Hu, Rui; Ding, Wenxiu; Yan, Zheng

Privacy-preserving Computation over Encrypted Vectors

Published in:
2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings

DOI:
10.1109/GLOBECOM42002.2020.9322329

Published: 01/12/2020

Please cite the original version:

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
algorithms have higher efficiency and practicality\cite{9}, which are

FHE algorithms, Partially Homomorphic Encryption (PHE) have high computation and communication overhead. Compared to multiplication operations over ciphertext\cite{6-8}, they introduce processing. Though Fully Homomorphic Encryption (FHE) their data in an encrypted form, which complicates data others\cite{5}. To avoid data privacy leakage, cloud users outsource their processed with the help of cloud computing. However, the cloud devalues, big data generated in IoT must be analyzed and and machine learning. Due to the limited resources of physical analyzing big data in IoT through methods such as data mining and make appropriate decisions by realizing physical entities. Owing to privacy concern, users are inclined to upload user data exposed and possibly be accessed by unauthorized amounts of data to a cloud server for storage and analysis, which are generated and transmitted in IoT\cite{1-4}. It helps to understand to provide various intelligent services and large amounts of data

 project under grant 2018 TD-007, and the 111 project under grant B16037, the Academy of Finland under Grants 308087, 314203 and JB191504, the Fundamental Research Funds for the Central Universities under Grant 335262, the National Postdoctoral Program for Innovative Talents under Foundation under grant 2018 M633461, the Shaanxi Innovation Team, and the Foundation under Grant 61802293 and 61672410, the Foundation of China under Grants 61802293 and 61672410, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents under Foundation under grant 2018 M633461, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents

I. INTRODUCTION

The Internet of Things (IoT) is widely applied in many areas

Corresponding author: wxding@xidian.edu.cn

Rui Hu *, Wenxiao Ding *, and Zhenhong Yan * †

We briefly review related work about dot-product of vectors, specifically, our contributions are summarized as below.

Vectors

To access disease diagnosis of a patient.

than ones specified user in an IoT system. For example, several fusion or analysis results always need to be accessed by more specific entity to decrypt a final encrypted result. However, data schemes are single-user systems\cite{14}, which allows only a single

Second, encryption restricts flexible access control over encryption schemes, and further show their efficiency and practicality of availing in the last section.

Finally, conclusion is followed by the notations and preliminaries in Section III. We describe our proposed schemes in Section IV. Section V gives comparisons with existing work. Finally, conclusion is

In the existing work, less attention is paid to the privacy-preserving computing over encrypted data. Specifically, our contributions are summarized as below.

• We analyze the security and complexity of our proposed

• We further extend our proposed dot-product schemes to

• We apply Attribute-Based Encryption (ABE) and

• We achieve secure outsourced computation of dot

In this paper, before designing a privacy-preserving SVM

for some machine learning algorithms.

Key words — cloud computing, privacy preserving, encrypted

The work is supported in part by the National Natural Science Foundation of China under Grants 61802293 and 61672410, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents under Foundation under grant 2018 M633461, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents

The work is supported in part by the National Natural Science Foundation of China under Grants 61802293 and 61672410, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents under Foundation under grant 2018 M633461, the Shaanxi Innovation Team, and the National Postdoctoral Program for Innovative Talents

- 123 -
• \( (\cdot, \cdot) \rightarrow \cdot \). It takes input message, and applies to the encrypted dot product of two encrypted vectors, which can be considered.

In our previous work \([22, 23]\), several basic operations were carried out, including Diffie-Hellman key of two servers. In addition, all of the above operations of two encrypted vectors and their results are not suitable for preserving hyperplane decision algorithms, it cannot support the secured dot product protocol and applied to achieve privacy-secured SVM algorithm. Although Bost et al. \([21]\) proposed a limited application and are difficult to be directly applied to data. The proposed scheme uses differential privacy technology efficiently privacy-preserving dot product scheme for mobile big data based on homomorphic encryption. Hu et al. \([19]\) proposed an interactively, and further complete secure vector computation oblivious transfer operations to exchange middle results of the original vectors from each other, and then perform multiple prime number and large integer to hide each element of the data and the practicality of dot product scheme. Wang et al. \([18]\) applied homomorphic encryption to design a shared history of the data and the practicability of dot-product scheme. To enhance the privacy of the input vector and the correctness of the dot product of two vectors by using oblivious transfer \([16]\) and research on dot product computation. We mainly use following several algorithms of CP-ABE in this paper.

Therefore, the two parties in the protocol by Ioannidis et al. \([17]\) ensures that only one computational dot product at the same time. A two-party protocol has the following features. In addition, HRB has the following features.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (\cdot) )</td>
<td>The length of ( (\cdot) )</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>Raw k-dimensional vectors;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>Encrypted access key;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The decryption key in ABE system;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The public key in ABE system;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The public parameter based on keys of DSP and public system parameters: generator, a large integer and the key pair of entity;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The private key of entity;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The key pair for access control;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The k-dimensional access structure, and the public parameter ( (\cdot) ) to get.</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The set up algorithm generates three keys for access control;</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The encryption algorithm generates the dot product of original vectors ( (\cdot) ).</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The partial decryption with ( (\cdot) ): DSP transfer ( (\cdot) ) to get.</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>User can encrypt ( (\cdot) ) with secret key.</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The decryption of the input message, and under the same access structure, it can get:</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The public key and a random number to encrypt ( (\cdot) ).</td>
</tr>
<tr>
<td>( (\cdot) )</td>
<td>The decryption with ( (\cdot) ): DSP transfer ( (\cdot) ) to get.</td>
</tr>
</tbody>
</table>

III. NOTATIONS AND PRELIMINARIES

The notations used in this paper are summarized in Table I.
In this section, we first give the system model and security model, followed by two detailed product schemes and a privacy-preserving SVM prediction algorithm.

A. System and Security Model

1) System Model: The system is composed of five types of entities as shown in Fig. 1: a) Data Providers (DPs) collect generated data and upload them to cloud for efficient storage and computation; b) Besides computational services, Computation Party (CP) and Data Service Provider (DSP) are responsible for secure data storage and access control, respectively; c) Data Requesters (DRs) as data consumers who acquire the processing result; d) Authority is responsible for key management.

2) Security Model: In our system, DSP and CP are regarded as semi-trusted, which act honestly and would never collude with each other. Except fully trusted Authority, other servers are curious about users' data but follow the design of system protocols strictly.

B. Product Computation for a Targeted Data Requester

The first scheme named Privacy-preserving Dot Product for Single-user (PPDPS) aims to obtain the dot product from two encrypted vectors for a specified data requester. And its processing procedure is shown in Fig. 2.

Step 1 (System Setup): The system calls algorithm to generate public parameters. Then each entity generates their key pairs by using public parameters.

Step 2 (Data Upload @ DPs): DPs recall to encrypt two $k$-dimensional vectors as $\mathbf{a}$ and $\mathbf{b}$, where $\mathbf{a} = (a_1, \ldots, a_k)$ and $\mathbf{b} = (b_1, \ldots, b_k)$. To avoid data flow, the length of each element in each vector should satisfy $(a_i^4) < (b_i^4)$ and $(b_i^4) < (a_i^4)$.

Step 3 (Data Preparation @ DSP): DSP chooses two random numbers as $\mathbf{r}$ and $\mathbf{s}$ where $r_1, \ldots, r_k \in [1, \frac{4}{k}]$ and $s_1, \ldots, s_k \in [1, \frac{4}{k}]$. Preprocess encrypted vectors to get $\mathbf{a}'$ and $\mathbf{b}'$ so as to conceal each raw vector from CP. Then DSP calls to transform $\mathbf{a}'$ and $\mathbf{b}'$ into another two encrypted vectors $\mathbf{a}''$ and $\mathbf{b}''$. Finally, DSP sends them to CP.

Step 4 (Data Process @ CP): Upon receiving encrypted vectors from DSP, CP obtains two vectors $\mathbf{a}'$ and $\mathbf{b}'$ by applying $\mathbf{a} = \mathbf{a}' \cdot \mathbf{b}' = \mathbf{a}'' \cdot \mathbf{b}''$. Then it performs dot product operation on masked vectors to get a middle result. CP further encrypts the result as $\mathbf{c}'$ with $\mathbf{e}'$ and sends it to DSP.

Step 5 (Additional Process @ DSP): DSP removes the mask from received ciphertext and get $\mathbf{a}''$.

Step 6 (Data Access @ DR): Upon receiving the encrypted computational result from DSP, the DR with corresponding secret key can decrypt the ciphertext $\mathbf{a}''$ to get the dot product of two original vectors.

C. Product Computation for Multiple Data Requesters

The second scheme named Privacy-preserving Dot Product for Multiple-user (PPDPM) introduces CP-ABE to enable flexible access control over computational result. And each entity processes encrypted vectors as shown in Fig. 3. We ignore the same previous three steps as PPDPS scheme and introduce PPDPM scheme from the fourth step as below.

Step 4 (Data Process @ CP): Upon receiving encrypted vectors $\mathbf{a}'$ and $\mathbf{b}'$ from DSP, CP first calls
random numbers and preprocessed encrypted data to get masked algorithm to obtain another ciphertext [],

partial key to generate a key pair, and sends it to CP.

and sets a random number as . DSP further masks from received ciphertext. Then it chooses a partial key product. Finally, CP calls to encrypt the middle computation on the two masked vectors to get a masked dot D. Privacy-Preserving SVM Prediction Algorithm [] to get the dot product of original vector.

a secret key from the authority and gets by calling cooperative process of DSP and CP is described as follows.

signature acquisition scheme in our previous work [22]. The privacy-preserving SVM algorithm with the assistance of the model parameters (, ) from DPs, DSP chooses two further gets encrypted access key through homomorphism ciphertext to obtain by using . Then it chooses a uses to encrypt as []. In addition, CP invokes conceal the ciphertext to get another ciphertext [] and operations on plaintexts to get a middle result, where to get masked plaintexts and performs dot product and addition operations, and returns the ciphertext to DSP.

=t ∗ ∗ . Then, CP encrypts as [] by .

After receiving an encrypted input vector [] and encrypted value [], DSP removes the mask from received ciphertext to get plaintexts and decrypts them to acquire raw information.

Upon receiving ciphertexts from DSP, CP first calls and retrieves the original vectors. And it chooses a random number as . Due to the security of HR, we can know can not obtain any information about original vectors.

The views of are merely the same elements in a vector, it can not get the real plaintext. Thus, final result due to the security of HR, the random numbers before encrypting the vector to eliminate the situation where the input vector is classified as negative.

Theorem 1. PPDPS can securely obtain the dot product theorem 2. PPDPM can securely obtain the dot product.

The security proof of PPDPM is a bit different from PPDPS, but its security can be ensured by HR and CP-ABE.

The data requester who satisfies the access policy can obtain any valid information.

V. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

Theorem 3. PPDPM can securely obtain the dot product.

Theorem 4. PPDPM can securely obtain the dot product.

Theorem 5. PPDPM can securely obtain the dot product.

Theorem 6. PPDPM can securely obtain the dot product.

Theorem 7. PPDPM can securely obtain the dot product.

Theorem 8. PPDPM can securely obtain the dot product.

Theorem 9. PPDPM can securely obtain the dot product.

Theorem 10. PPDPM can securely obtain the dot product.

Theorem 11. PPDPM can securely obtain the dot product.

Theorem 12. PPDPM can securely obtain the dot product.

Theorem 13. PPDPM can securely obtain the dot product.

Theorem 14. PPDPM can securely obtain the dot product.

Theorem 15. PPDPM can securely obtain the dot product.

Theorem 16. PPDPM can securely obtain the dot product.

Theorem 17. PPDPM can securely obtain the dot product.

Theorem 18. PPDPM can securely obtain the dot product.

Theorem 19. PPDPM can securely obtain the dot product.

Theorem 20. PPDPM can securely obtain the dot product.

Theorem 21. PPDPM can securely obtain the dot product.

Theorem 22. PPDPM can securely obtain the dot product.

Theorem 23. PPDPM can securely obtain the dot product.

Theorem 24. PPDPM can securely obtain the dot product.

Theorem 25. PPDPM can securely obtain the dot product.

Theorem 26. PPDPM can securely obtain the dot product.

Theorem 27. PPDPM can securely obtain the dot product.

Theorem 28. PPDPM can securely obtain the dot product.

Theorem 29. PPDPM can securely obtain the dot product.

Theorem 30. PPDPM can securely obtain the dot product.

Theorem 31. PPDPM can securely obtain the dot product.

Theorem 32. PPDPM can securely obtain the dot product.

Theorem 33. PPDPM can securely obtain the dot product.

Theorem 34. PPDPM can securely obtain the dot product.

Theorem 35. PPDPM can securely obtain the dot product.

Theorem 36. PPDPM can securely obtain the dot product.

Theorem 37. PPDPM can securely obtain the dot product.

Theorem 38. PPDPM can securely obtain the dot product.

Theorem 39. PPDPM can securely obtain the dot product.

Theorem 40. PPDPM can securely obtain the dot product.

Theorem 41. PPDPM can securely obtain the dot product.

Theorem 42. PPDPM can securely obtain the dot product.

Theorem 43. PPDPM can securely obtain the dot product.

Theorem 44. PPDPM can securely obtain the dot product.

Theorem 45. PPDPM can securely obtain the dot product.

Theorem 46. PPDPM can securely obtain the dot product.

Theorem 47. PPDPM can securely obtain the dot product.

Theorem 48. PPDPM can securely obtain the dot product.

Theorem 49. PPDPM can securely obtain the dot product.

Theorem 50. PPDPM can securely obtain the dot product.
2048 bits, the processing time of PPDP and PPDPM for 20 the cost in each step of the m. With the length of from 512 to tested in our previous work [23]. Hence, we focus on introducing CP-ABE and adding two processing steps in the and CP consumes 350 ms in the PPDP scheme. Due to section. When the length of is 2048 bits, DSP costs 950 ms is most time-consuming in both PPDP and PPDPM schemes, takes at most 300 ms. The computation of DSP in Step 3 (DSP3) in Table.

dimension vectors and the complexity of each entity are shown protocol schemes. We suppose that DPs provide two k-decryption, we analyze the complexity of our proposed dot attributes should be satisfied in access policy for successful attributes are embedded in access structure and that at most ϑ

B. Performance Evaluation

Test 1: Efficiency of PPDP and PPDPM

The efficiency of CP-ABE for access control has been

T A B L E II. COMPUTATIONAL COMPLEXITY ANALYSIS

<table>
<thead>
<tr>
<th>Entity</th>
<th>Scheme</th>
<th>Computations</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPDP</td>
<td>(6 + 2) ( \times ) ( \theta )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PPDPM</td>
<td>(6 + 5) ( \times ) ( \theta )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) ( \theta )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) + (2 ( \theta ) + 1) + ( \theta ) ( h ) (( \theta ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) + (2 ( \theta ) + 1) + ( \theta ) ( h ) (( \theta ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) + (2 ( \theta ) + 1) + ( \theta ) ( h ) (( \theta ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) + (2 ( \theta ) + 1) + ( \theta ) ( h ) (( \theta ))</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2 + 6) ( \times ) + 2 ( \times ) + (( \theta ) - 1) ( \times ) + (2 ( \theta ) + 1) + ( \theta ) ( h ) (( \theta ))</td>
<td></td>
</tr>
</tbody>
</table>

C. Computational Complexity Analysis

In this test, we simulated our proposed schemes and tested

Fig. 4. Execution time of PPDP

Fig. 5. Execution time of PPDPM

Table.

In sum, most of computation overhead are undertaken

As discussed in Section 3, most existing works cannot

Fig. 6. The scalability of secure SVM prediction algorithm

Fig. 7 displays the scalability of our secure SVM prediction algorithm with different lengths of integers still taken by DSP and CP, which benefits DPs and DRs with

Through simulation of privacy-preserving SVM prediction algorithm

Test 2: Efficiency of privacy-preserving SVM scheme

Test 3: Efficiency comparison with existing work

In summary, most existing works cannot achieve the computation of encrypted vectors. Since dot

As discussed, most existing works cannot still take

As discussed in Section 3, most existing works cannot still take

Fig. 4. Execution time of PPDP

Fig. 5. Execution time of PPDPM

Table.

In sum, most of computation overhead are undertaken

As discussed in Section 3, most existing works cannot

Fig. 6. The scalability of secure SVM prediction algorithm

Fig. 7 displays the scalability of our secure SVM prediction algorithm with different lengths of integers still taken by DSP and CP, which benefits DPs and DRs with

Through simulation of privacy-preserving SVM prediction algorithm

Test 2: Efficiency of privacy-preserving SVM scheme

Test 3: Efficiency comparison with existing work

In summary, most existing works cannot achieve the computation of encrypted vectors. Since dot
Paper has certain advantages in terms of efficiency. As shown in Fig. 8, the cost of CP is the same in our 300, the cost of DSP and CP in PPDP and previous work [22].

[References]:
[1] Wullianallur Raghupathi, and Viju Raghupathi, "Big data analytics in which is presented in Table III.

VI. CONCLUSION

REFERENCES

TABLE III. THE COMPARISON OF OUR INVENTION AND EXISTING WORKS

Ref. [21] Homomorphic encryption Plaintext Client-Server Y Y
Ref. [17] Random matrix perturbation Plaintext Two-party N N
Ref. [15] Oblivious transfer Plaintext Two-party N N
Ref. [16] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-not-anything disclosure of Reference Technologies or Methods Type of Input Data Application Scenarios PP FAC
Ref. [22] W. Ding, Z. Yan, and R. H. Deng, "Privacy-preserving data processing..."...