# On the Statistical Detection of Adversarial Instances over Encrypted Data

Sheikhalishahi, Mina; Nateghizad, Majid; Martinelli, Fabio; Erkin, Zekeriya; Loog, Marco

**DOI**
10.1007/978-3-030-31511-5_5

**Publication date**
2019

**Document Version**
Final published version

**Published in**
Security and Trust Management - 15th International Workshop, STM 2019, Proceedings

**Citation (APA)**
Sheikhalishahi, M., Nateghizad, M., Martinelli, F., Erkin, Z., & Loog, M. (2019). On the Statistical Detection of Adversarial Instances over Encrypted Data. In S. Mauw, & M. Conti (Eds.), *Security and Trust Management - 15th International Workshop, STM 2019, Proceedings* (Vol. 11738, pp. 71-88). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 11738 LNCS). Springer. https://doi.org/10.1007/978-3-030-31511-5_5

# On the Statistical Detection of Adversarial Instances over Encrypted Data

Mina Sheikhalishahi[1(✉)], Majid Nateghizad[2], Fabio Martinelli[1], Zekeriya Erkin[2], and Marco Loog[2]

[1] Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche (CNR), Pisa, Italy
{mina.sheikhalishahi,fabio.martinelli}@iit.cnr.it
[2] Department of Intelligent Systems, Delft University of Technology, Delft, The Netherlands
{m.nateghizad,z.erkin,m.loog}@tudelf.nl

**Abstract.** Adversarial instances are malicious inputs designed to fool machine learning models. In particular, motivated and sophisticated attackers intentionally design adversarial instances to evade classifiers which have been trained to detect security violation, such as malware detection. While the existing approaches provide effective solutions in detecting and defending adversarial samples, they fail to detect them when they are encrypted. In this study, a novel framework is proposed which employs statistical test to detect adversarial instances, when data under analysis are encrypted. An experimental evaluation of our approach shows its practical feasibility in terms of computation cost.

**Keywords:** Privacy · Adversarial machine learning · Homomorphic encryption

## 1 Introduction

Machine learning algorithms are generally constructed under the assumption that models are trained on instances drawn from a distribution expected to be the representative of test instances exploited for making the prediction. In an ideal scenario, the training and test distributions are identical. However, this hypothesis does not hold in the presence of adversaries. In a real scenario, every learning-based system which is trained and employed over economic, political, military, and security-critical data, is in the certain risk of attracting adversaries who gain advantages by manipulating the system to influence its decisions [1]. Such activities include, but are not limited to, Spam detection [2], terrorist Tweet analysis, adversarial advertisements, malware PDF file detection [3], and sign detection in autonomous vehicles [4].

The problem of adversarial machine learning has attracted considerable attention since 2014, when Szegedy et al. [5] showed that deep convolutional


S. Mauw and M. Conti (Eds.): STM 2019, LNCS 11738, pp. 71–88, 2019.
https://doi.org/10.1007/978-3-030-31511-5_5

neural network, utilized for object recognition, can be fooled by perturbed input image which is visually indistinguishable. From then on a lot of work has been devoted to this field, spanning from physical consequences of adversarial instances presented in autonomous vehicles [6], the analysis of classifiers' robustness against adversarial perturbation [7], to defenses designed to mitigate issues caused by adversarial instances [8]. Still, the majority of proposed defenses, e.g. defensive distillation [9], are not effective in adapting to changes in attack strategies.

To provide an arm race that is independent of the kind of attacks, Grosse et al. [10] proposed an approach based on the intuition that adversarial instances must inherently show some *statistical* differences with the correct data. More precisely, an attacker generally designs adversarial instances in such a way that it is similar to the training records labeled as he expects. These new fake elements – independently from how they have been created – must have different distributions compared to the training data. Grosse et al. [10] showed that *statistical tests* work efficiently in detecting adversarial instances, even when these instances have been generated through different adversarial instance crafting techniques.

However, their proposed approach fails to detect adversarial instances when they are crafted in an *encrypted* format. Generally, the primary organizations who determine and mandate laws about the way sensitive data may be utilized, such as European General Data Protection Regulation (GDPR)[1], Payment Card Industry Data Security Standard (PCI DSS)[2], and the Health Insurance Portability and Accountability Act (HIPAA)[3], permit the analysis of data only when they are encrypted. Encryption mechanisms, which are defined based on rigorous mathematical rules, provide the possibility of confirming security at every step they are employed [11].

In this study, as a very first work in addressing issues caused by adversarial samples in private setting, we propose encryption-based protocols, which enable the system to detect adversarial instances when they are encrypted. The proposed mechanism *securely* performs a statistical test on encrypted data to measure the distribution difference between two datasets. In the case that the difference is high, the crafted instances are suspicious of being designed by an adversary.

The contribution of the current study can be summarized as follows:

- We propose a novel framework that can be deployed as a tool to *securely* detect adversarial instances in private settings.
- We propose a mechanism for transforming a non-integer-based statistical test into an integer-based one.
- We propose a new protocol for computing the exponential function. The security proof for this protocol is provided.

[1] https://www.eugdpr.org/.
[2] https://www.pcisecuritystandards.org/.
[3] https://www.hhs.gov/hipaa.

– We report the computation cost of our protocol for different values of adversarial instances, number of features, and size of training data.
– Finally, we prove the security of our architecture.

The remainder of this paper is structured as follows. The next section presents the preliminary notations utilized in the current study. In Sect. 3, the motivating example and the architecture of our approach are proposed. In Sect. 4 the integer-based representation of statistical test is reported. The security proof of our architecture is presented in Sect. 5. We analyze the performance of our protocols in Sect. 6, and in Sect. 7 we discuss the achievements and shortcomings of the proposed framework. In Sect. 8 the related work is presented. Finally, Sect. 9 concludes by briefly proposing future research directions.

## 2 Preliminaries

This section provide the background knowledge used in this work, including statistical detection of adversarial instances and Homomorphic encryption.

### 2.1 Statistical Detection of Adversarial Instances

To learn a classifier from training data, the real distribution of features $D_{real}^{C_i}$ for each subset $C_i$ corresponding to a class $i$ must be extracted. These subsets define a partition of the training data. Due to the limited number of training instances, each machine learning algorithm only learns an approximation of this real distribution, say learned feature distribution $D_{train}^{C_i}$. The existence of adversarial instances is a manifestation of the difference between $D_{real}^{C_i}$ and $D_{train}^{C_i}$. In this way, an adversary finds a sample from $D_{real}^{C_i}$ that does not adhere to $D_{train}^{C_i}$. Generally, an adversary has no knowledge about $D_{real}^{C_i}$, thus, the existing algorithms for generating adversarial instances perturb the legitimate instances drawn from $D_{train}^{C_i}$.

Independently of how adversarial instances have been generated, all adversarial instances for a class $C_i$ will constitute a new distribution $D_{adv}^{C_i}$ of this class. This means that $D_{adv}^{C_i}$ is consistent with $D_{real}^{C_i}$, because each adversarial instance for a class $C_i$ is still a data point that belongs to this class. However, for adversarial instances we have $D_{adv}^{C_i} \neq D_{train}^{C_i}$.

Following this argument, *statistical tests* are a natural candidate for adversarial instance detection [10]. The intuition is that adversarial instances have to be inherently distributed differently from legitimate instances during training.

**Maximum Mean Discrepancy (MMD) Test:** The *Maximum Mean Discrepancy (MMD)*, as a well-known two-sample statistical test, is defined in terms of particular function spaces that witnesses the difference between distributions through kernel function [12]. Formally, for two distributions $\mathcal{X} = \{X_1, \ldots, X_m\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_n\}$, the amount of $MMD$ is computed as the following:

$$MMD(\mathcal{X},\mathcal{Y}) = (\frac{1}{m^2}\sum_{i,j=1}^{m}\kappa(X_i,X_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n}\kappa(X_i,Y_j) + \frac{1}{n^2}\sum_{i,j=1}^{n}\kappa(Y_i,Y_j))^{\frac{1}{2}} \tag{1}$$

where $\kappa(X,Y) = \exp(-\frac{\|X-Y\|^2}{2\sigma^2})$ represents the *Gaussian Kernel* function, in which $\sigma$ is generally (as in this study) considered to be 1 [12]. Trivially, if two distributions are exactly equal in an ideal scenario, then $MMD(\mathcal{X},\mathcal{Y}) = 0$. However, a threshold, say $\alpha$, can be specified by an expert such that if $MMD(\mathcal{X},\mathcal{Y}) \leq \alpha$, it is then said that the two distributions are close *enough*. It should be noted that fixing this threshold is out of the scope of the current study.

### 2.2 Homomorphic Encryption

We define our secure computation protocols based on a semantically secure homomorphic cryptosystem, named *Paillier* cryptosystem [13]. This scheme preserves a certain structure that can be employed to process ciphertexts without decryption. Given $\mathscr{E}_{pk}(m_1)$ and $\mathscr{E}_{pk}(m_2)$, a new ciphertext whose decryption yields the sum of the plaintext messages $m_1$ and $m_2$ can be obtained by performing a multiplication operation over ciphertexts under the additively homomorphic encryption scheme:

$$\mathcal{D}_{sk}(\mathscr{E}_{pk}(m_1) \otimes \mathscr{E}_{pk}(m_2)) = m_1 + m_2.$$

Moreover, exponentiation of any ciphertext with a public key yields the encrypted product of the original plaintext and the exponent as: $\mathcal{D}_{sk}(\mathscr{E}_{pk}(m)^e) = e \cdot m$.

In the rest of this study, we denote the ciphertext of a message $m$ by $[m]$. In what follows, we present two additive homomorphic-based protocols, named secure *comparison* [14] and *multiplication* [15] protocols, which serve as building blocks in our framework.

*Secure Comparison Protocol:* We use a secure comparison protocol (e.g., [14]) to compare two encrypted values. Given two ciphertexts $[a]$ and $[b]$, the secure comparison between $[a]$ and $[b]$ is defined as follows:

$$SecureComp([a],[b]) = \begin{cases} [1] & \text{if } a \leq b, \\ [0] & \text{otherwise.} \end{cases}$$

*Secure Multiplication Protocol:* We use a secure multiplication protocol (e.g., [15]) to compute the multiplication between two encrypted values. Given two ciphertexts $[a]$ and $[b]$, the secure multiplication of $[a]$ and $[b]$ is defined as $Mult([a],[b]) = [a \cdot b]$.

## 3 Framework

This section presents the challenges in detecting adversarial instances over encrypted data through a running example within spam detection, and then an overview of our framework and security assumption is presented to address the associated challenges.

### 3.1 Motivating Example

Spam messages cause several problems, spanning from direct financial losses to misuses of Internet traffic, storage space and computational power. Spam emails are also becoming a tool to perpetrate different cyber-crimes, such as phishing, malware distribution, social engineering fraud, or propaganda distribution (e.g. by terrorist group). To reach his goal, the spammer generally hides himself behind infected devices (botnets) which send millions of spam messages with similar text and template (spam campaign), against their users' will (or even awareness) at the spammer command. Identifying the devices which are part of a botnet and consequent removal of malicious code from the device, helps in strongly limiting the amount of generated spam traffic [16]. Along with, in some dangerous scenarios, e.g. distributing terrorist messages, cyber-criminal police is able to catch the spammer through a thorough analysis of spam campaign. However, such an analysis brings several privacy implications resulting from this fact that data analyzer (cyber crime police) should have access to all outgoing users' emails. To mitigate this issue, it is essential that the email server be able to protect the confidential content of users' emails, while at the same time the data analyzer remains still capable in detecting dangerous spammers.

*Our Solution:* Homomorphic encryption serves as a privacy preserving technique which enables data analyzer to perform some desired operations over protected data, without needing them to be decrypted. Thus, email server homomorphically can encrypt a set of emails, belonging to a suspicious user, and send them to (semi-trusted) data server. Cyber crime police also provides two separate collections of records representing benign and spam messages. It also sends them encrypted to the data analyzer. Without decrypting any email, the data server– as an expert component in analyzing encrypted data– is capable to detect if data belonging to a suspicious user shows *considerable* difference compared to benign records sent by police.

### 3.2 Architecture and Workflow

To detect adversarial instances over encrypted data, we employ an interactive privacy model in which two additional components (plus the data analyzer and data provider) are needed to securely perform analysis. More precisely, this privacy model comprises four main components:

– *Data-analyzer* who is interested in detecting adversarial instances.

– *Data provider* who provides a dataset suspicious to be designed by a potential adversary
– *Semi Trusted Party (STP)* is a semi-honest component that generates public ($p_k$) and private ($s_k$) keys. This component is assumed to have limited storage and computation capabilities.
– *Data Server (DS)* is a remote component, generally in the cloud with high storage capability that stores encrypted data. DS is controlled by an expert who performs the analysis on encrypted data through secure communication with *STP*.

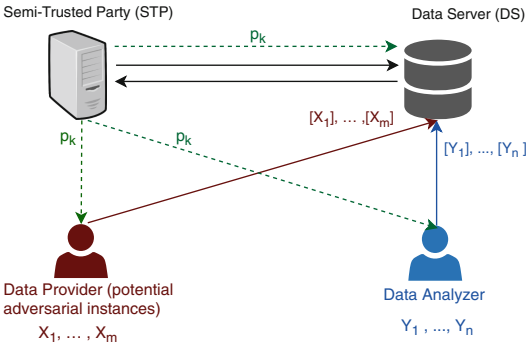


**Fig. 1.** Reference architecture

Figure 1 shows the architecture underlying the interactive privacy model along with the main components and their interactions. First, *STP* generates public ($p_k$) and private ($s_k$) keys; it sends the public key $p_k$ to data analyzer, data provider, and to *DS*. After receiving the public-key, data analyzer and data provider encrypt their data and send the associated set of encrypted vectors, denoted respectively as $[X_1], \ldots, [X_n]$ and $[Y_1], \ldots, [Y_m]$, to *DS*. From now on, the secure computation protocols are performed between *STP* and *DS*.

We assume a semi-honest security model, where all participants are honest-but-curious. This means that all components follow the protocols properly, but they are interested in learning the input of other parties. In our motivating example, email server and cyber-criminal police can be considered as data provider and data analyzer, respectively. Data server and Semi-trusted Party are two external components expert in performing analysis over encrypted data. It is noticeable that in this specific scenario, the data provider and STP can be one unique component (the same for data server and data analyzer).

## 4 Private Detection of Adversarial Instances

This section transforms non-integer-based MMD statistical test to the integer-based formula; it then presents secure computation protocols to compute it over encrypted input.

### 4.1 Integer-Based MMD Evaluation

Considering that homomorphic encryption techniques are only applicable on integer numbers, and Maximum Mean Distance (Relation 1) is not defined based on integer numbers, in what follows we propose a methodology to evaluate MMD condition through an integer-based formula.

To this end, let's suppose that two datasets $D_1$ and $D_2$, and their associated encrypted versions as $D'_1$ and $D'_2$, respectively, are given. Let us denote the MMD distance of $D_1$ and $D_2$ as $MMD(D_1, D_2)$, and the (decrypted) MMD distance of encrypted formats as $MMD'(D'_1, D'_2)$. It is expected that if the MMD distance of two dataset as plaintexts is higher than $\alpha$, then the MMD distance of equivalent ciphertexts also be higher than $\alpha$, or equivalently, if $MMD'(D'_1, D'_2) \leq \alpha$ then $MMD(D_1, D_2) \leq \alpha$.

We first consider that $x_i$'s are integer values. At the end of this section we will explain if this condition does not hold, how the problem can be addressed as well.

Given this assumption, the reason that Relation 1 may return a non-integer outcome is resulted from *Gaussian Kernel* function defined as $\kappa(X, Y) = e^{\frac{-1}{2}\|X-Y\|^2}$. In what follows, we transform it to integer-based relation.

By approximating the irrational value $e^{-\frac{1}{2}}$ with a rational value $d$ (by rounding it to its $t$'th decimal number), we obtain $d \leq e^{-\frac{1}{2}} \leq (d+\delta)$, where $\delta = 10^{-t}$. Therefore, by denoting the squared Euclidean distance of two vectors $X$ and $Y$ as $n_{XY} = \|X - Y\|^2$, we have: $d^{n_{XY}} \leq \kappa(X, Y) \leq (d+\delta)^{n_{XY}}$.

Now, we are looking for $\alpha'$ such that the satisfaction of the following relation, results in the satisfaction of Relation 1.

$$n^2 \sum_{i,j=1}^{m} \frac{(d \cdot 10^t)^{n_{X_iX_j}}}{(10^t)^{n_{X_iX_j}}} - 2mn \sum_{i,j=1}^{m,n} \frac{(d \cdot 10^t)^{n_{X_iY_j}}}{(10^t)^{n_{X_iY_j}}} + m^2 \sum_{i,j=1}^{n} \frac{(d \cdot 10^t)^{n_{Y_iY_j}}}{(10^t)^{n_{Y_iY_j}}} \leq \alpha'$$

**Theorem 1.** *By setting* $\alpha' = \sqrt{\alpha^2 - 2d\delta}$ *(for negligible* $\delta$*), the satisfaction of* $MMD'(D'_1, D'_2) \leq \alpha' \leq \alpha$ *results in* $MMD(D_1, D_2) \leq \alpha$.

*Proof.* Proof in Appendix. □

However, considering that additive homomorphic encryption does not provide secure division protocol, we multiply both sides by the common denominator of all fractions, i.e. $\tau = (10^t)^{\sum_{i,j=1}^{m} n_{X_iX_j} + \sum_{i,j=1}^{m,n} n_{X_iY_j} + \sum_{i,j=1}^{n} n_{Y_iY_j}}$:

$$n^2 \sum_{i,j=1}^{m} (d \cdot 10^t)^{n_{X_iX_j}} \tau_{X_iX_j} - 2mn \sum_{i,j=1}^{m,n} (d \cdot 10^t)^{n_{X_iY_j}} \tau_{X_iY_j} + m^2 \sum_{i,j=1}^{n} (d \cdot 10^t)^{n_{Y_iY_j}} \tau_{Y_iY_j} \leq \tau\alpha' = \alpha''$$

where for all $X, Y \in \mathcal{X}, \mathcal{Y}$, we have $\tau_{XY} = \frac{\tau}{(10^t)^{\|X-Y\|^2}}$, and since the phrase appeared in the denominators of the fractions, exists also inside the numerator as well, the outputs of this phrase is integer.

It is noticeable that although $\alpha$ is generally a decimal number (approximately 0.05), but since $\tau$ is a very big number, $\alpha'' = \tau\alpha'$ will be an integer at the end.

Thus, in the last relation all variables are integers. It also needs to be mentioned that we supposed $x_i$s are integer numbers. If this hypothesis does not hold, it is just enough to round them to $t'$'th decimal number and then multiply them by $10^{t'}$. The value of $t'$ should be set beforehand from the context.

### 4.2 Preliminary Protocols

This section presents HE-based protocols, named *secure scalar, Euclidean distance*, and *exponential protocol*, which serve as preliminary protocols for evaluating the satisfaction of integer-based MMD distance.

*Secure Scalar Product: Bob* owns two vectors of encrypted values as $[X] = ([x_1], \ldots, [x_k])$, $[Y] = ([y_1], \ldots, [y_k])$, and he is interested in obtaining the scalar product of $[X]$ and $[Y]$ which equals to $\left[\sum_{i=1}^{k} x_i \cdot y_i\right]$. We propose the following formula to compute scalar product securely through the application of secure multiplication protocol:

$$Scalar([X], [Y]) = \left[\sum_{i=1}^{k} x_i \cdot y_i\right] = \prod_{i=1}^{k} Mult([x_i], [y_i])$$

*Secure Euclidean Distance: Bob* owns two vectors of encrypted values as $[X] = ([x_1], \ldots, [x_k])$, $[Y] = ([y_1], \ldots, [y_k])$, and he is interested in finding the (squared) Euclidean distance of these vectors in encrypted format. We remind that the (squared of) Euclidean distance of two vectors $X$ and $Y$ is equivalent to $X \cdot X - 2X \cdot Y + Y \cdot Y$, where "$\cdot$"refers to the scalar product of two vectors. The (squared) Euclidean distance of two encrypted vectors can be computed as the following:

$$Dist([X], [Y]) = Scalar([X], [X]) \cdot (Scalar([X], [Y]))^{-2} \cdot Scalar([Y], [Y])$$

*Secure Exponential Protocol:* Suppose *Bob* owns encrypted number $[b]$, and $a$ is a public integer number. *Bob* is interested in finding $[a^b]$. We propose in Algorithm 1 a new protocol for secure computation of $[a^b]$. Our proposed procedure is based on masking $b$ with $(\kappa + \ell)$-bit random integer value $r$, and afterwards secure multiplication protocol is applied to remove the noise.

### 4.3 Secure MMD Distance Computation

To detect adversarial instances over encrypted values, we design MMD protocol according to the integer-based relation presented in Sect. 4.1 applying HE-based building blocks. It is supposed that $DS$ owns two sets of encrypted vectors, coming from potential adversary and data analyzer as $\mathcal{X} = \{[X_1], \ldots, [X_m]\}$ and $\mathcal{Y} = \{[Y_1], \ldots, [Y_n]\}$, respectively. The values $m, n, t$ and $\alpha$ are known by $DS$. The following steps are executed between $DS$ and $STP$:

**Algorithm 1.** *Exp*$(a, [b])$*:* Secure Exponential Function.

**Data**: *Alice* generates encryption keys. $a$ is a public integer number. *Bob* owns encrypted integer numbers $[b]$.
**Result**: *Bob* obtains $[a^b]$.
1 *Bob* additively masks $[b]$ with $r$ and sends $[b'] = [b + r] = [b] \cdot [r]$ to *Alice*.
2 *Alice* decrypts $[b']$ and sends $[a^{b'}]$ to *Bob*.
3 *Bob* obtains $[a^b] = Mult([a^{b'}], [a^{-r}])$.

– For $1 \leq i, r \leq m$ and $1 \leq j, t \leq n$, $DS$ first obtains the encrypted values $Dist([X_i], [X_r])$, $Dist([X_i], [Y_j])$, and $Dist([Y_j], [Y_t])$ through secure Euclidean distance protocol. It then locally computes:

$$[Z] = \prod_{i,r,j,t} Dist([X_i], [X_r]) \cdot Dist([X_i], [Y_j]) \cdot Dist([Y_j], [Y_t])$$

– After obtaining $[Z]$, for $1 \leq i, r \leq m$ and $1 \leq j, t \leq n$, $DS$ computes the following encrypted values through secure communication with $STP$:

$$\begin{aligned}
[\tau_{X_i X_r}] &= Exp(10^t, ([Z] \cdot (Dist([X_i], [X_r])^{-1}))) \\
[\tau_{X_i Y_j}] &= Exp(10^t, ([Z] \cdot (Dist([X_i], [Y_j])^{-1}))) \\
[\tau_{Y_j Y_t}] &= Exp(10^t, ([Z] \cdot (Dist([Y_j], [Y_t]))^{-1}))
\end{aligned}$$

At this step, by setting $a = d \cdot 10^t$, $DS$ computes the encrypted value of MMD distance through secure communication with $STP$, as the following:

$$\begin{aligned}
[MMD] = \prod_{i,r,j,t} & (Mult(Exp(a, Dist([X_i], [X_r])), [\tau_{X_i X_r}]))^{n^2} \\
& \cdot (Mult(Exp(a, Dist([X_i], [Y_j])), [\tau_{X_i Y_j}]))^{m \cdot n} \\
& \cdot (Mult(Exp(a, Dist([Y_j], [Y_t])), [\tau_{Y_j Y_t}]))^{m^2}
\end{aligned}$$

– Finally, secure comparison protocol is applied as $SecureComp$ $([MMD], [\alpha''])$, where the outcome $[0]$ means that the data provider is suspicious to be an adversary.

## 5 Security Analysis

In this section, we present a security sketch of the proposed privacy preserving protocols in the semi-honest model, where parties are assumed to be honest in following the protocol description, while they are curious to obtain more information than they are entitled to.

Based on this assumption, we provide proofs to show that our secure Maximum Mean Discrepancy (MMD) protocol is simulation secure in the semi-honest security model. By providing the simulation security, the probability that an

adversary can learn private information from truly generated data by the parties in our protocols is at most negligibly more than the probability that an adversary can learn from given randomly generated data. We use the simulatability paradigm [17] in our proofs, where the adversary takes the control of the network and try to obtain the final result of the protocol by itself as the only party in the protocol. In this paradigm, security is defined as a comparison of computation work-flow in "real world" and "ideal world". In real world, a protocol can be broken into sub-protocols or computations that are carried out by each party throughout the protocol. Let us denote $\pi$ as the MMD protocols; we can split $\pi$ into two parts: $\pi = \pi_{DS}$ and $\pi_{STP}$, which are performed in parties $DS$ and $STP$, respectively. $\pi_{DS}$ takes $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}}$, $\gamma'$, and $\alpha'$, which are the inputs, and outputs 1/0 (let's call this $\vartheta$), $[\vartheta] \leftarrow \pi_{DS}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}, \gamma', \alpha')$. $\pi_{STP}$ decrypts the given encryptions from $DS$, processes them, and sends their encrypted versions back to $DS$. Thus, to perform MMD the encrypted messages flow from one party to another party and together they generate the $[\vartheta]$ as the result of MMD. Assuming $DS$ is corrupted by an adversary $\mathcal{A}$, then $\mathcal{A}$ has access to his inputs, and $[\vartheta]$. Similarly, when $STP$ is corrupted, the adversary has access to the intermediate computation results.

In an ideal world, it is assumed that one of the parties is corrupted by an adversary. Then, he uses a simulator to generate the outputs of the other party. This would be similar to performing MMD with just one corrupted party. In the ideal world, an adversary $\acute{\mathcal{A}}$, who has control over $DS$, has only access to his inputs and the garbage inputs given from simulated $STP$ instead of the correct result of $\pi_{STP}$. The goal is to show that $\mathcal{A}$ can learn equal or negligibly more than $\acute{\mathcal{A}}$, meaning that they are computationally indistinguishable, then we can claim that MMD is a simulation secure protocol.

**Definition 1.** *Let $a \in \{0,1\}^*$ represents the parties' inputs, $n \in \mathbb{N}$ to be a security parameter, and $X = \{X(a,n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$ and $Y = \{Y(a,n)\}_{a\in\{0,1\}^*;n\in\mathbb{N}}$, two infinite sequences of random variables, are probability ensembles. Then, $X$ and $Y$ are computationally indistinguishable, denoted as $X \overset{c}{\equiv} Y$, if there is a polynomial $p(.)$ for every non-uniform polynomial-time probabilistic algorithm (nuPPT) $D$ such that:*

$$|Pr[D(X(a,n)) = 1] - Pr[D(Y(a,n)) = 1]| < 1/p(n) \tag{2}$$

The *Mult*, *SecureComp*, and *Equality* sub-protocols are proved to be secure in the same security setting [14,15], and [18], respectively. Moreover, since the *Scalar* and *Dist* sub-protocols are both built by only using *Mult*, we can claim that they are also simulation secure.

### 5.1 Security of SecureExp

Let denote the computation of $b'$ as $DS_{f_1}$, $[a^{b'}]$ as $STP_{f_1}$, and $[a^b]$ as $DS_{f_2}$. Then, we have $DS_f = (A_{f_1}, A_{f_2})$, $STP_f = (B_{f_1})$, and $f = (DS_f, STP_f)$.

**Theorem 2.** *The protocol SecureExp is simulation secure and securely computes the functionality $f$, when the $DS$ is corrupted by adversary $\mathcal{A}$ in the presence of semi-honest adversaries.*

*Proof.* We need to show that $DS$ cannot computationally distinguish between generated messages and outputs from $\mathcal{S}_2$ that is the simulation of $STP$, and randomly generated data. $DS$ receives two outputs from $\mathcal{S}_2$, $[a^{b'}]$ and result of $Mult$ sub-protocol. Given $a$, $[b]$, and $1^n$ (security parameter), $DS$ works as follow:

1. $DS$ chooses uniformly distributed random number $r$;
2. $DS$ executes $DS_{f_1}$ to obtain $[b']$, and sends it to $\mathcal{S}_2$;
3. $\mathcal{S}_2$ chooses a random number $R$, encrypts it and sends $[R]$ to $DS$.

The output of simulation can be written as: $Sim_{DS}(1^n, a, [b], DS_f, f) = (a, [b], r; [c]; [\acute{c}]; \phi))$. The real view of $DS$ can be presented as $view^f_{DS}(a, [b]) = (a, [b], r; [a^{b'}])$. And the output of the real view is $output^f(a, [b]) = ([a^b], \phi)$. It can be observed that $DS$ cannot computationally distinguish between $[a^{b'}]$ and $[c]$, since the underlying encryption scheme is semantically secure. Note that the $Mult$ sub-protocol is already proven secure in [15]. Therefore, we can claim that:

$$Sim_{DS}(1^n, a, [b], A_f, f) \stackrel{c}{\equiv} \{view^f_{DS}(a, [b]; \phi), output^f(a, [b]; \phi)\}$$

□

**Theorem 3.** *The protocol SecureExp is simulation secure and securely computes the functionality $f$, when the $STP$ is corrupted by adversary $\mathcal{A}$ in the presence of semi-honest adversaries.*

*Proof.* $STP$ works as follow:

1. $\mathcal{S}_1$ chooses a $\kappa + \ell + 1$-bit random number $r$, encrypts it, and sends $[r]$ to $STP$.
2. $STP$ executes $STP_{f_1}$ and sends the result back to $\mathcal{S}_1$.

Although $STP$ has the decryption key, it cannot distinguish between $r$ and $b$, since $b$ is masked with a $(\kappa + \ell)$-bit integer. Therefore,we can claim that:

$$Sim_S TP(1^n, \phi, STP_{f_1}, f) = \{view^f_S TP(a, [b], \phi, n), output^f(a, [b]; \phi)\}$$

□

Since the $SecureExp$ sub-protocol is proven to be secure, showing that simulation security of $MMD$ protocol is straightforward. $MMD$ protocol uses $Dist$, $Exp$, and $Mult$ sub-protocols, which all have been proven to be simulation secure; therefor, we can claim that $MMD$ protocol is also simulation secure.

## 6 Performance Analysis

To compute the computational complexity of MMD protocol, let us assume that potential adversary and data analyzer uploaded $m$ and $n$ records on DS's server. Moreover, each record is a vector of size $k$, and each component of a vector is expressed maximum in $\ell$-bit length. Based on $MMD$ distance protocol, the number of times that building block protocols are employed equals to:

$$\mathcal{N}(m,n) = (m^2 + m \cdot n + n^2)(Scalar + Dist + Add + SExp. + SExp. + Mult) \tag{3}$$

Our building block protocols require primitive operations addition, encryption, decryption, and exponential, as the following:

- *Multiplication protocol* (*Mult*) requires 5, 4, 1, and $2_{\kappa+\ell}$ addition, encryption, decryption, and exponential, respectively.
- *Scalar product* (*Scalar*) needs 6k, 4k, k, and $2k_{\kappa+\ell}$ addition, encryption, decryption, and exponential, respectively.
- *Exponential protocol* (*SExp.*) employs 6, 7, 2, and $2_{\kappa+\ell}$ addition, encryption, decryption, and exponential, respectively.
- *Distance protocol* (*Dist.*) requires 2 and $1_{\kappa+\ell}$ addition and exponential, respectively.

Given the above argument, the number of times that Relation 3 employs addition, encryption, decryption, and exponential operations can be approximated as follows:

$$\begin{aligned} &\mathcal{N}(m,n,k,\ell) = (m^2 + m \cdot n + n^2) \\ &((6k+21)Add. + (4k+20)Enc. + (k+6)Dec. + (2k+6)Exp.) + \ell(Add. + Enc. + Dec.) \end{aligned} \tag{4}$$

We implemented addition, encryption, decryption, and exponential protocols using $C++$ on a single machine running Ubuntu 14.04 LTS with 64-bit microprocessor and 8 GB of RAM. The cryptographic key length of Paillier is selected as NIST standard as 4096 bits. In our implementation, addition, encryption, and decryption for $10^6$ records required 8.3, 5.6, and 9 s, respectively. Moreover, by considering $\kappa = 112$, each exponential operation needs 200 additions for element with $\ell$-bit length equal to 20.

To assess the practical feasibility of our mechanism, we performed a number of experiments. Grosse et al. [10] showed that 50 adversarial instances are enough to infer a considerable MMD statistical distance between two datasets. Moreover, according to standard dimensioning technique, proposed in [19], the minimum size for a dataset to produce a reliable result is to dimension it as six times to the number of features. Therefore, to get a better insight on computation cost of the proposed approach, we consider the number of features to get their values as $k \in \{20, 50, 100, 200\}$, while the number of data in training set ($m$) is set to six times of $k$. The number of adversarial instances ($n$) varies from 50 and gets its value as $n \in \{50, 60, 70, 80, 90\}$.

Figures 2 and 3 show the computation costs (in log-scale) for different values of $n$ and $k$, respectively. As explained previously, $m$ is considered as a dependent variable to the number of features ($m = 6k$). From Fig. 2, it can be inferred that for fixed values of $k$ and $m$, the required runtime increases linearly (with a slight slop) when $n$ increases. On the other hand, for fixed value of $n$, when $k$ varies from 20 to 200, the runtime increases from 0.5 h to 288 h. Figure 3 confirms that $k$ has considerable impact on the computation cost. This result put the light on the fact that application of appropriate feature selection technique, prior to the adversarial instance detection over encrypted data, can noticeably reduce the computation cost.

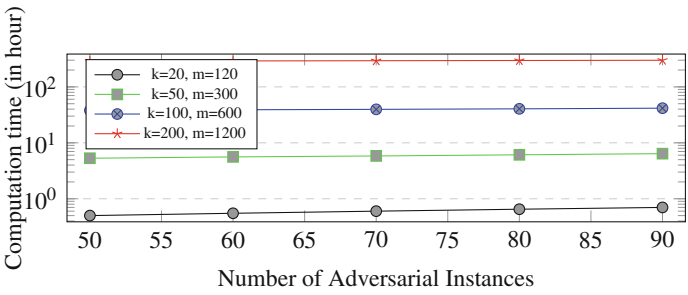


**Fig. 2.** Computation time (in hour) for different values of adversarial instances (n).

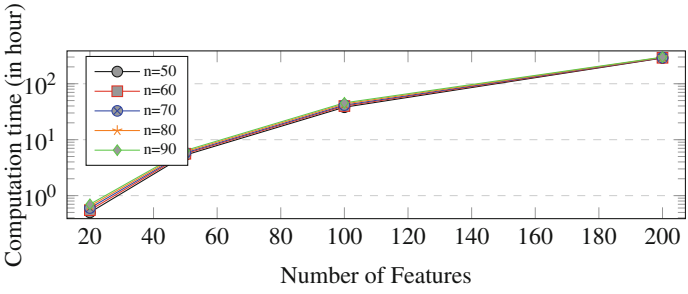


**Fig. 3.** Computation time (in hour) for different number of features (k).

## 7 Discussion

This section discusses some noticeable notions about the current study.

- We adapted the results of work done by Grosse et al. [10], which through mathematical explanation and experimental analysis proved the effectiveness of MMD statistical test in detecting adversarial samples. Therefore, we designed our experimental analysis not on detecting adversarial instances over encrypted data, but on evaluating the feasibility and efficiency of the proposed mechanism.
- Gaussian kernel is a de facto kernel which is employed in several data analysis approaches, e.g. image/signal processing, computational chemistry, SVM classifier, etc. Accordingly, the result of current study can be deployed in the aforementioned studies when analysis is desired to be performed over distributed encrypted data.
- The performance analysis of our approach shows that the proposed mechanism is effectively feasible when the number of features reduces. This result suggests that the application of an appropriate feature reduction technique considerably reduces the computation cost of our mechanism.
- Due to the fact that the current approach requires at least 50 adversarial samples from the attacker to be able to detect the adversarial instances, as one future work we plan to construct a robust classifier with one outlier class over encrypted data. The robust classifier is able to detect an adversarial instance upon being received [10].

## 8 Related Work

In this section we present works in the literature related to Adversarial Machine Learning and Encryption-based Mechanisms.

**Adversarial Machine Learning:** A growing body of work has been devoted to the field of adversarial machine learning, trying to solve the problem from different perspectives. A large number of work has been done (i) to develop attacks against machine learning, both at training time (poisoning attacks) [20], and at test time (evasion attack) [1], (ii) to design systematic methodologies for evaluation of the robustness of machine learning algorithms against such kinds of attacks [21], and (iii) to propose appropriate defense mechanisms for mitigating these threats [22]. However, there is no work in the literature which studies adversarial machine learning issues when the data under analysis are encrypted.

**Encryption-Based Mechanisms:** The main idea in encryption-based approaches is to obfuscate the privacy-sensitive data prior to processing. Cryptography-based techniques have been deployed in several domains of data analysis. In [23] the possible scenarios of applying homomorphic encryption on medical data is discussed. A working implementation of a prediction service in the cloud which takes private encrypted health data and returns the probability for suffering cardiovascular disease is returned in encrypted format. Erkin et al. in [15] propose a privacy-enhanced face recognition system, which allows to efficiently hide both biometrics and the result from the server for matching operation. Cryptography-based approaches have also been widely utilized in

constructing data mining algorithms collaboratively, e.g. constructing on whole encrypted data a clustering algorithm [24], or a classifier [25], and collaborative private feature selection [26]. These techniques also have been used in other scenarios, *e.g.* private text analysis [27], the general framework for privacy preserving distributed data analysis [28], etc. However, to the best of our knowledge the problem of adversarial machine learning has not been addressed in private setting.

## 9 Conclusion

This paper presents a framework for detecting adversarial instances which are crafted through encrypted format. To this end, we employed statistical test which measures the distance of two encrypted datasets' distribution. Due to the fact that the proposed approach is based on homomorphic encryption, we proposed a mechanism to transform non-integer statistical test to an integer-based one. We showed the practical feasibility of the proposed approach in terms of computation cost.

In future research, we plan to address other challenges in adversarial machine learning, e.g. constructing robust classifier, over encrypted distributed data. We also plan to perform other statistical tests, e.g. *energy distance*, on encrypted datasets, and compare their effectiveness and efficiency. Moreover, we are interested in applying efficient secure multi-party computation, e.g. *data packing*, to speed up the process when size of data increases. We also plan to evaluate the effect of feature selection techniques on accuracy and efficiency of our methodology.

**Acknowledgment.** This work was partially supported by the H2020 EU funded project SECREDAS [GA #783119] and by the H2020 EU funded project C3ISP [GA #700294].

## Appendix

In what follows we prove Theorem 1, claiming that if we set $\alpha' = \sqrt{\alpha^2 - 2d\delta}$ (for negligible $\delta$), then from $MMD'(D_1', D_2') \leq \alpha'$ we can conclude that $MMD(D_1, D_2) \leq \alpha$.

Basically, we are looking for $\alpha'$ such that if the following relation holds:

$$n^2 \sum_{i,j=1}^{m} d^{n X_i X_j} - 2mn \sum_{i,j=1}^{m,n} d^{n X_i Y_j} + m^2 \sum_{i,j=1}^{n} d^{n Y_i Y_j} \leq m^2 n^2 {\alpha'}^2$$

We can conclude that:

$$n^2 \sum_{i,j=1}^{m} \kappa(X_i, X_j) - 2mn \sum_{i,j=1}^{m,n} \kappa(X_i, Y_j) + m^2 \sum_{i,j=1}^{n} \kappa(Y_i, Y_j) \leq m^2 n^2 \alpha^2$$

To this end, we first find a relation between two above relations:

$$
\begin{aligned}
& n^2 \sum_{i,j=1}^{m} \kappa(X_i, X_j) - 2mn \sum_{i,j=1}^{m,n} \kappa(X_i, Y_j) + m^2 \sum_{i,j=1}^{n} \kappa(Y_i, Y_j) \\
\leq & n^2 \sum_{i,j=1}^{m} (d+\delta)^{n_{X_i X_j}} - 2mn \sum_{i,j=1}^{m,n} d^{n_{X_i Y_j}} + m^2 \sum_{i,j=1}^{n} (d+\delta)^{n_{Y_i Y_j}} \\
= & n^2 \Big( \sum_{i,j=1}^{m} d^{n_{X_i X_j}} + \sum_{i,j=1}^{m} \Big[ \frac{n_{X_i X_j}(n_{X_i X_j}-1)}{2} d^{n_{X_i X_j}-1}\delta + \ldots \Big] \Big) - 2mn \sum_{i,j=1}^{m,n} d^{n_{X_i Y_j}} \\
& + m^2 \Big( \sum_{i,j=1}^{n} d^{n_{Y_i Y_j}} + \sum_{i,j=1}^{n} \Big[ \frac{n_{Y_i Y_j}(n_{Y_i Y_j}-1)}{2} d^{n_{Y_i Y_j}-1}\delta + \ldots \Big] \Big)
\end{aligned}
$$

From the application of *binomial* theorem, we obtain:

$$
\begin{aligned}
& (n^2 \sum_{i,j=1}^{m} d^{n_{X_i X_j}} - 2mn \sum_{i,j=1}^{m,n} d^{n_{X_i Y_j}} + m^2 \sum_{i,j=1}^{n} d^{n_{Y_i Y_j}}) + (n^2 \sum_{i,j=1}^{m} [\frac{n_{X_i X_j}(n_{X_i X_j}-1)}{2} d^{n_{X_i X_j}-1}\delta + \ldots] \\
& + m^2 \sum_{i,j=1}^{n} [\frac{n_{Y_i Y_j}(n_{Y_i Y_j}-1)}{2} d^{n_{Y_i Y_j}-1}\delta + \ldots]) \leq m^2 n^2 \alpha^2
\end{aligned}
$$

This means that it is enough to set $\alpha'^2 = \alpha^2 - 2\delta d$, because:

$$
\begin{aligned}
\Rightarrow \alpha' &= m^2 n^2 \alpha^2 - (n^2 \sum_{i,j=1}^{m} [\frac{n_{X_i X_j}(n_{X_i X_j}-1)}{2} d^{n_{X_i X_j}-1}\delta + \ldots] \\
&\quad + m^2 \sum_{i,j=1}^{n} [\frac{n_{Y_i Y_j}(n_{Y_i Y_j}-1)}{2} d^{n_{Y_i Y_j}-1}\delta + \ldots]) \\
&\leq m^2 n^2 \alpha^2 - (n^2 \delta \sum_{i,j=1}^{m} d + m^2 \delta \sum_{i,j=1}^{n} d) \\
&= m^2 n^2 \alpha^2 - 2m^2 n^2 \delta d
\end{aligned}
$$

## References


1. Srndic, N., Laskov, P.: Practical evasion of a learning-based classifier: a case study. In: Proceedings of the 2014 IEEE Symposium on Security and Privacy, SP 2014, pp. 197–211, IEEE Computer Society, Washington, DC (2014)
2. Lowd, D.: Good word attacks on statistical spam filters. In: Proceedings of the Second Conference on Email and Anti-Spam (CEAS) (2005)
3. Maiorca, D., Corona, I., Giacinto, G.: Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious PDF files detection. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, ASIA CCS 2013, pp. 119–130. ACM, New York (2013)
4. Elsayed, G.F., et al.: Adversarial examples that fool both human and computer vision, CoRR abs/1802.08195 (2018)

5. Szegedy, C., et al.: Intriguing properties of neural networks, CoRR abs1312.6199 (2013)
6. Lu, J., Sibai, H., Fabry, E., Forsyth, D.A.: No need to worry about adversarial examples in object detection in autonomous vehicles, CoRR abs/1707.03501 (2017)
7. Fawzi, A., Fawzi, O., Frossard, P.: Analysis of classifiers' robustness to adversarial perturbations. Mach. Learn. **107**(3), 481–508 (2018)
8. Biggio, B., et al.: Evasion attacks against machine learning at test time. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013, Part III. LNCS (LNAI), vol. 8190, pp. 387–402. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_25
9. Papernot, N., McDaniel, P.D., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks, CoRR abs1511.04508 (2016)
10. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.D.: On the (statistical) detection of adversarial examples, CoRR abs1702.06280 (2017)
11. Potey, M.M., Dhote, C., Sharma, D.H.: Homomorphic encryption for security of cloud data. Procedia Comput. Sci. **79**, 175–181 (2016). Proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 2016
12. Gretton, A., Borgwardt, K.M., Rasch, M.J., Scholkopf, B., Smola, A.: A kernel two-sample test. J. Mach. Learn. Res. **13**, 723–773 (2012)
13. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
14. Nateghizad, M., Erkin, Z., Lagendijk, R.L.: An efficient privacy-preserving comparison protocol in smart metering systems. EURASIP J. Inf. Secur. (1), 11 (2016)
15. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-preserving face recognition. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 235–253. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03168-7_14
16. Sheikhalishahi, M., Saracino, A., Mejri, M., Tawbi, N., Martinelli, F.: Fast and effective clustering of spam emails based on structural similarity. In: Garcia-Alfaro, J., Kranakis, E., Bonfante, G. (eds.) FPS 2015. LNCS, vol. 9482, pp. 195–211. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30303-1_12
17. Lindell, Y.: How to simulate it – a tutorial on the simulation proof technique. Tutorials on the Foundations of Cryptography. ISC, pp. 277–346. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57048-8_6
18. Nateghizad, M., Erkin, Z., Lagendijk, R.L.: Efficient and secure equality tests. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2016)
19. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1 (2001)
20. Munoz-Gonzalez, L., et al.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec 2017, New York, NY, USA, pp. 27–38 (2017)
21. Biggio, B., Fumera, G., Roli, F.: Security evaluation of pattern classifiers under attack, CoRR abs/1709.00609 (2017)
22. Jordaney, R., et al.: Transcend: detecting concept drift in malware classification models. In: 26th USENIX Security Symposium, Vancouver, BC, pp. 625–642. USENIX Association (2017)

23. Bos, J.W., Lauter, K., Naehrig, M.: Private predictive analysis on encrypted medical data. J. Biomed. Inform. **50**, 234–243 (2014)
24. Sheikhalishahi, M., Martinelli, F.: Privacy preserving clustering over horizontal and vertical partitioned data. In: 2017 IEEE Symposium on Computers and Communications, ISCC 2017, Heraklion, Greece, 3–6 July 2017, pp. 1237–1244 (2017)
25. Bost, R. Popa, R.A., Tu, S., Goldwasser, S.: Machine learning classification over encrypted data. In: 22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, 8–11 February 2015 (2015)
26. Sheikhalishahi, M., Martinelli, F.: Privacy-utility feature selection as a tool in private data classification. Distributed Computing and Artificial Intelligence, 14th International Conference. AISC, vol. 620, pp. 254–261. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62410-5_31
27. Costantino, G., Marra, A.L., Martinelli, F., Saracino, A., Sheikhalishahi, M.: Privacy-preserving text mining as a service. In: 2017 IEEE Symposium on Computers and Communications, ISCC 2017, Heraklion, Greece, 3–6 July 2017, pp. 890–897 (2017)
28. Martinelli, F., Saracino, A., Sheikhalishahi, M.: Modeling privacy aware information sharing systems: a formal and general approach. In: 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016, pp. 767–774 (2016)