Accelerating Decryption of Aggregator in Mobile Sensing

Jie Wang*, Jiangjun Yuan

School of Shangmao Zhejiang Technical Institute of Economics Hangzhou，310018，China

***Abstract* — Mobile sensing is a new emergent application with the development of smart devices and wireless network. New appeared smart devices have been equipped with more powerful modules, such as GPS, camera and so on, and can conduct more complicated sensing tasks. An important application in mobile sensing is data aggregation. As people now pay more attentions to privacy, a lot of privacy-preserving data aggregation protocols have been proposed. Among them, some works are trying to decrease the computation cost of the devices in the sensing system. One good method to decrease the computation cost is to take advantages of multi-core architectures on mobile devices' processor. In this paper, we introduce multi-threads addition computation into the decryption of aggregator in a privacy-preserving data aggregation protocol. We implement the advanced protocol in java programming language and conduct two experiments which show the performance of the proposed protocol.**

Keywords - Cryptography; Privacy; Mobile Sensing; Multi-threads; Parallel Computation

I. INTRODUCTION

Mobile sensing is a new emergent application with the development of smart devices and wireless network. New appeared smart devices have been equipped with more powerful modules, such as GPS, camera and so on, and can conduct more complicated sensing tasks. Therefore, more and more applications come into being around these new appeared smart devices. Besides, wireless network technique also changes the communication way of the smart devices. The former sensor has to delivery its sensed data to a sink with the help of other nearby sensors. While nowadays, these new appeared smart devices can send data to its required part immediately which totally changes the communication model.

Data aggregation is an important task since the appearance of wireless sensor network. And in mobile sensing, data aggregation is also highly required. An important issue in today's data aggregation application is the concern of privacy. Therefore, privacy-preserving data aggregation protocol is paid many attentions by researchers. A lot of works[1-10] have been proposed to deal with this issue. Among them, some works are trying to decrease the computation cost of the devices in the sensing system.

Multi-core architectures have been widely applied for personal computers which have become an important field in our everyday computing. In addition, maximizing the performance of multi-core architectures, many frameworks and APIs have been proposed to do parallel programming on these advanced multi-core processors. Recently, processors in smart device or mobile phones also evolve into multi-core architectures[12]. Therefore, many researchers begin to develop parallel application on these multi-core mobile devices[13][14].

Taking advantages of multi-core architectures of these multi-core mobile devices, we can largely decrease the encryption and decryption time costs for privacy-preserving data aggregation protocols.

In this paper, we introduce multi-threads addition computation into the decryption of aggregator in a privacy-preserving data aggregation protocol. After the introduction of multi-threads addition computation, the privacy-preserving data aggregation protocol is more suitable for large mobile sensing system and is less time cost. We implement the advanced protocol in java programming language and conduct two experiments which show the performance of the proposed protocol.

II. RELATED WORKS

Security and privacy issues in mobile sensing system have been addressed by a lot of works [1][2]. However, they do not pay attention to data aggregation which is an important application[3].Many existing works can deal with a trusted aggregator, but will be not secure when the aggregator becomes untrusted [4][5]. In addition, the data in mobile sensing are usually time-series. The protocol proposed by Yang et al. [6] can deal with an untrusted aggregator, which however, requires expensive rekeying operation to support multiple time steps. Therefore, it may not be suitable for data aggregation in mobile sensing.

To deal with an untrusted aggregator, Shi et al. [7] proposed a data slicing and mixing techniques based protocol, Rastogi et al. [8] proposed threshold Paillier cryptosystem based encryption scheme and Rieffel et al. [9] proposed an efficient additive homomorphic encryption based protocol. These protocols either require more communication rounds or have the computation and storage cost. Li et al. [10] proposed a novel protocol based on straw-man construction and a light-weight additive homomorphic encryption[11]. The protocol enjoys communication and computation efficiency. But the computation overhead can be further decreased as we investigate it.

In this paper, we introduce multi-threads addition

computation into the decryption of aggregator in [10]. With our method, the privacy-preserving data aggregation protocol becomes more suitable for large mobile sensing system and is less time cost.

III. PRELIMINARIES

In this section, we will present the straw-man construction based privacy-preserving data aggregation protocol in mobile sensing. And the detailed protocol can be found in [10].

In the system model of the presented protocol, there are components, which are trusted authority, untrusted aggregator and mobile user. The trusted authority manages the sensing system and generates encryption/decryption related secret values for untrusted aggregator and mobile user. The untrusted aggregator is the outside user of the sensing system and wants to obtain the sum aggregation statics of the data sensed by mobile users. Mobile users work for the sensing system. And in each period, they will sense data and send them to the untrusted aggregator.

The protocol consists of three phases:

1. Trusted authority generates and assigns secret values
2. In each time period, mobile users sense data, encrypt the data and send the ciphertexts to the untrusted aggregator.
3. In each time period, the untrusted aggregator will calculate the sum aggregation statics from the ciphertexts sent by mobile users.

Phase 1: in the initial phase, the trusted authority generates *nc* secret values. Then it sends *c* secret values to each mobile use randomly. While it sends all the *nc* secret values to the aggregator.

Phase 2: in each time period, a mobile user senses data x_i and computes encryption key k_i according to *c* secret values. Then the mobile user will compute its ciphertext c_i with x_i and k_i according to the following equation:

$$c_i = (x_i + k_i) \bmod M \tag{1}$$

Phase 3:in each time period, the untrusted aggregator computes the decryption key k_0 with the assigned *nc* secret values. Then, it computes the sum aggregation statics from the *n* ciphertexts sent by mobile users

$$Sum = \left(\sum_{i=1}^{n} c_i - k_0 \right) \bmod M \tag{2}$$

Note that, *M* should be larger than the sum aggregation.

IV. MULTI-THREADS BASED VERSION

In Phase 3, the untrusted aggregator needs to add *n* ciphertexts into one ciphertext which will further decrypted by the decryption key.

Note that there is no priority in addition operation, which means that you can add some elements in an addition equation to obtain a part sum, and add the part sum with the left elements. We give an example as the following equation.

$$1+4+5+9+2=21 \tag{3}$$

We can first add

$$1+4+5=10 \tag{4}$$

Then add the part sum, which is 10, with the left elements

$$10+9+2=21 \tag{5}$$

According to this, we can modify equation 2 to the following one.

$$S = \left(\sum_{i=1}^{n} c_i - k_0 \right) \bmod M$$
$$\left(\sum_{i=1}^{d} \sum_{j=(i-1)\cdot(n/d)+1}^{i\cdot(n/d)} c_j - k_0 \right) \bmod M \tag{6}$$

For example, in a mobile sensing system with 20 mobile users, if we set *d* as 4, then the above equation can be modified as:

$$S = \left(\sum_{i=1}^{20} c_i - k_0 \right) \bmod M$$
$$= \left(\sum_{i=1}^{4} \sum_{j=5(i-1)+1}^{5i} c_j - k_0 \right) \bmod M$$
$$= \left(\left(\sum_{j=1}^{5} c_j + \sum_{j=6}^{10} c_j + \sum_{j=11}^{15} c_j + \sum_{j=16}^{20} c_j \right) - k_0 \right) \bmod M \tag{7}$$

The equation 2 can be handled with an addition operation thread while the equation 6 can be handled with multiple addition operation threads.

Figure 1 and Figure 2 show the difference of two equations when introducing multiple threads.

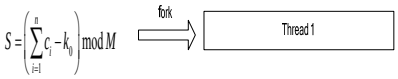

Fig.1 Computing sum aggregation with only one thread

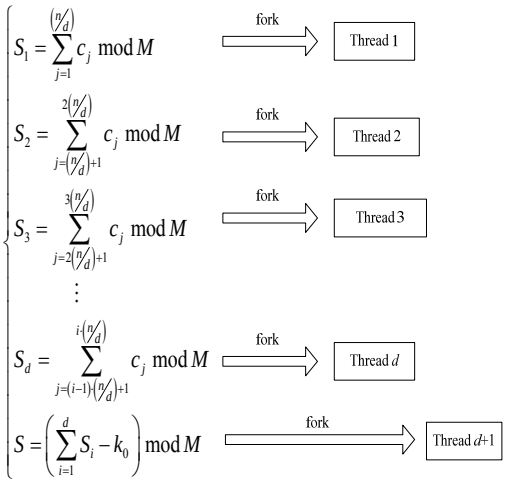

Fig.2 Computing sum aggregation with multi-threads

Figure 3 shows another case when introducing multiple threads.

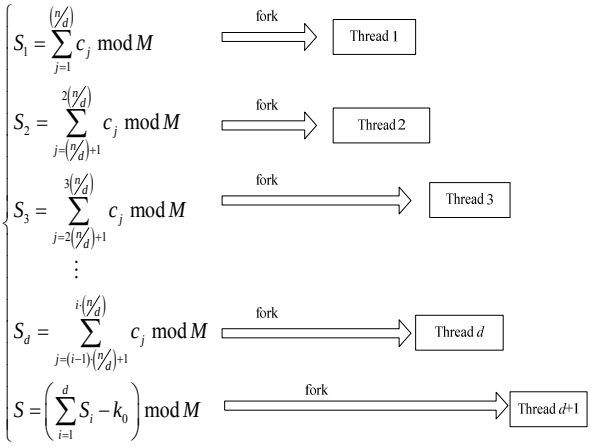

Fig.3 Another case that computing sum aggregation with multi-threads

We can see from the latter two figures, the total computation time cost of addition operation can be decreased largely.

V. PERFORMANCE AND COMPARISON

In this section, we conduct some experiments to show the performance of the multi-threads based version and make some comparisons between the with/without multi-threads based versions. Table I shows the experiment platform.

TABLE I. THE EXPERIMENT PLATFORM

Processor	**Inter® Core™ i5-2430M CPU@2.40GHz**
Memory Capacity	4G
Operating System	Window 10 Pro 64bit
JDK Version	1.8.0_51
Eclipse Version	Mars Release 4.5.0

Note that the processor of the experiment platform has two cores with four threads. And some memory capacities are used by graphics card.

Two experiments are conducted. The first experiment shows how the decryption time cost of the untrusted aggregator changes when the number of mobile users increases. While the second one shows how the number of threads will affect the decryption time cost of the untrusted aggregator.

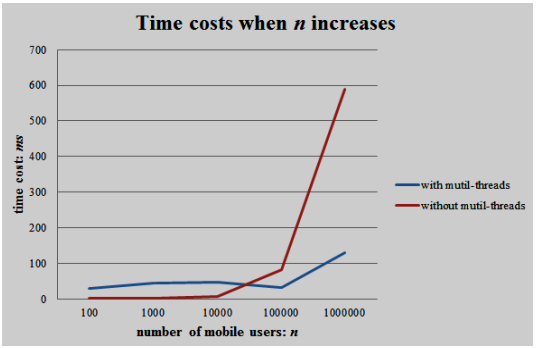

Fig.4 The time cost of the aggregator when the number of mobile users increases

We can see from Figure 4 that when the number of mobile users increases, the multi-threads version becomes more efficient. When the number of mobile users is small, the overhead of establishing threads surpasses the decreased computation overhead. Therefore, when computation overhead is small, the advantage of multi-threads is not very obvious. But, when the number of mobile users increases, the advantage of multi-threads becomes more obvious. Therefore, the multi-threads version, which we proposed, is suitable for large mobile sensing system, which is common in real applications.

Note that Figure 5 is a part of Figure 4 which clearly shows time cost changes when n increase from 100 to 10000. MT is short for multi-threads.

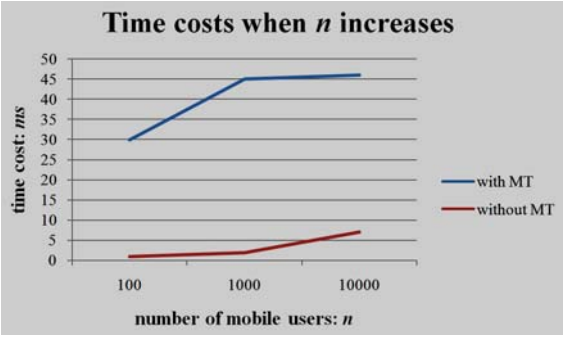

Fig.5 The time cost of the aggregator when n is less than 10000

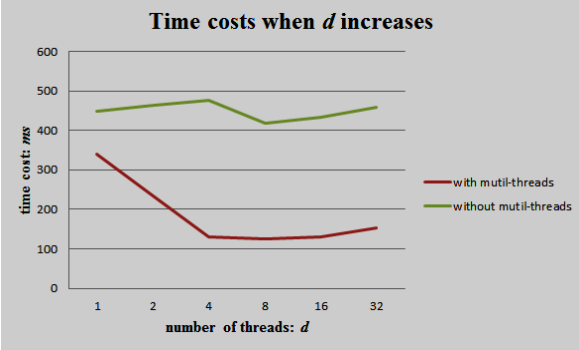

Fig.6 The time cost of the aggregator when the number of threads increases

The second experiment is conducted in the case that the number of mobile user is 1000000. And Figure5 shows the multi-threads version is better than the one without multi-threads. The decreased time cost is affected by the performance of the used processor which has two cores with four threads. Usually, when the number of used threads is closer to the processor owned threads, better performance can be achieved. Although, using multi-threads can decrease the time cost of computation, but the overhead of forking a new thread also should be considered.

VI. CONCLUSION

Mobile sensing is a new emergent application with the development of smart devices and wireless network. An important application in mobile sensing is data aggregation. As people now pay more attentions to privacy, a lot of privacy-preserving data aggregation protocols have been proposed. Among them, some works are trying to decrease the computation cost of the devices in the sensing system. One good method to decrease the computation cost is to take advantages of multi-core architectures on mobile devices' processor. In this paper, we introduce multi-threads addition computation into the decryption of aggregator in a privacy-preserving data aggregation protocol. After the introduction of multi-threads addition computation, the privacy-preserving data aggregation protocol is more suitable for large mobile sensing system and is less time cost. We implement the advanced protocol in java programming language and conduct two experiments which show the performance of the proposed protocol.

REFERENCES

[1] Q. Li and G. Cao, "Mitigating routing misbehavior in disruption tolerant networks," *IEEE Transactions on Information Forensics and Security*, 7(2), pp. 664–6752012

[2] E. D. Cristofaro and C. Soriente, "Short paper: Pepsi—privacy-enhanced participatory sensing infrastructure," in *Proceedings of the fourth ACM conference on Wireless network security (WiSec)*, pp. 23–28.2011

[3] Rajagopalan R, Varshney P K. Data-aggregation techniques in sensor networks: A survey[J]. *Communications Surveys & Tutorials IEEE*, 8(4), pp. 48-63.2006

[4] D. Bonet, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," *TCC*, 2005.

[5] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, ,5(3), pp. 20:1–20:36.2009

[6] Z. Yang, S. Zhong, and R. N. Wright, "Privacy-preserving classification of customer data without loss of accuracy," in *SIAM SDM*, pp. 21–23.2005.

[7] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proc. IEEE INFOCOM*, pp.758–766.2010.

[8] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," *ACM SIGMOD*, 2010.

[9] Rieffel E, Biehl J, Van Melle W, et al. Secured histories: computing group statistics on encrypted data while preserving individual privacy[J]. *Corr*, 2010.

[10] Li Q, Cao G. Efficient and privacy-preserving data aggregation in mobile sensing[C]// *2012 20th IEEE International Conference on Network Protocols (ICNP)IEEE*, pp.1-10.2012

[11] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, 5(3), pp. 20:1–20:36.2009

[12] Mair H, Gammie G, Wang A, et al. 23.3 A highly integrated smartphone SoC featuring a 2.5GHz octa-core CPU with advanced high-performance and low-power techniques[M]// *IEEE*, pp. 1-3.2015

[13] Abdullah, DhuhaBasheer, and M. M. Al-Hafidh. "Developing Parallel Application on Multi-core Mobile Phone." *International Journal of Advanced Computer Science &Application (4)*, pp. 11,*2013*

[14] P. Chanawangsa, C. Chen, "A New Smartphone Lane Detection System: Realizing True Potential of Multi-core Mobile Devices", *MoVid'12*, pp.19-24.2012