DATA ENCRYPTION STRATEGY IN CLOUD STORAGE

[1]R.Rajani,Professor,Dept of MCA,Narayana Engineering College,Nellore
[2]R.SaiPriya,Student,Dept of MCA,Narayana Engineering College,Nellore
[3] Y.Deva Kumar,Student,Dept of MCA,Narayana Engineering College,Nellore

Abstract: *Privacy has become a considerable issue when the applications of big data are dramatically growing in cloud computing. The benefits of the implementation for these emerging technologies have improved or changed service models and improve application performances in various perspectives. However, the remarkably growing volume of data sizes has also resulted in many challenges in practice. The execution time of the data encryption is one of the serious issues during the data processing and transmissions. Many current applications abandon data encryptions in order to reach an adoptive performance level companioning with privacy concerns. In this paper, we concentrate on privacy and propose a novel data encryption approach, which is called Dynamic Data Encryption Strategy (D2ES). my proposed approach aims to selectively encrypt data and use privacy classification methods under timing constraints. This approach is designed to maximize the privacy protection scope by using a selective encryption strategy within the required execution time requirements. The performance of D2ES has been evaluated in our experiments, which provides the proof of the privacy enhancement.*

Index Terms - Privacy-preserving, data encryption strategy, big data, mobile cloud computing, cyber security.

1.INTRODUCTION

Introducing mobile cloud computing techniques has empowered numerous applications in people's life in recent years. Involving humans in the cloud computing and wireless connection loops becomes an alternation for information retrieval deriving from observing human's behaviors and interactivities over various social networks and mobile apps. Moreover, as an emerging technology, cloud computing has spread into countless fields so that many new service deployments are introduced to the public, such as mobile parallel computing and distributed scalable data storage. Penetrations of big data techniques have further enriched the channels of gaining information from the large volume of mobile apps' data across various platforms, domains, and systems. Being one of technical mainstreams has enabled big data to be widely applied in multiple industrial domains as well as explored in recent researches.

Despite many benefits of using mobile cloud computing, there are great concerns in protecting data owner's privacy during the communications on social networks or mobile apps. One of the privacy concerns is caused by unencrypted data transmissions due to the large volume of data. Considering an acceptable performance level, many applications abandon using cipher texts in mobile cloud data transmissions. This phenomenon can result in privacy leakage issues since plain texts are unchallenging for adversaries to capture information in a variety of ways, such as jamming, monitoring, and spoofing. This privacy issue is exigent because it faces to a contradiction between the security levels and performance that is usually attached to timing constraints.

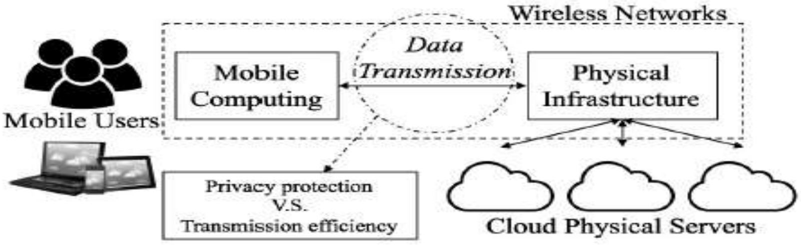


Fig. 1: High level architecture of mobile cloud computing illustrating the balance between privacy protection and transmission efficiency.

This paper addresses the issue of contradictions between data transmission efficiency and protection. To solve the problem, we propose a novel approach that selectively encrypts data in order to maximize the volume of encrypted data under the required timing constraints. The proposed model is called Dynamic Data Encryption Strategy (D2ES) model, which is designed to protect data owner's privacy at the highest level when using the applicable devices and networking facilities. Fig. 1 shows the high level architecture of mobile cloud with the illustrations of addressing the privacy protections.

Two major techniques used in D2ES are: (1) classifying data packages according to privacy level and (2) determine whether data packages can be encrypted under the timing constraints. I design and propose an algorithm, Dynamic Encryption Determination (DED) algorithm, which relies on the timing constraints and facilities' capacities to determine the data encryption alternatives. Detailed descriptions of D2ES are given in Section 3.

The main contributions of this work are threefold:

1) This work proposes a novel approach that selectively encrypts data packages to maximize the privacy protection level under timing constraints in big data. Two working modes are considered when creating the transmission strategy, including encryption and non-encryption modes.

2) The proposed algorithm offers an optimal solution providing the maximum value of total privacy weights. Two involved constraints are execution time and privacy levels.

3) The findings of this research provide big data-based solutions with an adaptive transmission approach focusing on protecting privacy. The proposed method can be also implemented in the distributed storages in cloud computing.

2 RELATED WORK

First, researches addressing the attacks in social networks have been paid attention by many scholars. Moreover, from the perspective of user controllability, securing efficient wireless communications is crucial in a high performance mobile cloud system. Furthermore, privacy concerns can be caused by various dimensions in mobile clouds. Untrustworthy data is the first aspect of creating privacy leakages that can be hardly perceived by users or service providers due to two main reasons. The first reason is that it is difficult to identify the collected data because of the low trustworthy. The other one is that adversaries do not distribute any identification information such that it is hard to generate threat alerts. In addition, the vulnerability detection is also an important aspect of preventing privacy leakage.

3 CONCEPTS AND THE PROPOSED APPROACH

3.1 Problem Definition

I describe the main research problem in this section. Definition 3.1 shows the identified research problem that is Maximum Data Package under Timing Constraints (MDPuTC) problem. Definition 3.1. Maximum Data Package Under Timing Constraints (MDPuTC) Problem: Inputs: data package types {Di}, the number of data for each data package type N_{Di}, execution time when encrypting data for each single data T^{e}_{Di} ,execution time without encryptions for each single data T^{n}_{Di} ,the privacy weight value for each data type W_{Di}. Outputs: strategy determining which data will be encrypted. The proposed problem is finding out the approach that can gain the maximum total privacy weight value under a given timing constraint. As illustrated in Definition 3.1, the main inputs include five variables. First, input data include a group of packages that are classified into different types, represented as a set {Di}. The number of data packages in each type D_i is represented as N_{Di} . Moreover, there are two kinds of execution modes, which include Operation with Encryptions (OwE) and Operation with Non-Encryption (OwNE). The execution time of each data package D_i in OwE mode is T^{e}_{Di}. Similarly, the execution time of each data package Di in OwNE mode is $T^{n}_{Di.}$ Furthermore, I introduce a parameter, Privacy Weight Value (PWV), for each data package type in order to calculate the beneficial acquisitions from encrypting data, represented as W_{Di}.

The meaning of PWV is a criterion showing security significance levels. The acquisitions of PWV values that categorize security issues into multiple levels can be gained by various approaches, such as scorecard sheet and security measurement category. In my proposed model, the PWV value represents the privacy importance for each data package. Therefore, the output is a encryption strategy that determines which data packages should be encrypted. Assume that the number of encrypted data packages for Di is N^{e}_{Di} . The object of my research problem is maximizing the sum of PWV values and the objective function is expressed in Eq. (1). In the function, we create a binary function S(i) to represent the selection. The encryption strategy is selected when S(i) = 1 and a non-encryption strategy is selected when S(i) = 0. Since unencrypted data packages do not earn any privacy weights, only encrypted data packages are counted in our model.

$$\text{Output} = MAX\left(\sum_{S(i)=1}\left(N^{e}_{Di} \times W_{Di}\right)\right) = p \qquad (1)$$

The condition is the total execution time is no longer than the required timing constraint Tc. The length of Tc must satisfy the following requirement, as shown in Eq. (2). The expression shows the minimum execution time of data operations, which excludes all encryptions.

$$T_c \geq \sum_{s(i)=0}\left(N_{Di} \times T^{n}D_i\right) \qquad (2)$$

After implementing D2ES approach, some data packages are selected to be encrypted. Configure that the encrypted data set is $\{D_i\}$ and the non-encrypted data set is {Dj} The total execution time can be gained by Eq (3):

$$T_{total} = \sum_{s(i)=1}\left(N^{e}_{D_k} \times T^{e}_{D_k}\right) + \sum_{s(i)=0}\left(N^{n}_{D_j} \times T^{n}_{D_j}\right) \qquad (3)$$

where $T_c \geqslant T_{total}$.

Identifying the critical problem is the fundamental of implementing D2ES model. The following section will explain the main mechanism of data alternatives in our model.

3.2 Dynamic Data Encryption Strategy (D2ES) Model

Based on the definitions given in Section 3.1, I present my D2ES model in this section. The crucial goal of D2ES model is solving the problem defined in Definition 3.1. There are mainly three phases forming the solution. Fig. 2 illustrates three crucial phases of D2ES model.

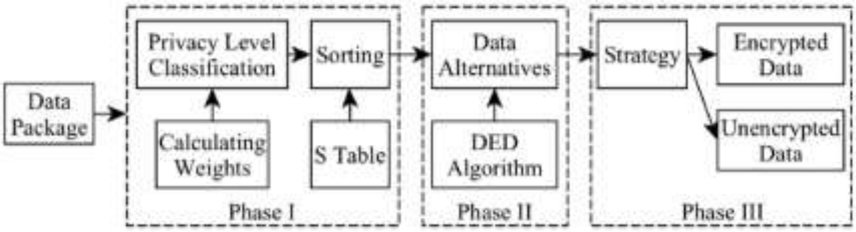


Fig. 2: Crucial phases of Dynamic Data Encryption Strategy (D2ES) Model

3.2.1 Phase I: Sorting by Weights

This is a preparation phase of the model. All data package types are sorted at this phase. The sorting operations consider both execution time and privacy protections; thus, two variables are involved, which are PWVs and the corresponding encryption execution time.

For each data package Di, the value used for sorting operations is defined as a Sorting Weight, denoted as a SDi , which can be obtained by Eq. (4).

$$S_{D_i} = W_{D_i}/T^e_{D_i} \qquad (4)$$

Definition 3.2. Paired Data: □ two data package type Di and Dj . □Di, if □ operating Di in plain texts needs a must encryption operation for Dj , the relation between Di and Dj is a Paired Data, represented as Di ↔Dj. Based on the definition of paired data, we propose a PMC mechanism to ensure that at least one data within the paired data have the encryption priority. The PMC definition is givenin Definition 3.3.

Definition 3.3. Pairs Matching Collision: Any two data Di and Dj matching the requirement of paired data Di ↔ Dj , the mechanism that can ensure at least one data, Di or Dj , are encrypted is defined as PMC mechanism. The deterministic process of finding out the paired data is a collision.

3.2.2 Phase II: Data Alternatives

This phase is the crucial step of selecting data packages for encryption operations. I propose the DED algorithm to accomplish this phase. S Table will be used for providing the reference of protection efficiencies. The operating principle is that data package with higher value of SDi has a higherlevel alternative priority than those data packages having lower values of SDi . There are a few sub-steps for selecting data packages.

First, a timing scope needs to be identified. The given timing constraint is Tc. Therefore, the timing scope is [0,Ts], in which the value of Ts can be gained from Eq. (5).

$$\text{Ts} = \text{Tc} - \textstyle\sum_{s(i)=0}(N_{Di} \times T^n D_{i)}) \qquad (5)$$

Next, data alternatives are executed. Each encrypted data package's execution time is T^e_{Di}. i first encrypt the data package with the highest SDi value. The operation will not be ended until two situations occur. The first situation is that all data packages are encrypted. The other situation is that the execution time T^e_{Di} is longer than the rest of the time.

Define the rest of the execution time is Tr, where Tr ≤ Ts.In our model, we calculate time Tr considering both execution time with executions and execution time without encryptions. Once the data package is selected to be encrypted, the execution time without encryption should be added to Tr. Assume that the selected data packages are {Ds}. Eq. (6) represents the formulas of calculating Tr.

$$\begin{aligned} T_r &= T_s - \sum_{s(i)=1}(N_{D_i} \times T^e_{D_i}) + \sum_{s(i)=0}(N_{D_i} \times T^n_{D_i}) \\ &= T_c - \sum_{s(i)=0}(N_{D_i} \times T^n_{D_i}) - \sum_{s(i)=1}(N_{D_i} \times T^e_{D_i}) + \sum_{s(i)=0}(N_{D_i} \times T^n_{D_i}) \\ &= T_c - \sum_{s(i)=1}(N_{D_i} \times T^e_{D_i}) \end{aligned} \qquad (6)$$

The data alternatives process ends when Tr is lower than any left data package's execution time with encryptions.

3.2.3 Phase III: Output

This phase mainly output an encryption plan deriving from the outcomes of Phase II. Those data with higher-level encryption priority will be selected for the encryptions under a certain constraints. The rest of data will not be encrypted such that plain texts operations are applied. In order to provide more concise presentation, Section 4 displays a motivational example.

4 MOTIVATIONAL EXAMPLE

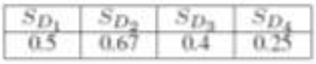
The application scenario is configured as follows: (1) There are 4 data package types, including D1,D2, D3, and D4. Each type has a certain amount of data packages and the execution time periods are distinct. (2) Timing constraint Tc is 25. All data packages need to be processed within 25-unit time. (3) Two working modes are included, M1 and M2. M1 is the mode with encryptions; M2 is the mode without encryptions. The objective is finding out the strategy that can earn the highest total PWV by choosing a set of data packages for encryptions. The retrievals of PWVs depend on the mechanisms of data protections. The operating principle is that a higher level complexity of the data encryption will earn a higher PWV. Table 1 shows a mapping of data package types with the corresponding values. We name this table as M Table. For example, D1 has 3 data packages, which requires 5-unit time for encryptions while needing 1-unit time for non-encryption. The privacy weight value of D1 is 2.5 that is higher than any other types.

TABLE 1: M Table: Table mapping data types, amount, and working modes. DPT (Data Package Type); M1: Mode1 (with encryptions); M2: Mode 2 (without encryptions)

DPT	Amount	M_1		M_2	
		$T^e_{D_i}$	W_{D_i}	$T^n_{D_i}$	W_{D_i}
D_1	3	5	2.5	1	0
D_2	4	3	2	0.5	0
D_3	2	3	1.2	0.5	0
D_4	2	4	1	1	0

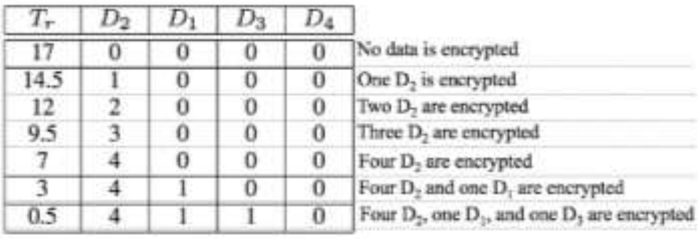
First, we calculate values of SDi for the sorting purpose. In this case, we list results in Table 2. Meanwhile, Ts= 17, which derives from (25 - (1×3 + 0.5 ×4 + 0.5 ×2 + 1 ×2)).

TABLE 2: S Table: Table for SDi Values



S_{D_1}	S_{D_2}	S_{D_3}	S_{D_4}
0.5	0.67	0.4	0.25

According to the results shown in Table 2, the priority sequence is SD2 → SD1→ SD3 →SD4 . Moreover, we use our DED algorithm to produce a table mapping the data alternatives. Fig. 3 shows the data alternatives process and the results. As shown in the table, we generate the following data encryption strategy: encrypt 4 D2, encrypt 1 D1,encrypt 1 D3, and do not encrypt D4. The value of P is 11.7. A simple comparison is completed between D2ES and greedy algorithm. Using greedy algorithm can generate a strategy plan as follows: encrypt 3 D1, encrypt 1 D2, do not encrypt D3, and do not encrypt D4. The P value is 9.5. Therefore, our approach's P value is 23.2% higher than greedy algorithm.



T_r	D_2	D_1	D_3	D_4	
17	0	0	0	0	No data is encrypted
14.5	1	0	0	0	One D_2 is encrypted
12	2	0	0	0	Two D_2 are encrypted
9.5	3	0	0	0	Three D_2 are encrypted
7	4	0	0	0	Four D_2 are encrypted
3	4	1	0	0	Four D_2 and one D_1 are encrypted
0.5	4	1	1	0	Four D_2, one D_1, and one D_3 are encrypted

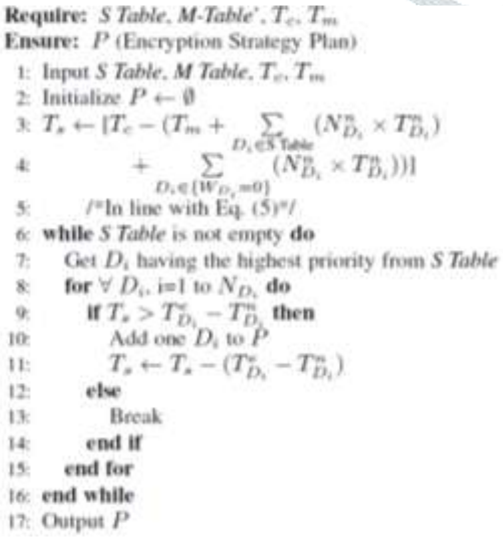
Fig. 3: Data alternative process.

5 ALGORITHMS

The main steps of DED algorithm are illustrated as follows:
1) Input timing constraint Tc and two tables S Table and M Table. Initialize a strategy plan dataset P as an empty set. Initialize a variable end Flag and assign a False value to it.
2) We use a While loop to create the strategy, which relies on the available time. We estimate whether the data packages should be encrypted one by one in a sequence depending on the priority weights. The data package having a higher-level priority will be

determined first. As shown in Algorithm 5.1, Tm refers to the shortest execution time, which can be considered a total execution time without encryptions.
3) Keep updating the execution time scope Ts. Each data package's non-encryption time needs to be added if the encryption time mode is selected during the process for updating the execution time scope.
4) Add the data package to the set P when the value of Ts is greater than 0 and the encryption time of certain data package is no longer than Ts. This process follows the principle that higher priority weight goes first.
5) End While loop when there is no data package matching the condition any more.

Algorithm 5.1 Dynamic Encryption Determination (DED) algorithm



```
Require: S Table, M-Table', $T_c$, $T_m$
Ensure: $P$ (Encryption Strategy Plan)
Input S Table, M Table, $T_c$, $T_m$
Initialize $P \leftarrow \emptyset$
$T_s \leftarrow \lfloor T_c - (T_m + \sum_{D_i \in S\ Table} (N^n_{D_i} \times T^n_{D_i})$
        $+ \sum_{D_i \in \{W_{D_i}=0\}} (N^n_{D_i} \times T^n_{D_i}))\rfloor$
    /*In line with Eq. (5)*/
while S Table is not empty do
  Get $D_i$ having the highest priority from S Table
  for $\forall D_i$, i=1 to $N_{D_i}$ do
    if $T_s > T^e_{D_i} - T^n_{D_i}$ then
      Add one $D_i$ to $P$
      $T_s \leftarrow T_s - (T^e_{D_i} - T^n_{D_i})$
    else
      Break
    end if
  end for
end while
Output $P$
```

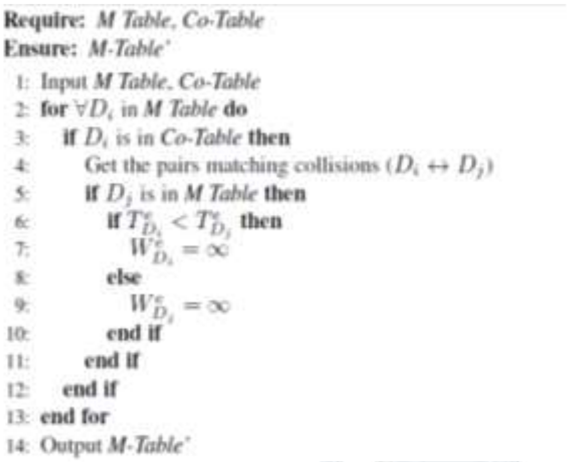
6) Output the set P that consists of a set of data packages Di. Encrypt all data packages in p.

5.2 Weight Modelization (WM) Algorithm

The main phases of Algorithm 5.2 include:
1) Input the original mapping table M Table and the predefined Co-Table.
2) For all data Di in M Table, determine whether data Di is involved in table Co-Table. Find out the paired data Dj when Di is in Co-Table and this pairing process is represented as Di ↔Dj . The rule of pairing data refers

to Definition 3.3.
3) Judge whether data Dj is in the mapping table M Table in order to determine whether the weight value needs to be modified. The weight value needs to be changed when Dj is in M Table.
4) Compare the encryption time lengths between Di and Dj . Assign an infinity value to D^{e}_{Di} when the execution time Di is shorter than $D^{"}_{Js}$. Otherwise, assign an infinity valuetoD $^{e}_{DJ}$, which means that we consider this data the highest encryption priority.

Algorithm 5.2 Weight Modelization (WM) Algorithm



```
Require: M Table, Co-Table
Ensure: M-Table'
Input M Table, Co-Table
for $\forall D_i$ in M Table do
  if $D_i$ is in Co-Table then
    Get the pairs matching collisions ($D_i \leftrightarrow D_j$)
    if $D_j$ is in M Table then
      if $T^e_{D_i} < T^e_{D_j}$ then
        $W^e_{D_i} = \infty$
      else
        $W^e_{D_j} = \infty$
      end if
    end if
  end if
end for
Output M-Table'
```

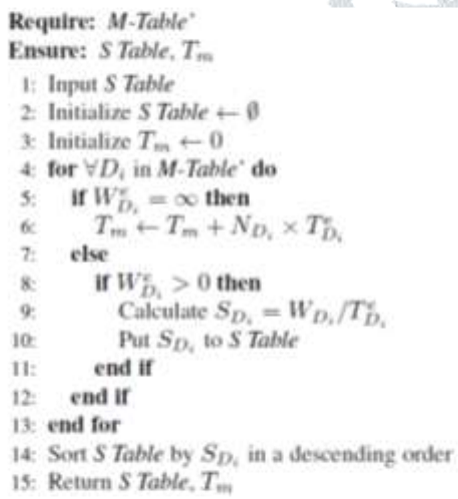
5) After all data are operated and updated, output the modified table M-Table'.

The time complexity of WM algorithm is T(n) = O(n).As a precedent work of the main algorithm, WM algorithm increase theprivacy protection level by using a secure mechanism. The next section describes the method for generating S Table.

5.3 S Table Generation (STG) Algorithm

The crucial steps of STG algorithm are described as follows:
1) Input table S Table and initialize the table by assigning an empty value. Initialize a variable Tm and assign a 0 value to it.
2) For all data Di in table M-Table', entry a FOR loop. For each data Di in the loop, calculate and update the Tm value if the corresponding W^{e}_{Di} 's value has been assigned as an infinity. The method is Tm ← Tm + $N_{Di} \times T^{e} D_i$
3) Otherwise, we need to calculate SDi by SDi = $W_{Di} / T^{e} D_i$ when the corresponding WeDi 's value is larger than 0. Add the gained SDi to the table STable.
4) End the FOR loop when all data Di are operated.
5) Sort all SDi in the updated STable in a descending order.Then, output both STable and Tm.

Algorithm 5.3 S Table Generation (STG) Algorithm



```
Require: M-Table'
Ensure: S Table, $T_m$
Input S Table
Initialize S Table $\leftarrow \emptyset$
Initialize $T_m \leftarrow 0$
for $\forall D_i$ in M-Table' do
  if $W^e_{D_i} = \infty$ then
    $T_m \leftarrow T_m + N_{D_i} \times T^e_{D_i}$
  else
    if $W^e_{D_i} > 0$ then
      Calculate $S_{D_i} = W_{D_i} / T^e_{D_i}$
      Put $S_{D_i}$ to S Table
    end if
  end if
end for
Sort S Table by $S_{D_i}$ in a descending order
Return S Table, $T_m$
```

6 Experimental Results

I illustrated a few experimental results in this section. Fig. 4 and 5 represented a group of comparison results concerning the total PWV and the required execution time between our D2ES and optimal solutions under Setting 1, respectively. Fig. 4 displayed that our approach had a similar performance to the optimal solutions in acquiring total privacy weight and Fig. 5 displayed the differences of the estimated execution time. Figures illustrated the results from the same experiment rounds. Most P values obtained from D2ES were close to the optimal results that were obtained by BF algorithm.

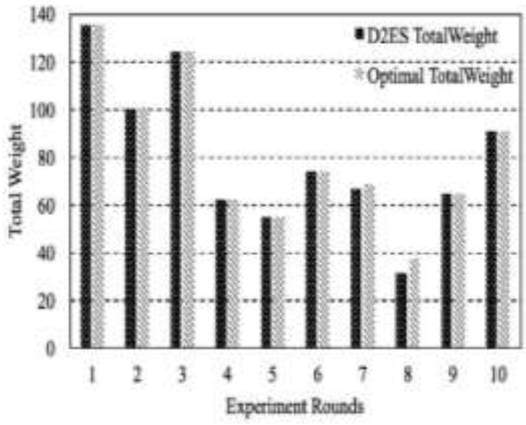


Fig. 4: Comparisons of total privacy weights between D2ES and optimal solution under Setting 1.

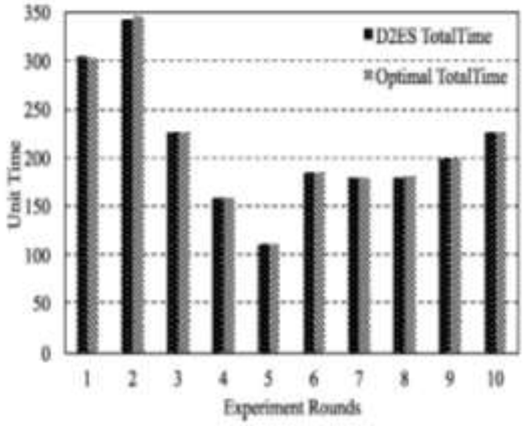


Fig. 5: Comparisons of total required execution time between D2ES and optimal solution pairing with Fig. 4 under Setting 1.

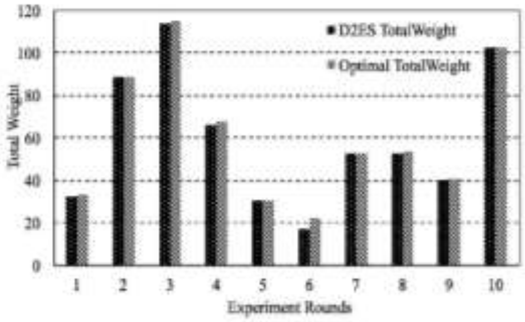


Fig. 6: Comparisons of total privacy weights between D2ES and optimal solutions under Setting 2

Moreover, Fig. 6 and 7 showed another group of comparison results addressing the P value and total required execution time between D2ES and optimal solutions under Setting 2. Fig. 6 represented a close performance between D2ES and optimal solutions in obtaining the total privacy weight. Fig.7 represented that our approach needed a shorter execution time than that of BF, which was aligned

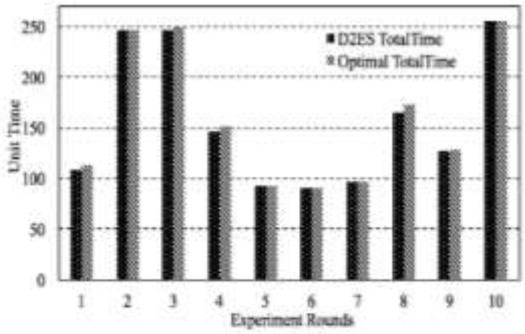


Fig. 7: Comparisons of total required execution time between D2ES and optimal solutions pairing with Fig. 6 under Setting 2

with the experiment rounds in Fig. 6. The reason for a shorter execution time was that my approach could acquire less P value than the optimal solutions. A lower level P value could result in a shorter required execution time.

REFERENCES

[1] S. Yu, W. Zhou, S. Guo, and M. Guo. A feasible IP traceback framework through dynamic deterministic packet marking. IEEE Transactions on Computers, 65(5):1418–1427, 2016.

[2] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic. Malware propagation in large-scale networks. IEEE Transactions on Knowledge and Data Engineering, 27(1):170–179, 2015.

[3] S. Liu, Q. Qu, L. Chen, and L. Ni. SMC: A practical schema for privacy-preserved data sharing over distributed data streams. IEEETransactions on Big Data, 1(2):68–81, 2015.

[4] S. Rho, A. Vasilakos, and W. Chen. Cyber physical systems technologies and applications. Future Generation Computer Systems, 56:436–437, 2016.

[5] L. Wu, K. Wu, A. Sim, M. Churchill, J. Choi, A. Stathopoulos, C. Chang, and S. Klasky. Towards real-time detection and tracking of spatio-temporal features: Blob-filaments in fusion plasma. IEEE Transactions on Big Data, 2(3), 2016.

[6] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar. Dependable demand response management in the smart grid: A stackelberg game approach. IEEE Transactions on Smart Grid, 4(1):120–132, 2013.

[7] M. Qiu, M. Zhong, J. Li, K. Gai, and Z. Zong. Phase-change memory optimization for green cloud with genetic algorithm. IEEE Transactions on Computers, 64(12):3528–3540, 2015.

[8] H. Liu, H. Ning, Y. Zhang, Q. Xiong, and L. Yang. Role-dependent privacy preservation for secure V2G networks in the smart grid. IEEE Transactions on Information Forensics and Security, 9(2):208–220,2014.

[9] F. Tao, Y. Cheng, D. Xu, L. Zhang, and B. Li. CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. IEEE Transactions on Industrial Informatics, 10(2):1435–1442, 2014.

[10] G. Wu, H. Zhang, M. Qiu, Z. Ming, J. Li, and X. Qin. A decentralized approach for mining event correlations in distributed system monitoring. Journal of parallel and Distributed Computing, 73(3):330–340, 2013.

[11] S. Yu,W. Zhou, R. Doss, andW. Jia. Traceback of DDoS attacks using entropy variations. IEEE Transactions on Parallel and Distributed Systems, 22(3):412–425, 2011.

[12] Y. Li, W. Dai, Z. Ming, and M. Qiu. Privacy protection for preventing data over-collection in smart city. IEEE Transactions on Computers, 65:1339–1350, 2015.

[13] S. Yu,W. Zhou,W. Jia, S. Guo, Y. Xiang, and F. Tang. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE Transactions on Parallel and Distributed Systems, 23(6):1073– 1080, 2012.

[14] Y. Yu, M. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min. Identity-based remote data integrity checking with perfect