown in Algorithm 2.

**Lemma 5**: The termination time of Algorithm 2 satisfies the distribution shown in (27).

**Proof**: We denote Algorithm 2 by $\mathcal{B}$ and let $B = \ln \frac{1 - \alpha}{\alpha}$. Without loss of generality, suppose that the average step size $D$ from (23) of the random walk $A_N$ is $> 0$. Following the randomized stopping condition in (33), we have that for any $k \in \mathbb{N}$,
\[
\mathbb{P}_L(\mathbb{E}_{\sigma_1} \tau_{\mathcal{B}}(\sigma_1) = k) = \mathbb{P}_L(\mathbb{E}_{\sigma_1} \Delta_{\mathcal{B}}(\sigma_1) \geq B + L)
\]
and
\[
\mathbb{E}_{\sigma_1} \Delta_{\mathcal{B}}(\sigma_1) \geq B + L
\]
where $\mathbb{E}_{\sigma_1} \Delta_{\mathcal{B}}(\sigma_1)$ is increasing in $k$ when $D > 0$.
It directly follows from Lemmas 5 and 6.

Algorithm 1: Statistical Model Checking of $\mathbb{P}_{\sigma \sim S}(\sigma \models \varphi) < p$ With Expected Differential Privacy.

**Require:** Probabilistic model $S$, desired significance level $\alpha$, and indifference parameter $\delta$, privacy level $\epsilon$.

1. $N \leftarrow 0$, $K \leftarrow 0$.
2. Log-likelihood ratio $\Lambda \leftarrow 0$.
3. $s_+ \leftarrow \ln \frac{1+\delta}{p-\delta}$, $s_- \leftarrow \ln \frac{1-p+\delta}{1-p-\delta}$, $B \leftarrow \ln \frac{1-\alpha}{\alpha}$.
4. $L \sim \text{Exp}(\frac{\epsilon}{s_+ + s_-})$, $B \leftarrow B + L$.
5. **while** True **do**
   6. Draw a sample signal $\sigma$ from $S$.
   7. $K \leftarrow K + \varphi(\sigma)$, $N \leftarrow N + 1$.
   8. $\Lambda \leftarrow \Lambda + \varphi(\sigma)s_+ + (1 - \varphi(\sigma))s_-$.
   9. **if** $\Lambda \geq B$ **then**
      10. Return $H_{\text{null}}$.
      11. **else** if $\Lambda \leq -B$ **then**
          12. Return $H_{\text{alt}}$.
          13. **else** Continue
          14. **end if**
      15. **end while**

This leads to (27) since $E_{\sigma_{1:\infty}}[\tau_{\varphi}(\sigma_{1:\infty})] \approx B/D$ from Lemma 2.

Following the same process, we can apply the expected sensitivity analysis for $o_{\varphi}$ for Algorithm 1.

**Lemma 6:** The expected sensitivity $\Delta_{o_{\varphi}}$ of the output $o_{\varphi}$ of Algorithm 1 is $0$, where $\Delta_{o_{\varphi}}$ is defined as

$$\Delta_{o_{\varphi}} := \max_{\sigma_{1:}\sigma_{1:}} |E_{\sigma_{1:}}[o_{\varphi}(\sigma_{1:})] - E_{\sigma^\prime_{1:}}[o_{\varphi}(\sigma^\prime_{1:})]|, \ (34)$$

and $\sigma_{1:}$ and $\sigma^\prime_{1:}$ are random sequences of i.i.d. signals obeying the distribution of the probabilistic model $S$.

**Proof:** Consider $n = 1$. For $D > 0$, Algorithm 1 returns $0$ by hitting the bound $-B$ and returns $1$ by hitting the bound $B$, as defined in Definition 3. By Lemma 2, when $\sigma_1 \models \varphi$, we have $F_{\sigma_{1:}}[o_{\varphi}(\sigma_{1:})] = 0 = e^{-B+s_{-}}$ by viewing the log-likelihood ratio $\Lambda_N$ in Algorithm 1 as a random walk starting from $s_{+}$ after the first step, instead of starting from zero in the first step. Similarly, when $\sigma_1 \not\models \varphi$, we have $F_{\sigma_{1:}}[o_{\varphi}(\sigma_{1:})] = 0 = e^{-B-s_{-}}$. Thus, we have

$$\Delta_{o_{\varphi}} := \max_{\sigma_1, \sigma_1^\prime} |E_{\sigma_{1:}}[o_{\varphi}(\sigma_{1:})] - E_{\sigma^\prime_{1:}}[o_{\varphi}(\sigma^\prime_{1:})]| \leq |e^{-B+s_{+}} - e^{-B-s_{-}}| \approx 0. \ (35)$$

The same analysis holds for $n \neq 1$.

Thus, changing an arbitrary $n$-th sample has almost no influence on the output of algorithm $\varphi$ in the expected sense. In sum, we present the following theorems.

**Theorem 1:** Algorithm 2 is $2\epsilon$-expectedly differentially private.

**Proof:** It directly follows from Lemmas 5 and 6.

In addition to privacy, we provide the following results for the significance level of Algorithm 2.

**Theorem 2:** Algorithm 2 has a significance level (i.e., the upper bound on the probability that this algorithm returns a wrong answer) less than $\alpha$.

**Proof:** In Algorithm 2, the hitting bounds $B + L$ and $-B - L$ are always expanded versions of those of Algorithm 1 since $L \geq 0$. Hence, the accuracy is improving because of drawing more samples. Specifically, following the standard analysis of sequential hypotheses testing [36], for any value of $L$ in Algorithm 2, the significance level $\alpha_L$ for hitting the bounds $B + L$ and $-B - L$ satisfies $1 - \alpha_L = \ln B + \leq \frac{1 - \alpha}{\alpha}$ since $L \geq 0$ in (33). This implies $\alpha_L \leq \alpha$ for any value of $L$. Thus, the theorem holds.

Finally, the causal randomization for differential privacy of termination time can potentially be applied to the statistical model checking of hyperproperties, which also uses the sequential probability ratio test to infer the true value of formulas [41].

**VI. CASE STUDY**

In this section, we apply the statistical model checking with EDP (Algorithm 2) to the Toyota Powertrain benchmark [42] in MATLAB Simulink. It is composed of an air-to-fuel (A/F) ratio controller and a model of a four-cylinder spark ignition engine that includes components starting from the throttle all the way to the crankshaft. This case study focuses on the performance of the A/F ratio controller by observing the deviation percentage of the A/F ratio $\mu(t)$ from a reference A/F ratio $\mu_{\text{ref}}$ as follows

$$e_{A/F}(t) := \frac{\mu(t) - \mu_{\text{ref}}}{\mu_{\text{ref}}}$$

over a simulation time horizon of $T$ seconds. The engine’s revolutions per minute (RPM) speed follows the Gaussian distribution

$$r \sim \text{Gaussian}(r_0, \sigma^2),$$

where $r_0 = 1600$, and $\sigma^2 = 1600$. The requirement for $e_{A/F}$ is to enter a desired region $|e_{A/F}| < 0.05$ within the time interval $[0.8, T]$ [43]. For a given powertrain $S$, we want to check if this requirement holds with probability greater than a desired threshold $p$. Formally, in the STL syntax introduced in Section II, we are interested in checking the following property:

$$\mathbb{P}_\sigma (\sigma \models \Diamond_{[0.8,T]}(|e_{A/F}| < 0.05)) > p. \ (36)$$
where \( T = 1 \) and \( p = 0.73 \). Evaluations were performed on a desktop with Intel Core i7-10700 CPU @ 2.90 GHz and 16 GB RAM.

Results Analysis: We applied Algorithm 2 to analyze the Toyota Powertrain with different combinations of the significance level, indifference parameter, and privacy parameters: \( \alpha \in \{0.01, 0.05\} \), \( \delta \in \{0.01, 0.03\} \), and \( \varepsilon \in \{0.01, 0.05\} \), respectively.

We estimated the satisfaction probability \( p_{\psi} \approx 0.84 \) with a standard deviation of about 0.04 using 10000 random samples. Thus, Assumption 1 for implementing Algorithm 2 holds since \( p_{\psi} - p \approx 0.11 \), which is approximately 2.75 times the standard deviation and is greater than both values of \( \delta \) considered. In addition, we know from the estimated \( p_{\psi} \) that the STL specification (36) is true, so Algorithm 2 should return the null hypothesis (\( H_{\text{null}} \)) with probability at least \( 1 - \alpha \). We ran Algorithm 2 \( N = 10^4 \) times for each of the considered combinations of parameters. Then, we calculated the algorithm’s accuracy (Acc.) for each combination. The Acc. is defined as follows:

\[
\text{Acc.} := \frac{1}{N} \sum_{i=1}^{N} I(o_i = H_{\text{null}}), \tag{37}
\]

where \( I \) is the indicator function and \( o_i \) is the output of the \( i \)th run. Additionally, the average termination time (\( \tau_{\phi} \)), the average span of computation in seconds (Span), and the predicted hypothesis (\( H_{\text{null}} \) or \( H_{\text{alt}} \)) were computed and presented in Table 1.

We then analyzed the differential privacy of Algorithm 2 by considering \( M \) pairs of sequences of samples \( \sigma_1, \sigma_\infty \) and \( \sigma_1', \sigma_\infty' \) that differed in the \( n \)th entry. Each entry in a sequence represents the satisfaction of \( \psi \) for the sampled execution, as defined in equation (3). The average termination time over \( M \) pairs of sequences (ATT) is as follows:

\[
\text{ATT} := \frac{1}{M} \sum_{i=1}^{M} \tau_{\phi}^{(i)}(\cdot), \tag{38}
\]

where \( M = 500 \) and \( \tau_{\phi}^{(i)}(\cdot) \) represents the termination time for the \( i \)th sample value of \( \sigma_1, \sigma_\infty \) and \( \sigma_1', \sigma_\infty' \). The ATT was calculated for each of \( 10^4 \) samples of \( L \), as defined in (33), and the resulting distribution can be seen for one of the parameter combinations in Fig. 1.

Discussion: In Table 1, the simulation corresponding to each parameter combination yielded an accuracy of 1.00, which agrees with the confidence level \( 1 - \alpha \). The high accuracy results from the fact that we chose a small indifference parameter \( \delta \), which increases termination time and thus the number of samples. This can be seen by observing that in the likelihood ratio definition in (8), increasing \( \delta \) (while satisfying Assumption 1) will reduce the termination time. Relaxing the confidence level from 0.99 to 0.95 while holding the indifference parameter \( \delta \) and privacy level \( \varepsilon \) constant reduced the average number of samples needed to output the null hypothesis.

The effects of increasing privacy level while holding the confidence level and indifference parameter constant can be seen in Table 1 and Fig. 1. Increasing \( \varepsilon \) decreases the privacy level and concentrates the distribution of \( L \) in Algorithm 2, leading to less randomness in termination time. However, this comes at the expense that it becomes easier to infer user data from sample distribution and statistical model checking output. From Table 1, the average number of samples decreased as \( \varepsilon \) increased. In Fig. 1, there are a few places where the distributions fall outside the bounds. This happens because of two reasons: 1) the statistical error in approximating the
distribution of the termination time with samples, and 2) the deviation of sample distributions from the expected case.

VII. CONCLUSION
This paper studied the privacy issue of statistical model checking to enable their applications in privacy-critical applications. We used differential privacy to mathematically capture the privacy level of sample system executions that are used for statistical model checking. Since the algorithms are sequential, we showed that the termination time can violate privacy and the standard exponential mechanism fails to achieve standard differential privacy. We proposed a new exponential mechanism that can achieve a new notion of privacy which we call expected differential privacy. Using the exponential mechanism, we developed expected-differentially private statistical model checking algorithms. The utility of the proposed algorithm was demonstrated in a case study.

REFERENCES
[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, “Cyber-physical systems: The next computing revolution,” in Proc. Des. Automat. Conf., 2010, pp. 731–736.
[2] X. Jin, J. V. Deshmukh, J. Kapinski, K. Ueda, and K. Butts, “Powertrain control verification benchmark,” in Proc. 17th Int. Conf. Hybrid Syst.: Comput. Control, 2014, pp. 253–262.
[3] I. Daniele, F. Alessandro, H. Marianne, B. Axel, and P. Maria, “A smart grid energy management problem for data-driven design with probabilistic reachability guarantees,” in Proc. 4th Int. Workshop Appl. Verification Continuous Hybrid Syst., 2017, pp. 2–19.
[4] J. E. Lee and O. Sokolsky, “Medical cyber physical systems,” in Proc. 47th Des. Automat. Conf. Anaheim, CA, USA: ACM Press, 2010, pp. 743–748.
[5] G. Agha and K. Palmoskog, “A survey of statistical model checking,” ACM Trans. Model. Comput. Simul., vol. 28, no. 1, pp. 1–39, Jan. 2018.
[6] R. Alur, Principles of Cyber-Physical Systems. Cambridge, MA, USA: MIT Press, 2015.
[7] S. Coogan and M. Arcak, “Efficient finite abstraction of mixed monotone systems,” in Proc. 18th Int. Conf. Hybrid Syst. Comput. Control, Seattle, Washington: ACM Press, 2015, pp. 58–67.
[8] N. Roohi, P. Prabhakar, and M. Viswanathan, “HARE: A hybrid abstraction refinement engine for verifying non-linear hybrid automata,” in Proc. Tools Algorithms Construction Anal. Syst. Ser. Lecture Notes in Computer Science, vol. 9779, pp. 3–13, 2016.
[9] H. Sibai, N. Mokhlesi, and S. Mitra, “Using symmetry transformations in equivariant dynamical systems for their safety verification,” in Proc. Automated Technol. Verification Anal. Ser. Lecture Notes Comput. Sci., 2017.
[10] B. Barbot, B. Béard, Y. Duplouy, and S. Haddad, “Statistical model-checking for autonomous vehicle safety validation,” in Proc. Conf. SIA Simul. Numérique, Société des ingénieurs de l’automobile, 2017, pp. 1–8.
[11] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “Statistical verification of PCTL using antithetic and stratified samples,” Formal Methods Syst. Des., vol. 54, pp. 143–163, 2019.
[12] S. Zializedi, “Statistical model checking for biological applications,” Int. J. Softw. Tools Technol. Transfer, vol. 17, no. 4, pp. 527–536, Aug. 2015.
[13] J. Ma, J. Hu, E. Leslie, F. Zhou, P. Huang, and J. Bared, “An eco-drive experiment on rolling terrains for fuel consumption optimization with connected automated vehicles,” Transp. Res. Part C: Emerg. Technol., vol. 100, pp. 125–141, 2019.
[14] Z. Yao, Y. Wang, B. Liu, B. Zhao, and Y. Jiang, “Fuel consumption and transportation emissions evaluation of mixed traffic flow with connected automated vehicles and human-driven vehicles on expressway,” Energy, vol. 230, 2021, Art. no. 120766.
[15] G. Oh, D. J. Leblanc, and H. Peng, “Vehicle energy dataset (VED), a large-scale dataset for vehicle energy consumption research,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 4, pp. 3302–3312, Apr. 2022.
[16] “Personal data in your car,” U.S. Dept. of Transp., Tech. Rep., 2017. [Online]. Available: https://tfp.org/wp-content/uploads/2017/01/consumerguide.pdf
[17] C. Dwork, “Differential privacy,” in Automata, Languages, and Programming, Berlin, Germany: Springer, 2006, pp. 1–12.
[18] U. Erlingsson, V. Pihur, and A. Korolova, “RAPTOR: Randomized aggregatable privacy-preserving ordinal response,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., New York, NY, USA, 2014, pp. 1054–1067.
[19] Apple Differential Privacy Team, “Learning with privacy at scale,” 2017. [Online]. Available: https://machinelearning.apple.com/research/learning-with-privacy-at-scale
[20] J. M. Abowd et al., “The modernization of statistical disclosure limitation at the U.S. Census Bureau,” Tech. Rep., 2020. [Online]. Available: https://www2.census.gov/adc/CED/Papers/CY2020-009-AbowdBenedettoGarfinkelDahetal-The%20modernization%20of.pdf
[21] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” Found. Trends Theor. Comput. Sci., vol. 9, no. 3/4, pp. 211–407, 2013.
[22] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in Proc. 48th Annu. IEEE Symp. Found. Comput. Sci., 2007, pp. 94–103.
[23] M. Ghassemi, A. D. Sarwate, and R. N. Wright, “Differentially private online active learning with applications to anomaly detection,” in Proc. ACM Workshop Artif. Intell. Secur., New York, NY, USA, 2016, pp. 117–128.
[24] P. Jain, P. Kothari, and A. Thakurta, “Differentially private online learning,” in Proc. 25th Ann. Conf. Learn. Theory, vol. 23, 2011, pp. 24.1–24.34.
[25] J. Tsitsiklis, K. Xu, and Z. Xu, “Private sequential learning,” in Proc. Conf. Learn. Theory, 2018, pp. 721–727.
[26] B. Yang, I. Sato, and H. Nakagawa, “Bayesian differential privacy on corrupted data,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, New York, NY, USA, 2015, pp. 747–762.
[27] A. Wald, “Sequential tests of statistical hypotheses,” Ann. Math. Statist., vol. 16, no. 2, pp. 117–186, 1945.
[28] A. Legay, B. Delahaye, and S. Bensalem, “Statistical model checking: An overview,” in Runtime Verification, H. Barringer et al., Eds. Berlin, Heidelberg, Germany: Springer, 2010, pp. 6418, pp. 122–135.
[29] O. Maler and D. Nickovic, “Monitoring temporal properties of continuous signals,” in Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Berlin, Heidelberg, Germany: Springer, 2004, pp. 152–166.
[30] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “Statistical verification of PCTL using stratified samples,” in Proc. IFAC Conf. Anal. Des. Hybrid Syst., Oxford, U.K., 2018, vol. 51, pp. 85–90.
[31] Y. Wang, M. Zarei, B. Bonakdarpour, and M. Pajic, “Statistical verification of hyperproperties for cyber-physical systems,” ACM Trans. Embedded Comput. Syst., vol. 18, no. 5s, pp. 1–23, 2019.
[32] N. Roohi and M. Viswanathan, “Revisiting MITL to fix decision procedures,” in Verification, Model Checking, and Abstract Interpretation, I. Dillig and J. Palsberg, Eds. Cham, Switzerland: Springer, 2018, pp. 474–494.
[33] M. Zarei, Y. Wang, and M. Pajic, “Statistical verification of learning-based cyber-physical systems,” in Proc. ACM Int. Conf. Hybrid Syst.: Comput. Control, Sydney, Australia, 2020, pp. 1–7.
[34] K. Sen, M. Viswanathan, and G. Agha, “Statistical model checking of black-box probabilistic systems,” in Computer Aided Verification, Ser. Lecture Notes in Computer Science, B. Alur and D. A. Peled, Eds. Berlin Heidelberg, Germany: Springer, 2004, pp. 202–215.
[35] N. Roohi, Y. Wang, M. West, G. E. Dullerud, and M. Viswanathan, “Statistical verification of the toyota powertrain control verification benchmark,” in Proc. ACM Int. Conf. Hybrid Syst.: Comput. Control, New York, NY, USA, 2017, pp. 65–70.
[36] G. Rosella and R. L. Berger, Statistical Inference, vol. 2. Duxbury Pacific Grove, CA, 2002.
[37] A. Narayanan and V. Shmatikov, “Robust de-anonymization of large sparse datasets,” in Proc. IEEE Symp. Secur. Privacy, Oakland, CA, USA, 2008, pp. 111–125.
HUSSEIN SIBAI received the undergraduate degree in computer and communication engineering from the American University of Beirut, Beirut, Lebanon, and the master’s and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2017 and 2021, respectively. He is currently a Postdoctoral Scholar with the Electrical Engineering and Computer Sciences Department, University of California, Berkeley, Berkeley, CA, USA. His research interests include formal methods, hybrid systems, machine learning, and control theory. He was the recipient of the Best Poster Award in HSCC 2018, and his papers have been nominated for Best Paper awards in HSCC 2017 and ATVA 2019 and the Rambus Fellowship twice in 2019 and 2020 and the Ernest A. Reid fellowship in 2021 from the ECE department at UIUC as a recognition of his research. He was a MAVIS Future Faculty Fellow for the 2020–2021 academic year.

MARK YEN received the B.S. and M.S. degrees in aerospace engineering from the University of Florida, Gainesville, FL, USA, where he is currently working toward the Ph.D. degree in aerospace engineering with the Department of Mechanical and Aerospace Engineering with the University of Florida. His research focuses on neural architecture search, neural network-based controllers, and statistical verification. He is a recipient of the Science, Mathematics, and Research For Transformation (SMART) Scholarship from the U.S. Department of Defense.