Mobile Health Application with Privacy Preserving Data Encryption Strategy for Big Data in Mobile Computing

[1]Santhosh S, [2]Sindhu Y Shirur, [3]Chaitra S, [4]Husna Banu

[1]Assoc. Professor, [2,3,4]M.tech, CSE, 4th SEM,

KIT, Tiptur-572201

Abstract-Mobile Health (mHealth) proposes health care delivering anytime and anywhere. It aims to answer several emerging problems in health services, including the increasing number of chronic diseases, high costs on national health services, and the need to provide direct access to health services, regardless of time and place. mHealth systems include the use of mobile devices and apps that interact with patients and caretakers. Privacy has become a considerable issue when the applications of big data are dramatically growing in cloud computing. 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. This paper concentrates on privacy and proposes a novel data encryption approach, which is called Dynamic Data Encryption Strategy (D2ES). The 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 in mobile health app by using a selective encryption strategy within the required execution time requirements. The performance of D2ES has been evaluated in the experiments, which provides the proof of the privacy enhancement.

Keywords: mhealth, Privacy-preserving, data encryption strategy, big data, mobile cloud computing, cybersecurity.

I. 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 humans' behaviours and interactivities over various social networks and mobile apps. Health telematics, also known as electronic health (eHealth), have offered patients major improvements in their lives by providing more accessible and affordable health care solutions. This is particularly true for patients that live in remote rural areas, travel constantly, are physically

incapacitated, elderly, or chronically ill. Telemedicine assumes the use of medical information, also known as electronic health records (EHRs), exchanged via electronic communications improving the patients' health status 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.

This paper addresses the issue of contradictions between data transmission efficiency and protection. To solve the problem, a novel approach that selectively encrypts data in order to maximize the volume of encrypted data under the required timing constraints is proposed. The proposed model is called Dynamic Data Encryption Strategy (D2ES) model, which is designed to protect data owners' privacy at the highest level IEEE Transactions on Big Data when using the applicable devices and networking facilities. The crucial issue is that most contemporary wireless transmissions carry plain-texts due to the workload volume and real-time service concerns. The implementation of big data further stops transmission from carrying cipher-texts. The target protection location is represented by the broken-line box in the figure, which depicts that the data transmissions between physical infrastructure and mobile computing in mobile cloud need to be protected. 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.

An algorithm, Dynamic Encryption Determination (DED) algorithm, which relies on the timing constraints and facilities' capacities to determine the data encryption alternatives, is proposed.

This paper is an extended work of research and prior work focused on the general data encryption strategy of big data in cloud systems. This paper, has extended the work by enriching the mechanism design for each specific mode phase. Two crucial terms are designed for implementing the data encryption strategy, which include Paired Data and Pairs Matching Collision. In addition, two crucial algorithms are proposed for

supporting the implementation of D2D algorithm, which are Weight Modelization (WM) Algorithm and S Table Generation (STG) Algorithm. These two new algorithms further identify the methods of identifying privacy values when making a determination on encrypting the input data. The research is significant for generating an adoptive solution to protecting data owners' privacy.

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.

II. SYSTEM MODEL

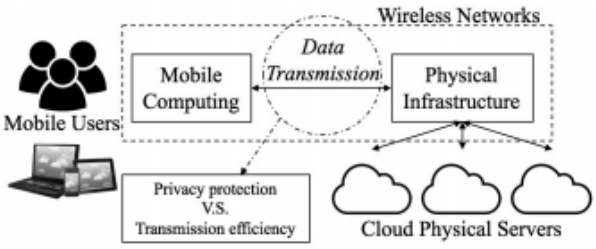

Fig. 1 High level architecture of mobile cloud computing illustrating the balance between privacy protection and transmission efficiency. The crucial issue is that most contemporary wireless transmissions carry plain-texts due to the workload volume and real-time service concerns. The implementation of big data further stops transmission from carrying cipher-texts. The target protection location is represented by the broken-line box in the figure, which depicts that the data transmissions between physical infrastructure and mobile computing in mobile cloud need to be protected.

III. PREVIOUS WORK

Mobile cloud is an inter-connective platform for mobile users, which supports information sharing among multiple parties across distinct infrastructure. Zhang et al. [17] proposed an approach named SCLPV for cloud-based Cyber Physical Social Systems (CPSS) to avoid malicious auditors. This approach concurrently provisioned certificateless public verification as well as resistance against malicious auditors for the purpose of verifying the integrity of outsourced data in CPSS. Wang et al. [18] focused on developing an approach offering a secure cloud system that could support privacy preserving public auditing. Moreover, from the perspective of user controllability, securing efficient wireless communications [19] is crucial in a high performance mobile cloud system. One research solved the problem by building up a two-dimensional paired connections over the Radio Frequency for Consumer Electronics (FR4CE) for both appliances and controllers, while users attempt to connect with appliances [20]. Another research also addressed usermachine interaction issues but in a different standpoint. The research argued that the significant hemisphere of protecting privacy is establishing an effective approach emphasizing both humans' involvements and system controls [21]. Both sides need to be matched and combined in order to accurately predict adversaries. Multi-channel communications [22] could be considered an alternative using various data protection methods for increasing the level of the privacy protection under different constraints. 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 [23]. The first reason is that it is difficult to identify the collected data

IV. PROPOSED METHODOLOGY

3.1 Problem Definition

We 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 NDi , 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 WDi . Outputs: a 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 Di is represented as NDi . 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 Di in OwE mode is T e Di . Similarly, the execution time of each data package Di in OwNE mode is T n Di . Furthermore, we introduce a parameter, Privacy Weight Value (PWV), for each data package type in order to calculate the beneficial acquisitions from encrypting data, represented as WDi . 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 [31], [32] and security measurement category [33]. In our 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 Ne Di . The object of our 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. Output = Max(X s(i)=1 (N e Di × WDi)) = P (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. Tc > X s(i)=0 (NDi × T n Di) (2) After implementing D2ES approach, some data packages are selected to be encrypted. Configure that the encrypted data set is {Dk} and the non-encrypted data set is {Dj} The total execution time can be gained by Eq. (3): Ttotal = X s(i)=1 (N e Dk × T e Dk) + X s(i)=0 (N n Dj × T n Dj) (3) where Tc > Ttotal. 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, we present our 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.

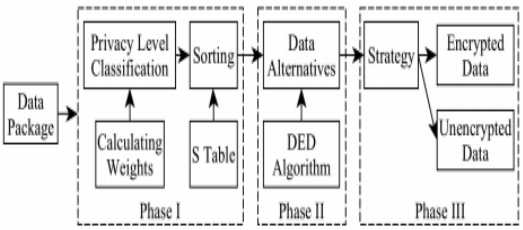

Fig. 2 Crucial phases of Dynamic Data Encryption Strategy (D2ES) Model

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).

$$SDi = WDi / Te\ Di \quad (4)$$

The expression in Eq. (4) shows the efficiency of protecting privacy. The sorting operation uses a descending order. The next step is to map all sorting results into a table that is called S Table. The values of the sorting results can determine the priority. Moreover, in order to improve the level of privacy protection, we introduce a Pairs Matching Collision (PMC) mechanism. This mechanism is designed to avoid the scenario when two plain texts can release users' privacy even though leaking each plain text will not be harmful. The operating principle of PMC mechanism is make sure two pre-defined pair data have at least one data encrypted. The paired data must contain privacy information when they are transmitted or operated in plain texts. Definition 3.2 provides the definition of paired data.

Definition 3.2. Paired Data**:** ∃ two data package type Di and Dj . ∀Di , if ∃ operating Di in plain texts needs a mustencryption 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 given in 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. We 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).

$$Ts = Tc - X\ s(i)=0\ (NDi \times T\ n\ Di\ ) \quad (5)$$

Next, data alternatives are executed. Each encrypted data package's execution time is T e Di . We 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 6 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.

$$Tr = Ts - X\ s(i)=1\ (NDs \times T\ e\ Ds\ ) + X\ s(i)=0\ (NDs \times T\ n\ Ds\ )$$

$$= Tc - P\ s(i)=0\ (NDi \times T\ nDi\ ) - P\ s(i)=1\ (NDs \times T\ eDs\ ) + P\ s(i)=0\ (NDs \times T\ nDs\ )$$

$$= Tc - X\ s(i)=1\ (NDs \times T\ e\ Ds\ ) \quad (6)$$

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

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.

V. SIMULATION/EXPERIMENTAL RESULTS

We present the main algorithm used in our D2ES model in this section, which include Dynamic Encryption Determination (DED) algorithm, S Table Generation (STG) algorithm, and Weight Modelization (WM) algorithm. DED algorithm is designed to dynamically select data packages that can be encrypted under certain conditions when considering both timing constraints and facilities' capacities. STG and WM algorithms are designed for supporting DED algorithm.

5.1 Dynamic Encryption Determination (DED) Algorithm

DED algorithm is designed to create the final privacy protection strategy corresponding with the timing constraints and security requirements. Inputs of DED algorithm include M Table, S Table, and Tc. Samples of M Table and S Table are given in Table 1 and 2. The output is the data encryption strategy plan P that directs which data packages need to be encrypted. The crucial part of this algorithm is calculating the remainder of the available time so that the encryption strategy can be determined. Algorithm 5.1 represents the pseudo codes of DED algorithm. 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 endFlag 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', Tc, Tm

Ensure: P (Encryption Strategy Plan)

1: Input S Table, M Table, Tc, Tm

2: Initialize P ← ∅

3: Ts ←[Tc − (Tm + P Di∈S Table (Nn Di × T n Di)

4: + P Di∈{WDi=0} (Nn Di × T n Di))]

5: /*In line with Eq. (5)*/

6: while S Table is not empty do

7: Get Di having the highest priority from S Table

8: for ∀ Di , i=1 to NDi do

9: if Ts > Te Di − T n Di then

10: Add one Di to P

11: Ts ← Ts − (T e Di − T n Di)

12: else 13: Break 14: end if15: end for16: end while 17: Output P

5.2 Weight Modelization (WM) Algorithm

The WM algorithm is developed for modifying M Table using weight values. The purpose of this algorithm is to check whether a data package is a must-encrypted objective, when considering the relations between packages. Thus, the pairs matching collisions (Definition 3.3) are applied in this algorithm in order to detect the paired data (Definition 3.2). Inputs include an M Table and a Co-Table. The output of this algorithm is a modified M Table, which is represented as an M-Table'. MTable' is an input for both Algorithms 5.1 and 5.3. Moreover, a Co-Table refers to a table mapping all paired data, which is pre-defined by security policies or developers. The Co-Table is used to manipulate pairs matching collisions. Algorithm 5.2 presents the pseudo codes of 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 De Di when the execution time Di is shorter than D0 j s. Otherwise, assign an infinity value to De 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'

1: Input M Table, Co-Table

2: for ∀Di in M Table do

3: if Di is in Co-Table then

4: Get the pairs matching collisions (Di ↔ Dj)

5: if Dj is in M Table then

6: if T e Di < Te Dj then

7: We Di = ∞

8: else

9: We Dj = ∞

10: end if

11: end if

12: end if

13: end for

14: Output M-Table'

5.3 S Table Generation (STG) Algorithm

STG algorithm is designed to generate an S Table that is one of the inputs of Algorithm 5.1. Input includes the modified M Table, MTable', that is the output of the Algorithm 5.2. Outputs include Stable and Tm. S Table is a table for sorting purposes. In general, Tm is a sum of execution time when all data are not encrypted.

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 We Di 's value has been assigned as an infinity. The method is Tm ← Tm + NDi × T e Di

3) Otherwise, we need to calculate SDi by SDi = WDi /Te Di when the corresponding We Di '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. The represented algorithms illustrate a fundamental mechanism used in our proposed D2ES model. There will be diverse implementations by adding parameters, conditions, or coeffi- cients, which are conditional on the application environment

Algorithm 5.3 S Table Generation (STG) Algorithm

Require: M-Table' Ensure: S Table, Tm

1: Input S Table

2: Initialize S Table ← ∅

3: Initialize Tm ← 0

4: for ∀Di in M-Table' do

5: if We Di = ∞ then

6: Tm ← Tm + NDi × T e Di

7: else

8: if We Di > 0 then

9: Calculate SDi = WDi /Te Di

10: Put SDi to S Table

11: end if

12: end if

13: end for

14: Sort S Table by SDi in a descending order

15: Return S Table, Tm

and practical demands. The following section explicates our experimental evaluations as well as partial experimental findings.

VI. CONCLUSION

This paper focused on the privacy issues of big data and considered the practical implementations in cloud computing. The proposed approach, D2ES, was designed to maximize the efficiency of privacy protections. Main algorithm supporting D2ES model was DED algorithm that was developed to dynamically alternative data packages for encryptions under different timing constraints. This paper proposed a data encryption solution for mobile health apps, called DE4MHA. The data encryption algorithm DE4MHA with cooperation mechanisms in mobile health allow users to safely obtain health information with the data being carried securely. More importantly, it offers a robust and reliable increase of privacy, confidentiality, integrity, and authenticity of their health information. Although it was experimented on a specific mHealth app, SapoFit, both DE4MHA and the cooperation strategy can be deployed in other mHealth apps. The experimental evaluations showed the proposed approach had an adaptive and superior performance.

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”. IEEE Transactions 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.