Lightweight Mobile Clients Privacy Protection Using Trusted Execution Environments for Blockchain

Jieren Cheng[1], Jun Li[2, *], Naixue Xiong[3], Meizhu Chen[2], Hao Guo[2] and Xinzhi Yao[2]

Abstract: Nowadays, as lightweight mobile clients become more powerful and widely used, more and more information is stored on lightweight mobile clients, user sensitive data privacy protection has become an urgent concern and problem to be solved. There has been a corresponding rise of security solutions proposed by researchers, however, the current security mechanisms on lightweight mobile clients are proven to be fragile. Due to the fact that this research field is immature and still unexplored in-depth, with this paper, we aim to provide a structured and comprehensive study on privacy protection using trusted execution environment (TEE) for lightweight mobile clients. This paper presents a highly effective and secure lightweight mobile client privacy protection system that utilizes TEE to provide a new method for privacy protection. In particular, the prototype of Lightweight Mobile Clients Privacy Protection Using Trusted Execution Environments (LMCPTEE) is built using Intel software guard extensions (SGX) because SGX can guarantee the integrity, confidentiality, and authenticity of private data. By putting lightweight mobile client critical data on SGX, the security and privacy of client data can be greatly improved. We design the authentication mechanism and privacy protection strategy based on SGX to achieve hardware-enhanced data protection and make a trusted connection with the lightweight mobile clients, thus build the distributed trusted system architecture. The experiment demonstrates that without relying on the performance of the blockchain, the LMCPTEE is practical, feasible, low-performance overhead. It can guarantee the privacy and security of lightweight mobile client private data.

Keywords: Blockchain, privacy protection, SGX, lightweight mobile client.

1 Introduction

In the past few years, consumers have been using smartphones, tablets, video surveillance cameras, and other lightweight mobile devices at a staggering rate. Different from traditional communication devices, they have independent operating systems on which users can install software and applications freely. Users must offer their personal

[1] School of Compute Science and Cyberspace Security, Hainan University, Haikou, 570228, China.

[2] Hainan Blockchain Technology Engineering Research Center, Hainan University, Haikou, 570228, China.

[3] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, 74464, USA.

[*] Corresponding Author: Jun Li. Email: 15556118727@163.com.

Received: 22 May 2020; Accepted: 07 July 2020.

information, store their critical data to these smart systems. The smartphone not only stores users' data but generates extensive personal data which may include user account passwords, pictures, multimedia files, and other kinds of sensitive information. However, because of inherent security concerns, lightweight mobile devices have yet to provide the same security found on desktop and laptop computer platforms. At the same time, as the performance and processing capacity of the lightweight mobile client continues to improve, Wi-Fi and rich sensors have also become essential functions of the lightweight mobile client, more and more user private data are stored on mobile devices. The role of security in mobile phones has received increased attention across several disciplines in recent years [Enck, Gilbert, Han et al. (2014); Enck, Octeau, McDaniel et al. (2011)]. Due to the diversity, intelligence, and openness of mobile clients, the privacy problems of lightweight mobile clients have greatly increased [Davidson, Rastogi, Christodorescu et al. (2017); Zheng, Yang, Shi et al. (2016)]. Malicious attackers can obtain user privileges by disguising malicious programs and other ways, to obtain users' text information, photos, audio, and other private data. Authorize security personnel in charge of the monitoring systems may abuse cameras for different purposes, such as snooping [Streiffer, Srivastava, Orlikowski et al. (2017)], network tracking [Wang, Amos, Das et al. (2018)], and unauthorized collection of data related to individual activities or behaviors [Yu, Zhang, Kuang et al. (2017)]. Several security methods have been developed and applied to prevent unauthorized access to critical data stored on mobile clients. Some methods encrypt the data to prevent access. However, perhaps the most serious disadvantage of this method is that simple encryption passwords may be broken, and more secure encryption techniques will become more complex, requiring more resources that may not be available on some resource-constrained devices. To build a secure and trusted privacy protection environment is an urgent problem to be studied. Existing mobile applications bring huge security and privacy concerns, and the future mobile internet of things service model should allow data owners to have complete control over their data and determine who can access their private data. What is therefore needed is a system architecture for lightweight mobile clients to guarantee data privacy, security, integrity, and authenticity.

In step with the continued advances in decentralization and low transaction costs, Bitcoin is gaining in prominence. It can establish peer-to-peer credible value transfer between unfamiliar nodes without relying on third-party trusted organizations, which helps to reduce transaction costs and improve interaction efficiency. Nakamoto [Nakamoto (2008)] expresses a view that 'What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party'. As can be seen, the development of ubiquitous and decentralized digital currencies like Bitcoin is increasingly used as a payment method [Bamert, Decker, Elsen et al. (2013)]. Nevertheless, according to Smith et al. [Smith, Cary, BaynhamHerd et al. (2019)], when verifying the transaction information, users need to download data larger than 200G in the lightweight client, however, the clients with a limited resource such as mobile devices cannot handle such a large amount of data. Besides, Gervais et al. [Gervais, Capkun, Karame et al. (2014)] show that existing Bitcoin clients exposed a large number of user privacies. Blockchain, as the underlying technology of cryptocurrencies, has attracted considerable attention as it offers security, stability, and efficiency in the past ten years. The introduction of blockchain technology and trusted execution environment (TEE) in lightweight mobile clients can solve the problems of privacy data leakage, insufficient computing capacity, and storage capacity of lightweight mobile clients. By using the Intel software guard extensions (SGX) as a hardware-enhanced encrypted storage device, multiple encryption algorithms are designed to encrypt and store data. By storing the hash memory table of data in the blockchain network, the authenticity of the data hash is guaranteed, thus overcoming the dependence on the third party and protecting the privacy of lightweight mobile clients. In this paper, our goal is to utilize SGX to solve the problem of lightweight mobile client data privacy protection. We proposed a hardware-based isolation system called LMCPTEE which gives critical data more protection against disclosure or modification. Specifically, we design data privacy protection methods inside the SGX, which involves remote attestation, data encryption and decryption, and data integrity check. Experiments demonstrate that the system solution not only protects the private data with hardware-enhanced data protection and low-performance overhead but also provides a secure and trusted privacy protection environment. It has certain important significance for the development of distributed privacy protection in lightweight mobile clients.

2 Related work

On the one hand, there is a large volume of published studies describing the problem of light client privacy. Matetic et al. [Matetic, Wüst, Schneider et al. (2018)] proposed a new method to protect the privacy of Bitcoin clients. They use the trusted execution environment SGX to design a system called BITE to protect user privacy. Besides, various studies have been proposed for privacy protection, such as local privacy protection classification [Yin, Zhou, Yin et al. (2019)], location recommendation privacy protection [Yin, Ju, Yin et al. (2019)], image information hiding approach based on deep learning [Luo, Qin, Xiang et al. (2020)], and location data record privacy protection [Gu, Yang and Yin (2018)]. Yin et al. [Yin, Shi, Sun et al. (2019)] found that the collaborative filtering recommendation algorithm can be improved based on differential privacy protection and time factor. Medhane et al. [Medhane, Sangaiah, Hossain et al. (2020)] proposed a blockchain-enabled distributed security framework for next-generation internet of things (IoT). Xia et al. [Xia, Tan, Wang et al. (2019)] designed a smart contract for multi-party bidding power resources based on blockchain technology and achieved the decentralized power trading decision to ensure the information is symmetric and fair. Zhang et al. [Zhang, Zhong, Wang et al. (2020)] summarized the systems and applications based on blockchain in data security protection. To solve the cloud computing service project of resource-limited devices, a new method was introduced in literature [Li, Niu, Duan et al. (2014)], which allows a multi-keyword ranking search on the encrypted database, but symmetric key encryption has the risk of key leakage. The study [Li and Niu (2016)] proposes a secure and private keyword search method, which is used to store the encrypted data of application programs in the cloud using elliptic curve cryptography (ECC). The scheme only considers whether or not the file exists, however, key words, not considering and results in the differences in the correlation of these file query keywords. Tang et al. [Tang, Yang, Dong et al. (2016)] proposed a two-wheel searchable encryption scheme to support for encrypted data ranked more keyword search for document retrieval. It uses a vector space model and homomorphic encryption to ensure data security. However, the computing and communication costs of this scheme are very large. These solutions can indeed solve the pressure on the resource-constrained client, but it does not immediately address the client's privacy issues. Passive mode cannot effectively protect the privacy and security of user data. In addition, running these methods adds resource overhead, which is especially valuable in resource-constrained lightweight mobile clients. On the other hand, several studies have shown that TEE is a very popular research direction. The paper [Lind (2018)] designed TEEchain using a trusted execution environment with SGX, a new, secure under-chain payment protocol that is efficient, without simultaneous access to the blockchain. Zhang et al. [Zhang, Cecchetti, Croman et al. (2016)] use SGX to provide authenticated data feeds system for smart contracts. Ren et al. [Ren, Liu, Ji et al. (2018)] draw attention to design the first incentive mechanisms of nodes for data storage based on the blockchain technology in wireless sensor networks. Matetic et al. [Matetic, Schneider, Miller et al. (2018)] designed a system using a trusted execution environment with SGX, enabling users to log in to different online services using credentials from other users, allowing users to determine access rights. SGX has been utilized to build trust authentication environments [Gu (2015)] and trusted network communication channels [Jain, Desai, Kim et al. (2016)]. Microsoft blockchain framework [Microsoft (2017)] is an open-source system that supports a large-scale security blockchain network that meets the needs of all key enterprises, providing a means to accelerate the adoption of blockchain technology by manufacturing companies. The research [Davidsen, Gajek, Kruse et al. (2018); Park and Kim (2017); Ayoade, Karande, Khan et al. (2018)] used the trusted execution environment to ensure the integrity and privacy of IoT data. Our work is more common than these previous jobs because it supports the application of existing blockchain platforms.

These latest studies have achieved good results and have been widely used. However, they only consider protecting the security and privacy of user data on lightweight mobile clients. We proposed to move the important data of lightweight mobile clients to nearby SGX and store encrypted hash tables in the blockchain to protect the security and privacy of user critical data.

3 LMCPTEE system

3.1 System model

The aim of this section is to introduce the LMCPTEE system model. As shown in Fig. 1, the system is divided into three parts:

Lightweight mobile clients: In general, lightweight mobile clients tend to be used to refer to mobile phones, laptops, cameras, tablets, etc. Lightweight mobile clients aim to establish a connection with the SGX to request and transmit data that the user would like to store encrypted. Besides, users can choose to check the security of the SGX platform through access to the certification reports on the bulletin board to the intel attestation service (IAS) [Johnson, Scarlata, Rozas et al. (2016)].

SGX platform: On the one hand, the SGX platform has three basic functions. Firstly, SGX establishes a trusted connection with lightweight mobile clients to provide further services.

Secondly, the raw data is stored on the SGX platform. SGX creates an enclave to perform authentication mechanisms, data encryption and decryption, and data integrity checks for different types of data. Thirdly, SGX establishes a trusted connection with the blockchain network to prepare for encrypted transmission of data. On the other hand, the SGX platform will send the attestation report to public bulletin boards to provide a remote attestation for users.

Blockchain network: The main goal of the blockchain network is to store the hash memory table with which you can uniquely identify the data from SGX. The LMCPTEE system provides strong distributed privacy protection for lightweight mobile clients and is particularly well suited for blockchain.

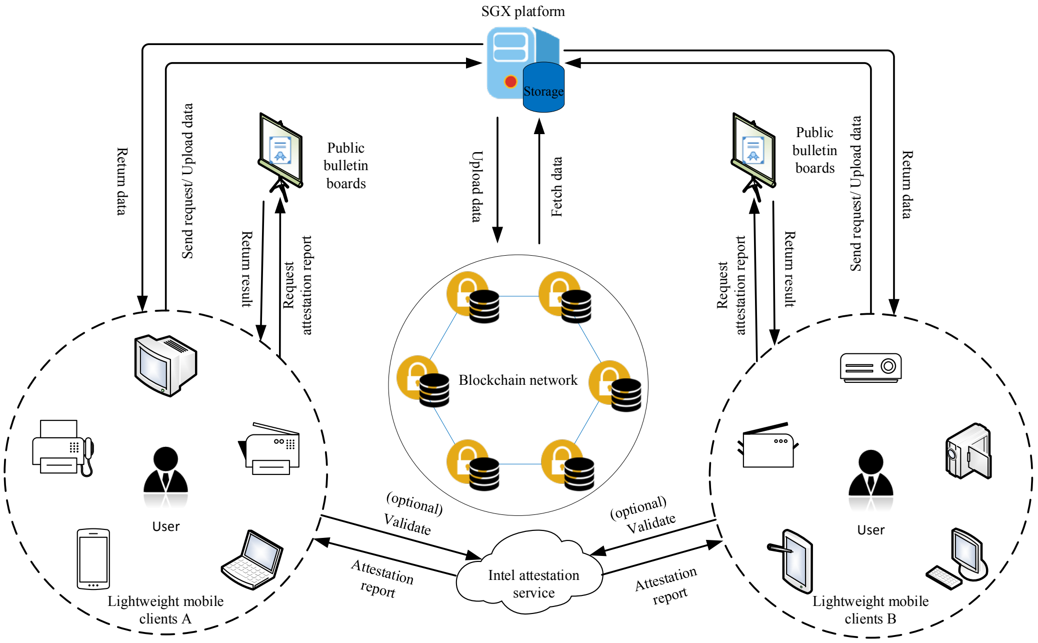

Figure 1: System model

3.2 System architecture

The main goal of the lightweight mobile client privacy protection system is to protect the entire lifecycle of the data, which includes data encryption and decryption, data integrity check, and data storage. SGX not only plays a trusted role in the system, but it is also an intermediary agent, which is designed to provide data protection for lightweight mobile client privacy and prevent malicious access by unauthorized parties. As shown in Fig. 2, the details of the system architecture are as follows:

● Step 1: First, when SGX creates an enclave, it generates an attestation report. Then lightweight clients can choose to verify whether SGX is legal by using the attestation report to IAS. Communication at this stage can be set up in any available method.

● Step 2: The lightweight mobile client MC1 will establish a trusted communication channel to the SGX. Here we use transport layer security (TLS) to make reliable communication.

● Step 3: The lightweight mobile client MC1 sends a data request or upload critical data to SGX, which is used to secure storage. Note that lightweight clients for different purposes may send different types of data.

● Step 4: When SGX receives the data, it executes the privacy protection strategy, which includes data classification, encryption, storage, etc.

● Step 5: When SGX processes the data, it gets a hash table uploaded to the blockchain network. Hash tables are very small and do not take up much of the storage space of blockchain nodes.

● Step 6: SGX makes a request to the blockchain network when the privacy strategy needs to get the hash table on the blockchain.

● Step 7: When the blockchain receives a request from SGX, it returns a hash table of data saved on the chain.

● Step 8: When the privacy strategy is completed, SGX returns the corresponding data to the lightweight mobile client.

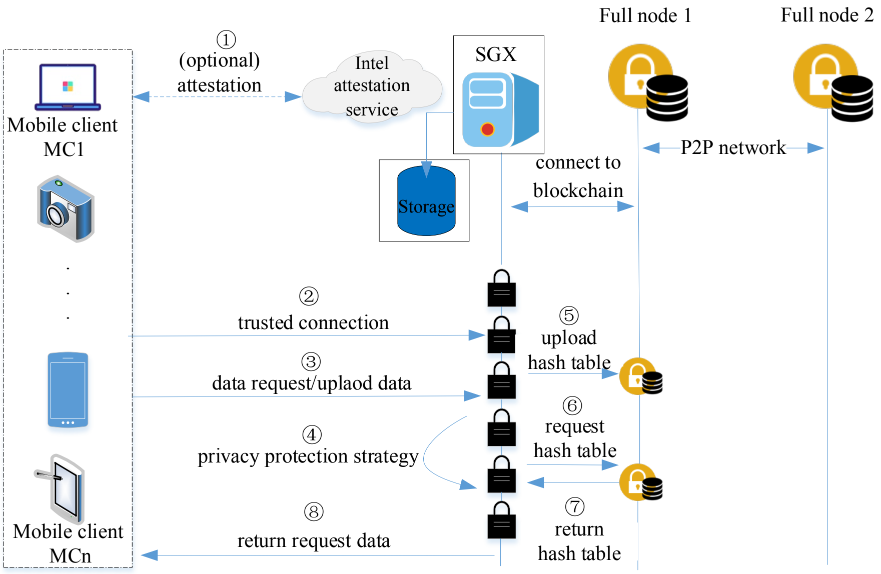

Figure 2: System architecture

3.3 SGX design

The SGX design aimed to develop a better understanding of SGX hardware-enhanced privacy protection. Fig. 3 provides the overall structure of the SGX platform. The application implemented in the SGX platform consists of two modules: untrusted modules and trusted modules. The trusted module is loaded and executed inside the SGX enclave while the untrusted module is running outside the enclave. By leveraging a trusted execution environment based on SGX, we have designed three sub-modules, which is an authentication mechanism, data encryption and decryption, and data integrity check. The data privacy protection strategy includes encryption, decryption, and integrity check. Note that modules in the blue background are trusted, while the ones in the white background are untrusted.

Remote attestation: The lightweight mobile client performs a remote attestation to the SGX enclave. The LMCPTEE starts its execution in SGX, generates an attestation report which contains public keys, secret keys, and publishes its identities, etc. So that users can verify them remotely. The attestation report is distributed by the SGX platform through public bulletin boards such as public blockchains. Note that for most users, interaction with IAS is optional, as any user or third party can distribute the IAS proof verification results so that the user can verify the proof report himself.

Data encryption and decryption: The data received by the SGX platform will be encrypted in the enclave and then stored in the local disk. The data encryption and decryption method use the SGX file protect system library. When the user makes a data request, the data will be decrypted in the enclave, and then to check the data integrity to determine whether the data has tampered. Data encryption and decryption aim to ensure the sensitive data is complete, autonomous, and controllable.

Data integrity check: When a user uploads the data, the first steps performed by the data integrity check is to calculate its hash value. The hash value will be uploaded to the blockchain network for storage. When SGX receives a data request from a lightweight mobile client, it then performs a data integrity check method to access the blockchain network to get the data hash memory table that has already been saved on the blockchain network. Then compares the two hash values, if the hash is the same, the data is safe, otherwise, if the hash value is different, the data has been modified. That way we can tell if the data was modified by looking at the hash value. SGX would then return the original data that was uploaded to the SGX platform.

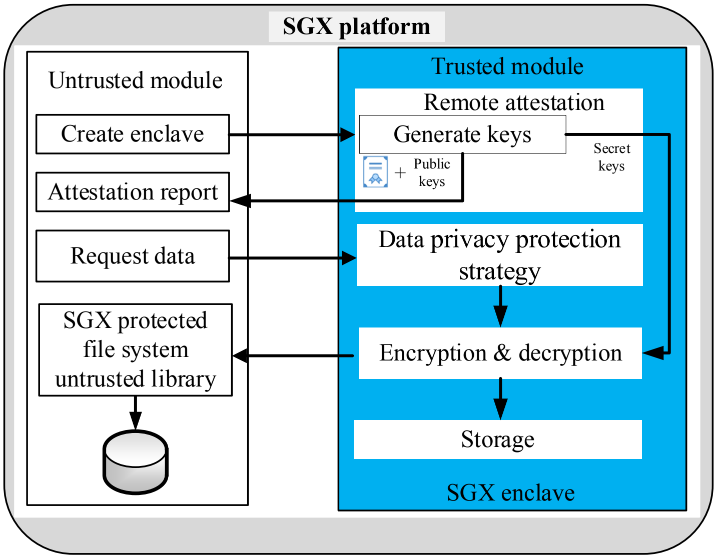

Figure 3: SGX design structure

4 Performance evaluation

In this section, we will introduce the implementation and show that the performance overhead spent by the solution stays within a reasonable range. We evaluate LMCPTEE on a Dell Precision 3630 desktop that is SGX-enabled with the 9th generation Intel Core i7-9700K CPU, 16GB, Intel SGX platform software (PSW), and ubuntu 16.04 LTS with Linux kernel 4.4.

4.1 Enclave response time

This experiment measured the response time taken of the enclave, which is the time from the request made of a lightweight mobile client to the response of the enclave. For the purpose of analysis, we performed three group experiments, each group has 30 times and recorded the results of statistical data. All times are in milliseconds. Fig. 4 compare and summarizes the experimental data on the total response time of the enclave. From the measurement results, it can be seen that the response time of each group fluctuated around 4 milliseconds. What is interesting about the data in Fig. 4 is that the Mean time of the enclave is about 4 milliseconds. This is easily acceptable in practice since response times below 0.1 seconds give users a sense of instant response [Nielsen (2010)].

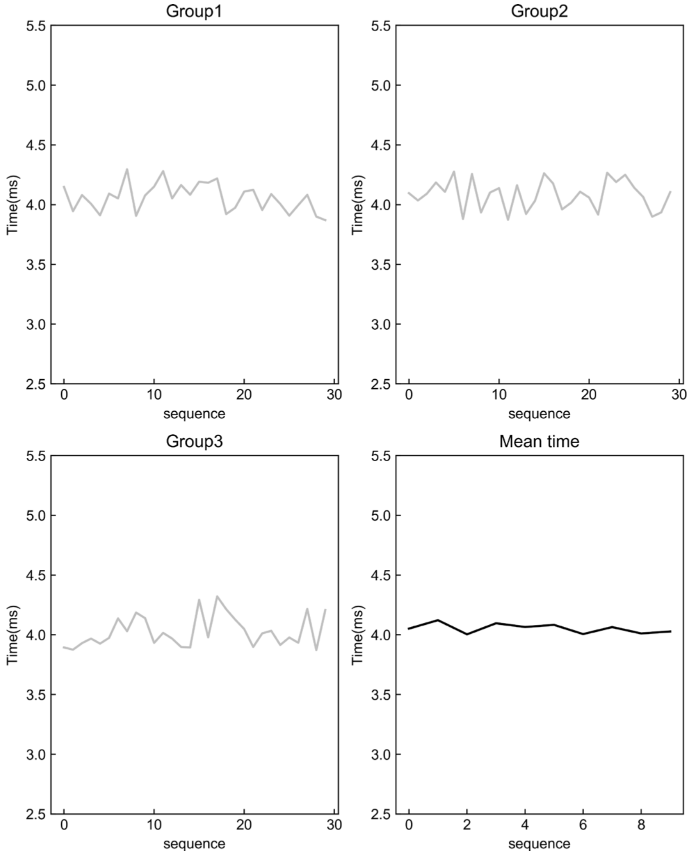

Figure 4: Enclave response time

4.2 Encryption and decryption time

This experiment set out to compare the time required for SGX to encrypt and decrypt files of different sizes. We tested the encryption and decryption time required for four different file sizes, including 10 MB, 40MB, 80 MB, and 130 MB. Tab. 1 illustrates the amount of time required to encrypt and decrypt files in the enclave. Fig. 5 compares our experimental results with the encryption time and decryption results obtained from the preliminary study [Kubadia, Idnani and Jain (2019)]. Looking at Fig. 5, it is apparent that the encryption and decryption time of groups 10 MB, 40 MB, 80 MB, and 130 MB are significantly lower than that of the control group of the same size. The encryption and decryption results confirm that this a good choice for using advanced encryption standard (AES) to securely protect critical data. We can also see that with successive increases in the data size, the encryption workload moved to a slow increase. However, it will not affect the overall performance of the system.

Table 1: Encryption and decryption time

Message size (MB)	Encryption time (sec)	Decryption time (sec)
1	0.033806	0.050442
2	0.087838	0.089903
4	0.165196	0.160118
8	0.316141	0.318748
16	0.647923	0.649272
32	1.289156	1.293616
64	2.659563	2.59036
128	5.182357	5.174406
10	0.028342	0.007523
40	0.115297	0.014210
80	0.237621	0.018938
130	0.365723	0.024851

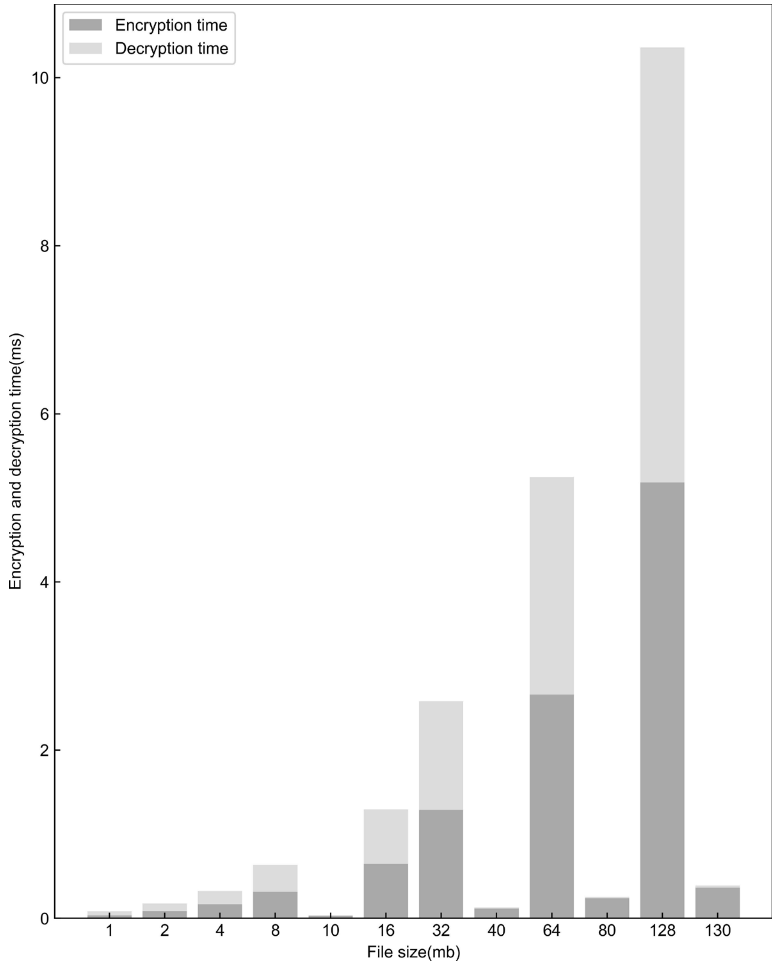

Figure 5: Comparison of encryption decryption time

4.3 Communication overhead

To analyze the communication overhead of transferring different files, we compared send different files to the lightweight mobile client by SGX. Test files include text, images, audio, and video. The data sizes we tested ranged from 10 MB to 120 MB, and the communication overhead required for both types of files fluctuated in the range of 0.12 seconds to 1.4 seconds. Fig. 6 displays the median of communication overhead for each type of file. In Fig. 6, when the size of the test data is 10 MB, the communication overhead is close to 0.12 seconds. When the size of the test data is 80 MB, it takes 0.9 seconds. When the amount of data is large, the test data is 120 MB, and the communication overhead is 1.4 seconds. A positive correlation was found between file size and communication overhead. The communication overhead of the test results is very low, mainly because we tested on the local platform, not on the remote platform. As shown in Fig. 6, the x-axis is the size of the different files, and the y-axis is the communication overhead.

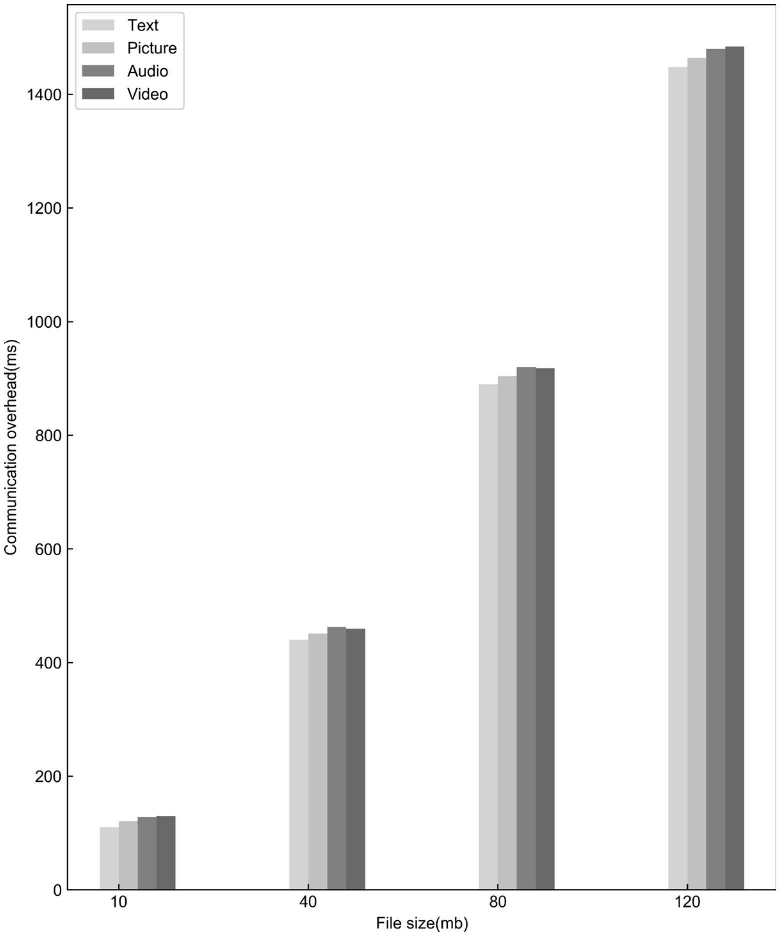

Figure 6: Communication overhead

4.4 Transmission cost

The purpose of this experiment was to provide an overview of the transmission cost required by SGX to obtain the data hash table of different sizes on the blockchain network. In the experiment, the data is downloaded to the SGX platform, each time the data size is 10 KB, 25 KB, and 50 KB. As we can see, when the file size is 10 KB, the transmission cost is about 0.6 seconds. Data from Tab. 2 shows when the data size is the same, the average time overhead required by our method is less, and the use of distributed queries further protects the security of private data. We must have to indicate that the time required for data and transmission cost is greatly affected by network speed, but when the size of data increases, such as the data size is 25 KB and 50 KB, the average transmission cost is 0.59 seconds and 0.68 seconds. The experimental results show that the transmission cost is still within a reasonable range.

Table 2: Transmission cost

Hash table size (KB)	10	25	50
Time (sec)	0.66	0.64	0.71
	0.71	0.65	0.73
	0.59	0.52	0.65
	0.65	0.51	0.62
	0.62	0.62	0.70

5 Discussion and limitations

Blockchain differs from SGX in some important ways. The blockchain has the characteristics of decentralized consensus, high availability, and transparency of data, but only data and code within the blockchain can ensure trust. It does not extend to interaction with the real world. However, SGX can guarantee the integrity, confidentiality, and authenticity of code and data, but SGX can only run on a single device, it can neither provide high availability of data nor provide decentralized trust. Therefore, it is clear that blockchain and SGX have complementary characteristics. Through the combination of blockchain and SGX, we can solve the problems of protection and calculation of confidential information. In this study, results indicate that with the support of SGX, our solutions can provide high performance, low latency, and confidentiality for sensitive data.

SGX attack: SGX will be attacked, such as side-channel attacks and cache attacks. The main means of a side-channel attack is to obtain data through the attack surface, derive control flow and data flow information, and finally obtain enclave code and data information. In this article, we assume that devices equipped with SGX are secure and will not be compromised by malicious attackers. To handle the attacks, LMCPTEE uses AES to encrypt and decrypt files in the SGX platform because the AES instructions can protect AES against side channels attacks. Storing the hash on blockchain gives the guarantee that whenever the user tries to download the hash, the user would get back the same hash that was originally stored. This is guaranteed by the inherent nature of Blockchain. For the attacks which explore the hardware vulnerabilities, Intel has submitted patches to solve the problems.

DDoS attack: There are two types of distributed denial-of-service (DDoS) attacks in the system. First, an attacker who can control many lightweight mobile clients in the LMCPTEE system may consume LMCPTEE network resources by requesting multiple data through SGX to the blockchain. The attacker may just send requests continuously and then does not do any operations. To solve the problem, in our design, when SGX receives the request, it sends it to the blockchain and returns the data which the user wants, but SGX does not wait to verify if the user receives the data. In this way, our system can mitigate this attack. Second, an attacker may choose to take SGX directly without consuming network resources by sending a large number of data validation requests from the attack client. In this way, SGX is paralyzed. This attack will be where the system will enhance in the future. Finally, we refer the interested readers to [Cheng, Zhou, Liu et al. (2016); Cheng, Xu, Tang et al. (2018)] for an understanding of DDoS attacks and potential solutions.

Replay attack: To protect the private data from the replay attack, each file must be hashed with a nonce and then stored on the external hard disk. The hashed result and the nonce are stored in the enclave. The main purpose of this method is to improve the anti-tampering ability of the file and protect the integrity of the file. When a user needs to call the file, the enclave first hash the file and then compares the hash value to the hash table stored on the blockchain, if the two results are the same, the file is safe, otherwise, the file has been modified. We conclude that LMCPTEE using SGX which guarantees the confidentiality and the integrity of the private data can provide much more security and privacy than traditional lightweight clients. All the data of the user request are encrypted and confidential information is stored inside the enclaves. Malicious attackers can only stop service, modify or record the encrypted messages, which will not destroy the integrity and confidentiality of data. It is worth noting that our ideas are easy to achieve.

Limited enclave memory: At present, the memory limit of SGX is 128 MB. If future versions of SGX will design more memory, we can store more critical data in memory and design a new in-memory database system, such as SGX-based memory proposed by Priebe et al. [Priebe, Vaswani and Costa (2018)] database systems. Combined with larger memory, the system performance of this design will also be greatly improved.

6 Conclusion

With the rapid development of lightweight mobile clients equipped with a lot of features, the data privacy protection problem is increasing. The current mainstream lightweight clients cannot help users store critical data without leak user privacy. This work contributes to the existing knowledge of lightweight mobile client privacy protection by providing deeper insight into SGX. SGX emerged as a reliable new approach which has hardware-assisted confidentiality and integrity-added protections to increase the security of critical data. In this article, first of all, we presented the LMCPTEE which is a confidential and trusted distributed privacy protection system that uses SGX trusted hardware to complement and solve data privacy protection problems in the existing lightweight mobile client. Secondly, the findings reported here shed new light on SGX which can help increase privacy and security for data confidentiality, integrity, and authenticity. Finally, the evidence from this study suggests that our solutions can provide low latency, secure data storage, reliable authentication mechanism for lightweight mobile clients. In addition, future investigations are necessary to establish whether a large enclave memory can improve data handling performance.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 61762033), Hainan Provincial Natural Science Foundation of China (Grant Nos. 2019RC041 and 2019RC098), Opening Project of Shanghai Trusted Industrial Control Platform (Grant No. TICPSH202003005-ZC), Ministry of Education Humanities and Social Sciences Research Program Fund Project (Grant No. 19YJA710010), and Zhejiang Public Welfare Technology Research (Grant No. LGF18F020019).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Ayoade, G.; Karande, V.; Khan, L.; Hamlen, K. (2018): Decentralized IoT data management using blockchain and trusted execution environment. *IEEE International Conference on Information Reuse and Integration for Data Science*, pp. 15-22.

Bamert, T.; Decker, C.; Elsen, L.; Wattenhofer, R.; Welten, S. (2013): Have a snack, pay with Bitcoins. *IEEE P2P Proceedings*, pp. 1-5.

Cheng, J. R.; Zhou, J.; Liu, Q.; Tang, X.; Guo, Y. (2016): A DDoS detection method for socially aware networking based on forecasting fusion feature sequence. *Computer Journal*, vol. 61, no. 7, pp. 959-970.

Cheng, J. R.; Xu, R. M.; Tang, X. Y.; Sheng, V. S.; Cai, C. T. (2018): An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95-119.

Davidsen, M.; Gajek, S.; Kruse, M.; Thomsen, S. (2018): *Empowering the Economy of Things*. Weeve, Berlin.

Davidson, D.; Rastogi, V.; Christodorescu, M.; Jha, S. (2017): Enhancing android security through app splitting. *International Conference on Security and Privacy in Communication Systems*, pp. 24-44.

Enck, W.; Gilbert, P.; Han, S.; Tendulkar, V.; Chun, B. G. et al. (2014): TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems*, vol. 32, no. 2.

Enck, W.; Octeau, D.; McDaniel, P.; Chaudhuri, S. (2011): A study of Android application security. *Proceedings of the 20th USENIX Security Symposium*, pp. 315-330.

Gervais, A.; Capkun, S.; Karame, G.; Gruber, D. (2014): On the privacy provisions of bloom filters in lightweight Bitcoin clients. *Proceedings of the 30th Annual Computer Security Applications Conference*, pp 326-335.

Gu, K.; Yang, L. H.; Yin, B. (2018): Location data record privacy protection based on differential privacy mechanism. *Information Technology and Control*, vol. 47, no. 4, pp. 639-654.

Gu, J. J. (2015): Intel hardware-based security technologies bring differentiation to biometrics recognition applications part 1.

https://software.intel.com/en-us/articles/intel-hardware-based-security-technologies-bring-differentiation-to-biometrics-recognition.

Jain, P.; Desai, S.; Kim, S.; Shih, M. W.; Lee, J. H. et al. (2016): OpenSGX: an open platform for SGX research. *The Network and Distributed System Security Symposium*, http://doi.org/10.14722/ndss.2016.23011.

Johnson, S.; Scarlata, V.; Rozas, C.; Brickell, E.; Mckeen, F. (2016): Intel software guard extensions: EPID provisioning and attestation services.

https://software.intel.com/content/www/us/en/develop/download/intel-sgx-intel-epid-provisioning-and-attestation-services.html.

Kubadia, A.; Idnani, D.; Jain, Y. (2019): Performance evaluation of AES, ARC2, Blowfish, CAST and DES3 for standalone systems: symmetric keying algorithms. *3rd International Conference on Computing Methodologies and Communication*, pp. 118-123.

Li, G. Q.; Niu, P. F.; Duan, X. L.; Zhang, X. Y. (2014): Fast learning network: a novel artificial neural network with a fast learning speed. *Neural Computing and Applications*, vol. 24, pp. 1683-1695.

Li, G. Q.; Niu, P. F. (2016): Combustion optimization of a coal-fired boiler with double linear fast learning network. *Soft Computing*, vol, 20, pp. 49-156.

Lind, L. (2018): Teechain: scalable blockchain payments using trusted execution environments. https://arxiv.org/abs/1707.05454v1.

Luo, Y. J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Liu, Q. et al. (2020): Coverless real-time image information hiding based on image block matching and dense convolutional network. *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125-135.

Matetic, S.; Wüst, K.; Schneider, M.; Kostiainen, K.; Karame, G. et al. (2018): Bite: bitcoin lightweight client privacy using trusted execution. *In USENIX Security*, pp. 783-800.

Matetic, S.; Schneider, M.; Miller, A.; Juels, A.; Capkun, S. (2018): Delegatee: brokered delegation using trusted execution environments. *27th {USENIX} Security Symposium*, pp. 1387-1403.

https://www.usenix.org/conference/usenixsecurity18/presentation/matetic.

Medhane, D. V.; Sangaiah, A. K.; Hossain, M. S.; Muhammad, G.; Wang, J. (2020): Blockchain-enabled distributed security framework for next generation IoT: an edge-cloud and software defined network integrated approach. *IEEE Internet of Things Journal*, https://doi.org/10.1109/JIOT.2020.2977196.

Microsoft. (2017): The confidential consortium blockchain framework: technical overview. https://www.microsoft.com/en-us/research/project/confidential-consortium-framework/.

Nakamoto, S. (2008): Bitcoin: a peer-to-peer electronic cash system. *Consulted*, vol. 1, no. 2012, pp. 28.

Nielsen, J. (2010): Website response times. https://www.nngroup.com/articles/website-response-times/.

Park, J.; Kim, K. (2017): TM-Coin: trustworthy management of TCB measurements in IoT. *IEEE Percom Workshop on Security Privacy and Trust in The Internet of Things*, pp. 654-659.

Priebe, C.; Vaswani, K.; Costa, M. (2018): EnclaveDB: a secure database using SGX. *IEEE Symposium on Security and Privacy*, pp. 264-278.

Ren, Y. J.; Liu, Y. P.; Ji, S.; Sangaiah, A. K.; Wang, J. (2018): Incentive mechanism of data storage based on blockchain for wireless sensor networks. *Mobile Information Systems*, https://doi.org/10.1155/2018/6874158.

Smith, P.; Cary, N.; BaynhamHerd, X.; McGarraugh, C.; Kgil, M. et al. (2019): Blockchain.info. https://blockchain.info.

Streiffer, C.; Srivastava, A.; Orlikowski, V.; Velasco, Y.; Martin, V. et al. (2017): ePrivateEye: to the edge and beyond! *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 18.

Tang, D. Y.; Yang, J.; Dong, S. B.; Liu, Z. (2016): A lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems. *Applied Soft Computing*, vol. 49, pp. 641-662.

Wang, J. J.; Amos, B.; Das, A.; Pillai, P.; Sadeh, N. et al. (2018): Enabling live video analytics with a scalable and privacy-aware framework. *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 3s, pp. 64.

Xia, Z. Q.; Tan, J. J.; Wang, J.; Zhu, R. L.; Xiao, H. G. et al. (2019): Research on fair trading mechanism of surplus power based on blockchain. *Journal of Universal Computer Science*, vol. 25, no. 10, pp. 1240-1260.

Yin, C. Y.; Zhou, B.; Yin, Z. C.; Wang, J. (2019): Local privacy protection classification based on human-centric computing. *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 33.

Yin, C. Y.; Ju, X. K.; Yin, Z. C.; Wang, J. (2019): Location recommendation privacy protection method based on location sensitivity division. *Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 266.

Yin, C. Y.; Shi, L. F.; Sun, R. X.; Wang, J. (2019): Improved collaborative filtering recommendation algorithm based on differential privacy protection. *Journal of Supercomputing*, pp. 1-14. https://doi.org/10.1007/s11227-019-02751-7.

Yu, J.; Zhang, B. P.; Kuang, Z. Z.; Lin, D.; Fan, J. P. (2017): iPrivacy: image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005-1016.

Zhang, J. Y.; Zhong, S. Q.; Wang, T.; Chao, H. C.; Wang, J. (2020): Blockchain-based systems and applications: a survey. *Journal of Internet Technology*, vol. 21, no. 1, pp. 1-14.

Zhang, F.; Cecchetti, E.; Croman, K.; Juels, A. (2016): Town crier: an authenticated data feed for smart contracts. *ACM Conference on Computer & Communications Security*, pp. 270-282.

Zheng, X.; Yang, L.; Shi, G.; Meng, D. (2016): Secure mobile payment employing trusted computing on TrustZone enabled platforms. *IEEE Trustcom/BigDataSE/ISPA*, pp. 1944-1950.