Securely outsourcing machine learning with multiple users

Ping Li[1]*, Hongyang Yan[2], Chong-Zhi Gao[1], Yu Wang[1], Liaoliang Jiang[1], and Yuefang Huang[1]
[1]Guangzhou University, Guangzhou, Guangdong, China
[2]Nankai University, Tianjin, China

Abstract

In recent years, machine learning has been widely used in data analysis for predicting models, such as face/pattern recognition, image processing, simultaneous interpretation and speech recognition. However, these massive data are sensitive, which raises privacy concerns. Therefore, to protect the data privacy, in this paper, we design a scheme for securely training machine learning model on the jointed data that provided from different sources. Our scheme falls in the two-server-aided model and allows one server to conduct most of computations, and another server to provides auxiliary computation. We prove the security of our scheme in the semi-honest model.

Keywords: secure outsourcing, machine learning, data privacy

1 Introduction

Machine learning has became an indispensable tool for learning/mining knowledge from massive data in recent years. Due to the openness of cloud computing platform data resources and the clouds are not fully trusted, machine learning has aroused data privacy concerns. For example, images data, medical health data and credit recordings may contain personal sensitive items—users' face, medical history, your consumption record. If these data leaked, attackers can learn/mine some "knowledge" from these data and use "knowledge" illegally. This may lead to economic loss or personal safety. Consequently, data privacy and confidentiality is necessary to be protected when processing machine learning related outsourcing computations [1, 7, 9, 10]. One way to preserve the data privacy and confidentiality is to pre-process the data by using cryptographic methods before uploading it to the server [12, 4, 5, 8].

In this paper, we focus on machine learning model for training neural networks over combining data from different sources and utilize the two-server-aided model. There are two phases included in our proposed scheme: in the setup phase, multiple users secretly share and encrypt their data among two non-colluding and independent cloud servers; in the training phase, based on the jointed data, two non-colluding and independent cloud servers train neural network models and get nothing except the trained model. Specifically, the main contributions of this paper are summarized as follows:

- Our scheme is designed to conduct the data encrypted under different public keys and to preserve the data privacy of users. Users only perform the encryption/decryption operation.
- Our scheme is divided into two independent phases: setup phase and training phase. Two non-colluding cloud servers only perform the computation operation in training phase.
- To efficiently compute the active function, we use secure multi-party computation (MPC) as the key technique and the weight vector can be treated as weight matrix for multiple input vectors.

*Corresponding author: School of Computer Science, Guangzhou University, Guangzhou City, No. 230, Waihuan Xi Road, Guangzhou Higher Education Mega Center, Guangzhou 510006, P. R. China, Tel: +86-15013151389, Email: liping26@mail2.sysu.edu.cn

2 Preliminaries

BCP Homomorphic Encryption [3]. In this paper, to realize the privacy-preserving machine learning over the jointed data, we take the BCP cryptosystem as our encryption scheme, which consists of the following algorithms.

1) $(\mathtt{pp},\mathtt{msk}) \leftarrow \mathsf{ParaGen}(1^\kappa)$: the parameter generation algorithm takes as input a security parameter κ, and outputs the *master key* $\mathtt{msk} = (p',q')$ and the system's *public parameters* $\mathtt{pp} = (N,k,g)$: let $N = pq$ be a safe prime RSA-modulus with bit-length κ, where $p = 2p'+1$ and $q = 2q'+1$ for distinct prime p' and q'. Given a value $g \in \mathbb{Z}^*_{N^2}$ with order $pp'qq'$ such that $g^{p'q'} \equiv 1+kN \pmod{N^2}$ for $k \in [1,N-1]$.

2) $(\mathtt{sk},\mathtt{pk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$: given a random element $a \in \mathbb{Z}_{N^2}$ and set $h = g^a \bmod N^2$, the key generation algorithm returns the public key $\mathtt{pk} = h$ and the private key $\mathtt{sk} = a$.

3) $(A,B) \leftarrow \mathsf{Enc}_{(\mathtt{pp},\mathtt{pk})}(m)$: encryption algorithm outputs a ciphertext (A,B), where $A = g^r \bmod N^2$ and $B = h^r(1+mN) \bmod N^2$.

4) $m \leftarrow \mathsf{uDec}_{(\mathtt{pp},\mathtt{sk})}(A,B)$: user decryption algorithm returns message m as $m = \frac{\frac{B}{A^a}-1 \bmod N^2}{N}$, or the special symbol '$\perp'$ if it is an invalid ciphertext.

5) $m \leftarrow \mathsf{mDec}_{(\mathtt{pp},\mathtt{pk},\mathtt{msk})}(A,B)$: master decryption algorithm outputs message m or the special symbol '$\perp'$ if it is an invalid ciphertext. Firstly, compute the user private key as $a \bmod N = \frac{h^{p'q'}-1 \bmod N^2}{N} \cdot k^{-1} \bmod N$. Secondly, compute $r \bmod N = \frac{A^{p'q'}-1 \bmod N^2}{N} \cdot k^{-1} \bmod N$, then set $\tau = ar \bmod N$. Finally, the massage m can be computed as $m = \frac{(\frac{B}{g^\tau})^{p'q'}-1 \bmod N^2}{N} \cdot \alpha^{-1} \bmod N$, where k^{-1} and α^{-1} denote the inverse of $k \bmod N$ and $p'q' \bmod N$, respectively.

Pseudorandom Generator. Loosely speaking, a secure pseudorandom generator (PRG) [2] is a deterministic algorithm that expands short random seeds (with some fixed length) into much longer bit sequences that appear to be "random". In other words, the PRG is security if the output sequences of PRG on a uniformly random seed and the truly random sequences are computationally indistinguishable.

Neural networks. In general, most neural networks are constructed by groups of unites called layers: *input layer-hidden layer-output layer*. Assume that x_{ji} is the i-th input to neuron j (with corresponding weight w_{ji}). Let $net_j = \sum_i w_{ji}x_{ji}$ be the inner product of input and weight for neuron j and $f_j = f(net_j)$ be the output computed by activation function f. The target output of neuron j is y_j. The error function $E_d(\cdot)$ for the training sample d can be defined as $E_d(\mathbf{w}) = \frac{1}{2}\sum_{k\in D'}(f_k - y_k)^2$, where D' is the set of output neurons in the neural networks.

To train an acceptable weight vector, we need to modify the weight at each step according to the perceptron training rule: $w_{ji} = w_{ji} + \triangle\, w_{ji}$, where $\triangle\, w_{ji} = -\eta\frac{\partial E_d(\mathbf{w})}{\partial w_{ji}}$. To compute $\triangle\, w_{ji}$, two cases should be considered when handle this task:

1. Neuron j is an output neuron for the backward propagation network: the weight update rule is implemented by $\triangle\, w_{ji} = -\eta\frac{\partial E_d(\mathbf{w})}{\partial w_{ji}} = \eta(y_j - f_j)f_j(1-f_j)x_{ji}$.

2. Neuron j is an hidden neuron in the backward propagation network: the training rule for w_{ji} is influenced by the neurons whose direct inputs include the output of neuron j. We use D to denote all these neurons. Each element j in D, net_j can influence the outputs and the error function $E_d(\cdot)$. Therefore, $\frac{\partial E_d(\mathbf{w})}{\partial net_j} = \sum_{k\in D}\frac{\partial E_d(\mathbf{w})}{\partial net_k}\cdot\frac{\partial net_k}{\partial net_j} = \sum_{k\in D}(\frac{\partial E_d(\mathbf{w})}{\partial net_k})\cdot(w_{kj}\cdot(f_j(1-f_j)))$. Finally, $\triangle\, w_{ji} = -\eta\frac{\partial E_d(\mathbf{w})}{\partial w_{ji}}x_{ji} = \eta\delta_j x_{ji}$, where $\delta_j = f_j(1-f_j)\sum_{k\in D}\delta_k w_{kj}$.

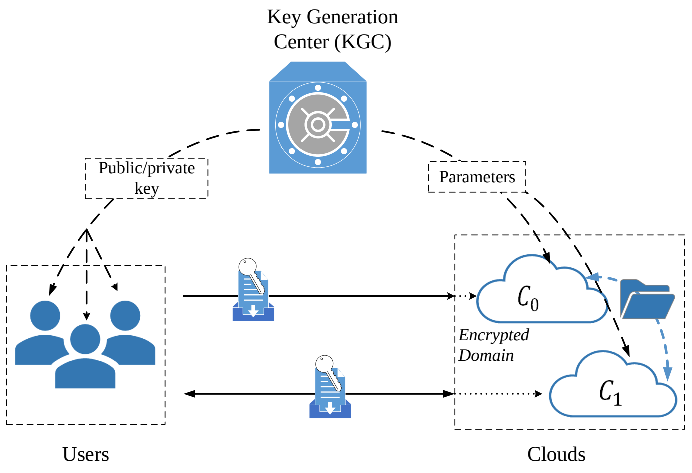

Figure 1: Our system model under consideration

3 Problem Formulation

3.1 System Model

In this subsection, we consider a architecture of privacy-preserving machine learning over the jointed data. As illustrated in Fig.1, our system consists of three parties: Users, Clouds ($\mathscr{C}_0$ and $\mathscr{C}_1$) and a Key Generation Center (KGC).

- Key Generation Center (KGC). In our system we assume that KGC is an authority entity trusted by all the parties. KGC setups the BCP cryptosystem and distributes the parameters and keys to the Clouds and Users.

- Users. Let Users be a set with n mutually distrusting users $\{\mathsf{U}_1, \mathsf{U}_2, \cdots, \mathsf{U}_n\}$. In our system, the data can be arbitrarily (i.e., vertically or horizontally) partitioned. To protect the data privacy against unauthorized access, each user secretly shares its plaintext and sends the ciphertexts of shares which encrypted under its public key to two non-colluding cloud servers.

- Clouds. Clouds are composed of two non-colluding and independent servers $\mathscr{C}_0$, $\mathscr{C}_1$. They perform several cryptographic interactive protocols with each other such that the outsourced data from Users are transformed into data encrypted under a unified pubic key. And they jointly execute a secure machine learning model by using some interactive protocols.

3.2 Security Model

To protect the data privacy and achieve the goal of privacy-preserving machine learning, we assume Users and Clouds are *semi-honest*, and require Clouds are *non-colluding* and *independent*. Specially, we stress that there is no collusion between $\mathscr{C}_0$ and $\mathscr{C}_1$, between any two of users or between any user and at most one of Clouds. In this semi-honest model, Clouds only *passively* perform the protocol and have no ability to *actively* modify the data flow. Specifically, Clouds only obtain the size of data.

3.3 Attack Model

We assume a semi-honest adversary $\mathscr{A}$ who may obtain the plaintext of Users with the following abilities: (i) $\mathscr{A}$ may corrupt any subset of Users to obtain all plaintexts belonging to Users; (ii) $\mathscr{A}$ may corrupt $\mathscr{C}_0$ (or $\mathscr{C}_1$) to achieve plaintext of all outsourced data from the Users.

4 Privacy-preserving Machine Learning

4.1 Our Scheme

Assume that Users is a set of n mutually distrusting users $\mathsf{U}_1, \mathsf{U}_2, \cdots, \mathsf{U}_n$, where each user U_i has its privacy data m_i and public/private key $(\mathtt{pk}_i, \mathtt{sk}_i)$, $i = 1, 2, \cdots, n$. To get some "knowledge " from the jointed data of Users and protect each user's data privacy, our solution can be divided into two phases:
Set up phase: 1) Initially, KGC sets up BCP cryptosystem and generates $(\mathtt{pp}, \mathtt{msk})$ and $(\mathtt{sk}, \mathtt{pk})$ by running $\mathsf{ParaGen}$ and KeyGen, respectively. For the private key $\mathtt{sk}$, KGC divides it into n pieces $\mathtt{sk} = \mathtt{sk}_1 + \cdots \mathtt{sk}_n$, where $\mathtt{sk}_i$ $(i \in [1, n])$ is chosen uniformly at random from the key space of BCP cryptosystem. KGC computes $\mathtt{pk}_i = g^{\mathtt{sk}_i} \bmod N$ and distributes $(\mathtt{pp}, (\mathtt{pk}_i, \mathtt{sk}_i))$ $(i \in [1, n])$ to U_i. At the same time, KGC sends $(\mathtt{pp}, \mathtt{msk})$ and $\mathtt{pp}$ to cloud $\mathscr{C}_0$ and $\mathscr{C}_1$, respectively. 2) After receiving $\mathtt{pp}$ and $(\mathtt{pk}_i, \mathtt{sk}_i)$, each user U_i uses PRG generates a pseudorandom number R_i, let $m_{i0} = R_i$ and $m_{i1} = m_i - R_i$. Then, U_i uploads $c_{i0} = \mathsf{Enc}_{(\mathtt{pp}, \mathtt{pk}_i)}(m_{i0})$ and $c_{i1} = \mathsf{Enc}_{(\mathtt{pp}, \mathtt{pk}_i)}(m_{i1})$ to the cloud $\mathscr{C}_0$ and $\mathscr{C}_1$, respectively, where $m_i = m_{i0} + m_{i1}, i = 1, 2, \cdots, n$.

After the set up phase, $\mathscr{C}_0$ and $\mathscr{C}_1$ have collected a part of outsourced data from different users respectively. Right now, $\mathscr{C}_0$ and $\mathscr{C}_1$ can jointly perform some cryptographic interactive protocols such that the data encrypted under different public keys transformed into data encrypted under the unified public key. This is due to the fact that MPC only works for encryptions under the same public key. After executing the transformation protocol, $\mathscr{C}_0$ and $\mathscr{C}_1$ can learn/train machine learning by some cryptographic interactive protocols in the training phase.
Training phase: The key operation in this phase is the computation of the activation functions. Since the BCP cryptosystem is additive homomorphic encryption, we need the activation functions only including the operation of addition and multiplication, so they can be computed over the ciphertext domain.

In the existing literatures, lots of works use polynomial approximation to compute the activation function, such as [13, 14] applied a low-degree polynomial function to approximate the sigmoid function. However, the low-degree (3 or 5) polynomial approximation is still high to be used with HE schemes, since the computational complexity and accuracy is depend on this degree.

In this work, we take rectified linear unit (ReLU) function $g(z) = max\{0, z\}$ [6, 11] as activation function, since the ReLU function is nearly linear. We only need to compare the size of 0 and z over the ciphertext domain.

After all computations are done, each user retrieves the encrypted output of the $\mathscr{C}_1$ and decrypts jointly with its respective private key.

4.2 Security analysis

The security analysis is considered in the semi-honest model, meaning that every protocol parties are honestly follow the protocol but try their best to analyze or infer the data flow to learn additional information about other parties' inputs, intermediate results and output results by gathering the protocols' transcripts. Recall that some cryptographic interactive protocols between $\mathscr{C}_0$ and $\mathscr{C}_1$ are executed depending on the "blinding-the-message" techniques.

We require the adversary $\mathscr{A}$ has the polynomial bounded of computing power, since the BCP cryptosystem guarantees the semantic security in the semi-honest model. $\mathscr{A}$ may corrupt any subset of Users to obtain their ciphertext, however, $\mathscr{A}$ will not be able to decrypt the ciphertext without knowing the corrupted Users' private key due to the semantic security of the BCP cryptosystem. $\mathscr{A}$ may compromise $\mathscr{C}_0$ (or $\mathscr{C}_1$) to achieve the ciphertext and private data of Users. However, $\mathscr{A}$ is unable to recover the Users's private key to decrypt the ciphertext, because the message is pre-processed by splitting into two pieces, where one piece is a pesudorandom number and another price is the message reduce a pesudorandom number. $\mathscr{C}_0$ holds the master key and only obtains a pesudorandom number after decrypting; $\mathscr{C}_1$ only gets a blinded message (the message reduce a pesudorandom number) without leaking the message information.

5 Conclusion and Further Work

In this paper, we have proposed a scheme for securely training machine learning model while supporting that uses' inputs data are learned by the accessed servers. In our scheme, users split their data into two parts and encrypt each split data with different pubic keys and outsource the ciphertexts to two servers. The servers can transform the data encrypted under different public keys into data encrypted under the same public key. Based one the transformed data, servers can perform a secure machine learning algorithm without leaning any private information. Our scheme can be applied in real-word collaborated learning, and we expect to improve the efficiency and give a comprehensive application in further work.

6 Acknowledgments

This work was supported by Guangzhou scholars project for universities of Guangzhou (No. 1201561613)

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. Mcmahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proc. of the 2016 ACM Sigsac Conference on Computer and Communications Security (CCS'16), Vienna, Austria*, pages 308–318. ACM, October 2016.

[2] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM journal on Computing*, 13(4):850–864, 1984.

[3] E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In *Proc. of the 9th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'03), Taipei, Taiwan*, volume 2894 of *Lecture Notes in Computer Science*, pages 37–54. Springer, Berlin, Heidelberg, November – December 2003.

[4] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proc. of the 33rd International Conference on International Conference on Machine Learning (ICML'16), New York, New York, USA*, volume 48, pages 201–210, June 2016.

[5] T. Graepel, K. Lauter, and M. Naehrig. Ml confidential: Machine learning on encrypted data. In *Information Security and Cryptology - the 15th International Conference on Information Security and Cryptology (ICISC'12), Seoul, Korea, Revised Selected Papers*, volume 7839 of *Lecture Notes in Computer Science*, pages 1–21. Springer, Springer, Berlin, Heidelberg 2012.

[6] E. Hesamifard, H. Takabi, and M. Ghasemi. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*, 2017.

[7] P. Li, J. Li, Z. Huang, C. Z. Gao, W. B. Chen, and K. Chen. Privacy-preserving outsourced classification in cloud computing. *Cluster Computing*, 21(1):1–10, 2017.

[8] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76–85, 2017.

[9] T. Li, Z. Huang, P. Li, Z. Liu, and C. Jia. Outsourced privacy-preserving classification service over encrypted data. *Journal of Network & Computer Applications*, 106:100–110, 2018.

[10] T. Li, J. Li, Z. Liu, P. Li, and C. Jia. Differentially private naive bayes learning over multiple data sources. *Information Sciences*, 444:89–104, 2018.

[11] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *Proc. of the 38th IEEE Symposium on Security and Privacy (S&P'17), San Jose, CA, USA*, pages 19–38. IEEE, May 2017.

[12] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics & Security*, 13(5):1333–1345, 2018.

[13] J. Yuan and S. Yu. Privacy preserving back-propagation neural network learning made practical with cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):212–221, 2014.

[14] Q. Zhang, L. T. Yang, and Z. Chen. Privacy preserving deep computation model on cloud for big data feature learning. *IEEE Transactions on Computers*, 65(5):1351–1362, 2016.

Author Biography

Ping Li received the M.S. and Ph.D. degree in Applied Mathematics from Sun Yat-sen University in 2011 and 2016, respectively. She is currently a postdoctoral researcher under supervisor Jin Li in School of Computer Science of Guangzhou University. Her current research interests include cryptography, privacy-preserving and cloud computing.

Hongyang Yan received her M.S. degrees from School of Mathematics and Information Science, Guangzhou University in 2016. Currently, she is a Ph.D. student in Nankai University. Her research interests include secure access control such as attribute-based cryptography and identity-based cryptography, IoT security.

Chong-zhi Gao received his Ph.D. degree in Applied Mathematics from Sun Yat-sen University in 2004. Currently, he is a professor at the School of Computer Science and Educational Software, Guangzhou University. His research interests include cryptography and privacy in machine learning.

Yu Wang received his Ph.D. in Computer Science from Deakin University, Australia. He is currently an associate professor with the School of Computer Science, Guangzhou University, China. His research interests include network traffic analysis, mobile networks, social networks, and cyber security.

Liaoliang Jiang received the B.S. degree in Information Security from Jiangxi University of Science and Technology, China, in 2014. Her research interests include applied cryptography and security in cloud computing.

Yuefang Huang received her B.S. degree in Mineral Processing from Northeast University in 2016. Currently, she is a master student in School of Computer Science of Guangzhou University. Her research interests include IoT security, secure access control.