Exploratory Study of Privacy Preserving Fraud Detection

Rémi Canillas
SIS ID
Lyon, France
remi.canillas@sisnet.fr

Rania Talbi
INSA Lyon – LIRIS
Lyon, France
rania.talbi@insa-lyon.fr

Sara Bouchenak
INSA Lyon – LIRIS
Lyon, France
sara.bouchenak@insa-lyon.fr

Omar Hasan
INSA Lyon – LIRIS
Lyon, France
omar.hasan@insa-lyon.fr

Lionel Brunie
INSA Lyon – LIRIS
Lyon, France
lionel.brunie@insa-lyon.fr

Laurent Sarrat
SIS ID
Lyon, France
laurent.sarrat@sisnet.fr

ABSTRACT

With the wide adoption of the Internet, digital transactions surge exponentially and so do the impersonation fraud. While machine learning techniques show strong promise to be the building block for digital fraud detection systems, clients may be reluctant to share the raw data with such systems due to privacy concerns. The emerging privacy preserving machine learning techniques that employ homomorphic encryption to resolve this conundrum unfortunately increases the computational overhead of detection. In this paper, we present a first-of-a-kind empirical performance study of a private fraud detection system developed at SiS ID, a French business security platform. A privacy-preserving decision tree which can classify transactions into four risk classes (safe, moderately risky, very risky and fraud) is trained on more than 160000 real world transactions, and we quantitatively compare the classification accuracy, latency and network bandwidth under various combinations of encryption parameters and learning hyper-parameters, in order to explore the impact of the configuration on the performances. Our results show that the computation and communication overhead of processing encrypted data increases by an order of magnitude of 5, and highly depends on the configuration of the encryption key and the number of nodes in the decision tree.

ACM Reference Format:
Rémi Canillas, Rania Talbi, Sara Bouchenak, Omar Hasan, Lionel Brunie, and Laurent Sarrat. 2018. Exploratory Study of Privacy Preserving Fraud Detection. In *19th International Middleware Conference Industry (Middleware '18 Industry), December 10–14, 2018, Rennes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3284028.3284032

Middleware '18 Industry, December 10–14, 2018, Rennes, France

ACM ISBN 978-1-4503-6016-6/18/12…$15.00
https://doi.org/10.1145/3284028.3284032

1 INTRODUCTION

The success of today's online business platforms highly depends on the security and integrity of the digital transaction that they process. However, due to the increasing volume of transaction and the globalization of business exchanges, it becomes harder and harder to detect and prevent frauds. Frauds and identity theft losses amounted to $16 billion in the U.S. in 2016 [3]. A recent study estimates that fraud causes UK private sector losses of £144 billion a year [1], and another study reports that 56% of French companies were victims of identity theft in 2016 [6]. Consequently, today's platforms keenly seek for online fraud detection solutions that can verify the legitimacy of each transaction upon request of their clients and can be easily integrated into the process.

Machine learning techniques, such as decision tree, are a promising solution in classifying transactions on the fly based on a classifier that is trained from transaction history in an off-line fashion. However, the adoption of such a system is hindered by two major concerns: Firstly, such a machine learning-based fraud detection system might be the target of attacks from malicious users who try to alter the detection results. Secondly, clients may be reluctant to reveal the details of current transactions to the detection system because of strong concerns about the privacy and security of their data. Consequently, transaction data needs to be encrypted not only during the communication between the users and the detection system but also during the computation phase of the online classification. Allowing operations on encrypted data is essentially the basic principle of homomorphic encryption, thus allowing the classifier to predict classes based solely on the encrypted data.

While privacy-preserving machine learning tools have started to emerge for business practice, little is known on their exact performance impact on the classification latency for real world applications. Clearly, the security and privacy guarantees come at the additional processing overhead of homomorphic encryption. The persisting question is: are existing privacy-preserving solution sufficiently performant to be adopted by real business systems? We particularly focus on a fraud detection system.

In this paper, we empirically study the performance cost of online fraud detection systems that classify transactions over the encrypted data. Specifically, we apply the existing

homomorphic encryption-based tool Ciphermed [4] on the fraud detection system for SiS ID, a French business security platform that specializes in fraud detection in digital transactions. We consider the private preserving decision tree and focus on its inference phase, under the assumption that the model training phase is conducted in an off-line and secure manner. We aim to quantify the performance impact of privacy preserving schemes and provide an empirical analysis to help guide the design of private machine learning systems, and in particular private fraud detection.

Our evaluation is based on the more than 160000 transactions from 19072 clients collected from SiS ID. We first train the decision tree classifier on the historical transactions. The classifier separates transactions into four classes: safe, risky, very risky, and fraud. We measure the latency of such a fraud detection system under different configurations of classifiers (e.g. the depth of the decision tree) and encryption schemes (e.g. the sizes of encryption keys). Our results show that privacy-preserved fraud detection systems can achieve the same accuracy as non-preserving ones but the computational time and network bandwidth requirement per classification request increases by 4 to 6 orders of magnitude compared to non privacy-preserving systems. Moreover, we show that larger encryption key sizes and larger numbers of interior nodes in the decision tree increase the computational time of a private tree.

1.1 Related Work

Privacy-preserving machine learning is indeed an active research area, examples include privacy-preserving bayesian networks [12, 13], privacy-preserving random forest [14], or privacy-preserving decision tree [4]. Privacy-preserving neural network also received a lot of attention [2] [15]. However, none of this work focuses solely on Fraud Detection in a B2B business environment.

1.2 Organisation

The remainder of the paper is organized as follows: Section 2 introduces the problem of fraud detection in a B2B environment, describes the B2B data of the SiS ID company, and describes the system used by SiS ID for fraud detection. Section 3 presents our experience in applying privacy preserving vs. non-privacy preserving classifiers for B2B fraud detection. Finally, in Section 4 we draw our conclusions and perspectives for future work.

2 SIS ID PLATFORM

SiS ID is a business security platform where multiple clients verify the identity of their beneficiary before conducting finance related transactions. One of the key service offered by SiS ID is fraud detection. An overview of the fraud detection system's architecture is depicted in Fig. 1. It consists of two flows: (i) the online detection (illustrated with red arrows), where potentially fraudulent transactions are classified, and (ii) the off-line model training (shown with black arrows), where the system uses past transactions (called "historical") to train the classification system. Upon receiving a transaction request from a client, the fraud detection system classifies it into four classes (namely: safe, moderately risky, very risky and fraud), based on the data model derived from historical transactions. The platform then forwards the classification results to the clients who make final decision to execute the transactions or abort them.

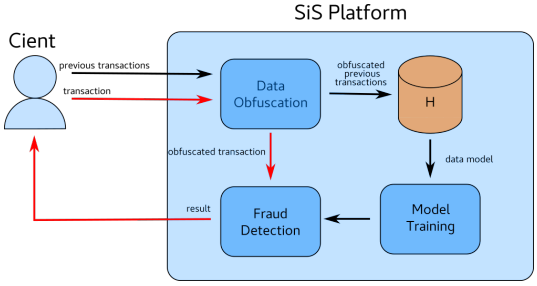


Figure 1: Schematic overview of the SiS Platform.
A client first sends a set of historical transactions to the SiS platform that stores them in a repository H. The platform uses all the transactions contained in H to train a classifier. This classifier is then used to detect fraud in transactions.

In the following, we first explain the threat model considered in SiS ID, and then give a detailed overview of the transactions used in its fraud detection system.

Threat Model

We assume a honest-but-curious threat model [10], implying that all users try to learn about the input of the others as much as possible by observing all the information they can receive. In the particular context of SiS ID platform, clients want the system to learn as less information as possible from any transaction sent to the platform, while SiS ID wants clients to learn as less information as possible about the detection model.

2.1 Transactions

Our study is based on the finance-related transactions extracted from the SiS business platform. We first explain the composition of transactions and then provide a statistical analysis of the available dataset.

A transaction typically involves two companies, i.e., emitter and beneficiary, who engage in monetary exchanges, usually when the beneficiary delivers a service or sells merchandise to the emitter. The raw information contained in a transaction is the IBAN (banking identification number) and SIRET (company identification number) of the beneficiary. Using IBAN and SIRET, the SiS ID platform then performs a feature extraction, creating a so-called "enriched transaction".

An enriched transaction has a vector of 11 features, resulting from different functions conducted using both the platform internal information and publicly available online resources. These features are a mix of categorical, numerical and Boolean features. The exact definition of each feature is summarized in table 1.

Name	Definition	Data Type	Range
client-payment-history	Verify the status of the IBAN and SIRET of the beneficiary company in the history of the emitter company.	Categorical	[couple-registered, company-iban-registered, company-registered, iban-registered, unknown]
community-payment-history	Verify the status of IBAN and SIRET of the beneficiary company in the history of all clients of the platform.	Categorical	[couple-registered, company-iban-registered, company-registered, iban-registered, unknown]
company-exists	Check if the SIRET of the beneficiary company is already associated with a company on the platform.	Boolean	[0, 1]
match-company-iban-country	Check if the SIRET of the beneficiary company is registered in its corresponding country.	Boolean	[0, 1]
payment-identity-exists	Verify the status of the IBAN of the beneficiary company on the platform.	Categorical	[invalid, pending, valid, disabled, unknown]
ping-iban-siret	Verify if the company identified by the SIRET of the beneficiary company has registered the IBAN used for the transaction on the platform .	Categorical	[invalid, pending, valid, disabled, unknown]
valid-company	Check if the company identified by the SIRET of the beneficiary company is still officially registered.	Boolean	[0, 1]
valid-company-managers	Verify if the SIRET of the beneficiary company is managed by valid users of the platform, and if yes, how many.	Numeric	$\mathbb{N}$
valid-iban-format	Check if the IBAN of the beneficiary company is correctly formatted.	Boolean	[0, 1]
valid-iban-swift	Check if the IBAN of the beneficiary company is linked to a valid bank account	Boolean	[0,1]
valid-siret-format	Check if the SIRET of the beneficiary company is correctly formatted.	Boolean	[0, 1]

Table 1: Features of transactions at the SiS ID platform.

2.1.1 Dataset Statistics. We use 165592 transactions collected from the SiS ID database as the base of this study. This dataset contains enriched transactions that were submitted to the fraud detection system of the SiS ID platform during the span of recent six months. We briefly summarizes the relevant information, focusing on the relevance of each feature for our fraud detection application.

Relevance - Four features (*valid-company, valid-iban-format, valid-iban-swift* and *valid-siret-format*) have constant values for every record in the dataset. Therefore, they provide no relevant information for detection fraud and thus are removed from our analysis.

2.2 Private Fraud Detection

To build an online fraud detection system, there are several challenges that have to be solved. First of all, a classifier that can accurately predict transaction outcomes need to be trained using the history of transactions from the platform's users. Machine learning models are a classical and efficient way to classify data instances (here transactions) into multiple classes (here, the four risk levels), and due to the relative simplicity of our transaction, Decision Trees seems to be a relevant solution for fraud detection. Secondly, users might be extra cautious or not even willing to provide information of active or historical transactions to SiS ID for online detection, due to the fact that these transactions are very sensitive business information that might be potentially harmful in adversarial hands. This motivates us to augment the fraud detection module of SiS ID by enabling online classifying encrypted transaction, using an homomorphic cryptography-based classifier. While ideally, both training and testing phase should be conducted in a privacy-preserving way, the focus of our experimentation is solely on the performance of the classification phase, and the training of the model is considered out of scope of this paper. However, in order to bootstrap the fraud detection platform, a selection of companies were willing to let their history of transaction in clear to SiS ID in order to train the data models.

2.2.1 Machine Learning Model. Here, we specifically consider decision tree as the classification model. A decision tree is a classifier that partitions the feature vector space one attribute at a time; interior nodes in the tree correspond to partitioning rules, and leaf nodes correspond to class labels.

Feature	Feature importance
client-payment-history	0.028986569822114
community-payment-history	0.957814663067485
company-exists	7.66851075909508E-08
match-company-iban-country	0.000134607543773
payment-identity-exists	0
ping-iban-siret	0.012589215547515
valid-company-managers	0.000474867334005

Table 2: Decision tree classifier – Feature Importance

A data instance is classified by walking the tree starting from the root, using the partitioning rule at each node to decide which branch to take until a leaf node is reached. A decision tree has several hyper-parameters such as the tree depth (which limits the partitioning to avoid over fitting), or the minimum number of samples needed to create a new partition, that emphasis on the purity of the leafs.

2.2.2 Homomorphic Cryptography-Based Classifier. We use the Ciphermed open-source prototype that applies additive homomorphic cryptography for private data classification [4]. It allows a client to privately submit a feature vector to an untrusted server, which then classifies the obfuscated feature vector according to a model previously established, and privately outputs the result of this classification to the client, so that the client learns no information about the model. Ciphermed is based on three different cryptosystems: (1) the QR (Quadratic Residuosity) cryptosystem of Goldwasser-Micali, (2) the Pallier cryptosystem [11] and (3) a leveled fully homomorphic encryption (FHE) scheme, HELib [8]. These cryptography systems are then used to design *building blocks* for privacy-preserving classifiers, e.g., a homomorphic cryptography-based *comparison* function, a homomorphic cryptography-based *argmax* function, and a homomorphic cryptography-based *dot product* function. These functions are finally combined to create different kind of private classifiers, in particular a private decision tree classifier that we use in our private fraud detection system.

3 EXPERIMENTAL ANALYSIS

Our objective is to evaluate the effectiveness and performance impact of privacy preserving fraud detection on a real-life dataset, using 165592 transactions from 19072 clients, collected from the production database of SiS ID company. We focus on comparing the regular and private decision tree, using metrics of detection accuracy and classification time (also called "latency") and number of bytes exchanged (also called "bandwidth") on the client side. We first explain the experimental setup and then analyze the performance results.

3.1 Experimental Setup

Our test bed for fraud detection consists of virtualized computer system composed of a server running Ubuntu 18.04.1 LTS with 4 GiB of RAM and 4 cores 2,49 Ghz, and a client also running Ubuntu 18.04.1 LTS with 4 GiB of RAM and 4 cores 2,49 Ghz. In the privacy preserving case, the client and server are executed as separate single-threaded programs communicating on a local network. In this work, we make the strong assumption that the communication overhead is negligible in regard to the computation overhead.

Due to scalability issues from the underlying encryption system, we uniformly partition our dataset in samples of 5000 records, creating 34 different partitions. 90% of a partition is used to train our classifiers using the scikit-learn python library, and the remaining 10% is reserved for the online inference phase. The prototype used for this evaluation in available online[1].

Metrics of interests. We focus on three types of metrics of interests: (i) F-measure representing the detection accuracy, (ii) Time per detection request in milliseconds, and (iii) Exchanged Bytes in bytes. F-measure is the harmonic average of precision and recall, where the former is the percentage of true positive over all positive prediction and the latter is the percentage of true positive over all positive samples. F-measure ranges between [0 1] and 1 indicates perfect precision and recall for the detection.

As for the network bandwidth, we measure the amount of bytes exchanged between the client and server side for each classification request to the system.

3.2 Private v.s. Non-private Tree

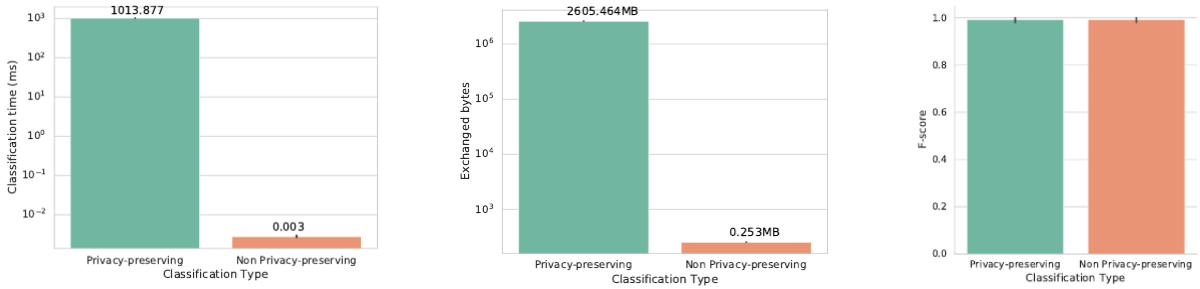
We first build a regular decision tree with the number of interior nodes of tree ranging from 3 to 9, using the tree depth parameter. Similarly, we then develop a online private classifier, that is based on private-preserving decision tree with different encryption key sizes. Prior to explaining their performance differences, we explain the effectiveness of regular tree in predicting fraud detection. Table 2 summarizes the feature importance, a higher value of which implies a better explanation power to the class outcome.

Fig. 2 gives an overview of the performance differences between private and non-private tree classifiers, averaging all scenarios and parameters considered. It shows that, on average, the time taken to classify a transaction increases up to 6 orders of magnitude, growing from 0.003 ms to 1013.877 ms. This extreme result is due to the many operations involved in the evaluation of homomorphically encrypted ciphers. Fig. 2(b) shows that the size of the data exchanged between the client and the server also increase by more than 4 orders of magnitudes, from 0.253 MB to 2605.464 MB. This is due to the larger messages of encrypted data than its clear counterpart. Again, a large variation in the size of messages sent can be seen. However, Fig. 2c shows that using homomorphic encryption can ensure the confidentiality of the feature vector and at the same time does not incur any loss in the accuracy score of the decision tree model.

3.3 Impact of Model Complexity on Private Tree

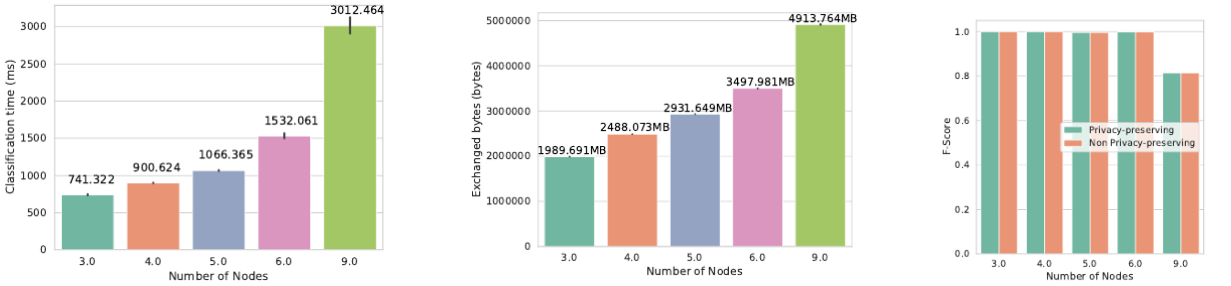
Here, we specifically zoom into the complexity of decision tree, i.e., the number of interior nodes the tree, and study its impact on the private tree. Fig. 3 summarizes the performance in terms of classification time, number of exchanged bytes,

[1] https://github.com/remicanillas/PPML_Bench



(a) Classification request latency (b) Network bandwidth (c) Accuracy of classification

Figure 2: Decision tree classifier – Privacy preserving vs. non-privacy preserving classification



(a) Classification request latency (b) Network bandwidth (c) Overall accuracy of classification

Figure 3: Impact of the number of interior nodes in a decision tree – Privacy preservation vs. non-privacy preservation

and F-score, for trees with 3, 4, 5, 6 and 9 interior nodes, with a fixed key size of 512 bytes for the encryption systems. Although a decision tree with only three interior nodes might seem overly simple, we aim to explore the trade-off between simplicity and computation time.

Fig. 3(a) shows that the computation times of the privacy-preserving fraud detection system increases with the number of interior nodes of the decision tree, from 741.322 ms to 3012.464 ms, while the non privacy-preserving systems shows no increase in computation time. The homomorphic operations results into 4 times more computational overhead, compared to the regular tree operations. Fig. 3b shows that the quantity of bytes exchanged follows the same behavior: an increase from 1989.690 MB to 3248.841 MB. As for the accuracy, Fig.3(c) shows that a more complex model does not offer a better F-score. This result might be explained by the fact that more complex models tend to lead to over-fitting, and thus fail to classify feature vectors that are not part of the training dataset.

Moreover, we quantitatively identify the relationship between the computation time and the number of interior nodes by fitting different degrees of polynomial regression functions. We see that the second order function can best explain the dependency,

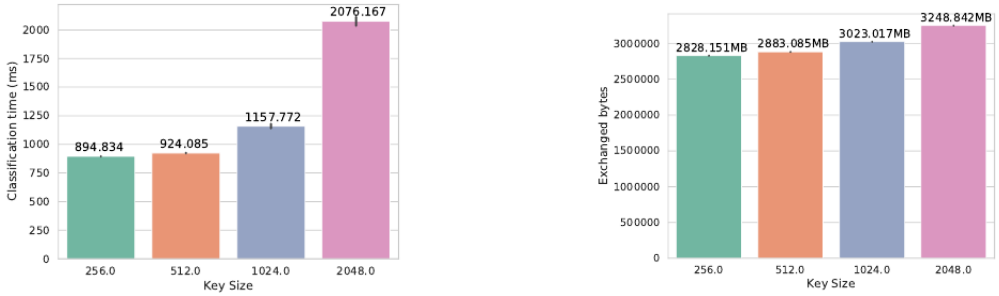
$$latency = 215.31 + 54.36(no.i.nodes) + 19.44(no.i.nodes)^2,$$

indicating a quadratic increase in the number of interior nodes.

3.4 Impact of Security Level on Privacy Preserving Fraud Detection

This section focuses on the analysis of the impact of the security level on the performance of the fraud detection system. In our context of encryption, the security of our system is given by the size of the encryption key: indeed, most of the attacks on encryption systems nowadays targets private keys, and the smaller the key, the easier it is to break it [9]. Fig. 4 summarizes the performance of different instances of our privacy-preserving system using different key sizes, respectively 256, 512, 1024 and 2048 bytes, for a fixed number of interior nodes (5 nodes).

Our experimental analysis shows that the computation time is, not surprisingly, strongly affected by the key size, with a minimum computation time of 894.834 ms of a key of 256 bytes, and a maximum computation time of 2076.167 ms for a key of 2048 bytes, as shown in Fig. 4a. Also, we remark that a higher key tends to lead to a greater variation in terms of classification time, as shown by the higher error bar for a key size of 2048. Fig. 4b shows us a similar trend for the quantity of bytes exchanged between the client and server side: a small key of 256 bytes leads to a transfer of



(a) Classification request latency **(b) Network bandwidth**

Figure 4: Performance and cost of privacy preserving classification with various security levels

2828.151 MB while a key of 2048 bytes leads to a transfer of 3248.841 MB.

This might be explained by the fact that, the longer the key, the longer the cipher produced by the encryption system, and therefore the longer it is to compute operation on them.

It is worth noting that the size of the key has no impact whatsoever on the F-score of the classifiers, as shown in Fig. 2c. This comes from the fact that, apart from being encrypted, the feature vectors are not altered in any way during the classification, and, due to the fact that homomorphic encryption allows the exact same operations to be conducted on encrypted data, no information is lost during the classification.

4 CONCLUSION

In this paper, we first present a comprehensive overview of the fraud detection system used by SiS ID, a French business security platform, and then provide an exploratory study of private fraud detection systems that enable the online classification of fully encrypted transactions. We empirically study if one can efficiently apply the state-of-art homomorphic privacy-preserving machine learning on fraud detection systems. Our key findings are that (i) per classification request latency is significantly affected by the homomorphic encryption overhead, increasing from few micro-seconds for non-private learning to up to a second, (ii) the number of interior nodes can increase the classification time per request in a quadratic fashion, and (iii) the security level of the system, represented by the size of the encryption key, can also significantly increase both classification time and network bandwidth. These results indicate that, in order to be practical in a real-life situation, it is important to strive to achieve the optimal trade-off between model complexity and security guarantees for privacy-preserving learning systems. Future work includes evaluating the impact of other Machine Learning systems (Deep Neural Networks, Bayesian Networks ...) and other privacy preserving techniques (Differential privacy [5] , SMC [7] ...)

ACKNOWLEDGMENT

This work benefited from the support of the French National Research Agency (ANR), through the SIBIL-Lab project (ANR-17-LCV2-0014).

REFERENCES

[1] A. Williams. 2017 (accessed February 1, 2018). Fraudsters Target UK Directors Through Companies House. The Financial Time. (2017 (accessed February 1, 2018)).

[2] M. Al, T. Chanyaswad, and S. Y. Kung. 2018. Multi-Kernel, Deep Neural Network and Hybrid Models for Privacy Preserving Machine Learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2891–2895. https://doi.org/10.1109/ICASSP.2018.8462336

[3] B. Sullivan. 2017 (accessed February 1, 2018). Identity Theft Hit an All-Time High in 2016. USA Today. (2017 (accessed February 1, 2018)).

[4] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. 2015. Machine Learning Classification over Encrypted Data. *Proceedings 2015 Network and Distributed System Security Symposium* February (2015), 1–31. https://doi.org/10.14722/ndss.2015.23241

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

[6] Euler Hermes. 2016 (accessed February 1, 2018. French fraud study reveals rapidly increasing business cyber crime threat. DFCG. (2016 (accessed February 1, 2018).

[7] Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version* 78 (1998).

[8] S. Halevi. (accessed November 7, 2017). HElib - An Implementation of Homomorphic Encryption. https://github.com/shaih/HElib. ((accessed November 7, 2017)).

[9] Arjen K. Lenstra and Eric R. Verheul. 2001. Selecting cryptographic key sizes. *Journal of Cryptology* 14, 4 (2001), 255–293. https://doi.org/10.1007/s00145-001-0009-4

[10] Goldreich Oded. 2004. Foundations of Cryptography. Basic Applications, vol. 2. (2004).

[11] P Paillier. [n. d.]. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 99)*.

[12] O. Regnier-Coudert and J. McCall. 2011. Privacy-preserving Approach to Bayesian Network Structure Learning from Distributed Data. In *The 13th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO '11)*. Dublin, Ireland. https://doi.org/10.1145/2001858.2002103

[13] S. Samet and A. Miri. 2009. Privacy-Preserving Bayesian Network for Horizontally Partitioned Data. In *The 2009 International Conference on Computational Science and Engineering*, Vol. 3. https://doi.org/10.1109/CSE.2009.94

[14] G. Szucs. 2013. Random Response Forest for Privacy-Preserving Classification. *Journal of Computational Engineering* (2013).

[15] Qiang Zhu and Xixiang Lv. 2018. 2P-DNN : Privacy-Preserving Deep Neural Networks Based on Homomorphic Cryptosystem. *CoRR* abs/1807.08459 (2018). arXiv:1807.08459 http://arxiv.org/abs/1807.08459