A Secured Outsourced Description FEPOD Scheme in Blockchain

A. Dr. Asma Parveen[1], B. Mahewesh Khanam[2]
[1]*Head of the Department, Dept. of Computer Science and Engineering, Khaja Bandanawaz University, Kalaburagi, Karnataka, India*
[2]*Student, Dept. of Computer Science and Engineering, Khaja Bandanawaz University, Kalaburagi, Karnataka, India*

Abstract **- With the aim of developing application that demand both capacity and data sharing, the notion of functional encryption is employed here to overcome the limitation of public key encryption. The actual limitations existing in functional encryption is the fact the most functional encryption plans are based on bilinear pairings which are exceedingly expensive computation. A generally acknowledged solution for this issue is to outsource the important task to a trusted extraordinary outsider while leaving the easy calculations to the client. Expecting an outsider (such as third party) to supply free services is impractical in any case. In the existing concept of functional encryption with outsourced description (FE-OD) scheme no thought has been given to the payment procedure between the client and the outsider as far as anyone is concerned, based on the idea that neither of them should be trusted. In this project the goal is to design the existing functional encyption with payable outsourced decoding FE-POD scheme. A blockchain based digital currency is used in FE-POD system to allow the client to pay an outsider after the client accurately performs the outsourced decoding. Also, we aim to define the non-exclusive model of FE-POD system and then represent general implementation of FE-POD scheme. Hence, we also evaluated the execution of the proposed concept by implementing a FE-POD system over a blockchain-based platform.**

Index Terms **- outsource, FE-OD (functional encryption with outsourced description), FE-POD(functional encryption with payable outsourced description), digital currency.**

INTRODUCTION

Let us take an example of a cloud storage where all the data is encoded through encryption mechanism such as functional encryption[5],[6] and is stored in encoded form to provide the data security and authenticity. In general terms we can define outsourcing as a group of customs undertaken by a company or authorities to perform bulk operation from the third party which are the outsiders to the organisation. As far as the basic operation is concerned the classical model to control an external mandated has involved companies in multiple shade management of data, not only between the outsiders and the customers but also within different sectors of the same company. The concept of digital outsourcing[7] is owned here to boost the competitive advantage with regards to content delivery and payment delivery to the respective parties. This concept of useful encryption was developed to solve the shortcoming of public key encryption in a variety of rising application that requires data storage and contribution.

Consider that Abbey, who is a user of cloud storage uses a application with limited resources. Abbey now wants to obtain the encoded data from the cloud but cannot perform bulk operation of decoding due to the limited resources. FE-OD system such as attribute based encryption[8] and identity-based encryption[9] with outsourced decoding is the simple solution to this problem. These schemes help the user(i.e; Abbey) to outsource most of the computation to the powerful outsider without disclosing any sensitive data from the original file. Providing such a service the outside party would expect to be paid from the user.

Undertaking the notion of of digital outsourcing we will try to understand the concept of FE-POD scheme by elaborating the above example: Abbey promises that "I will pay $1 to the content provider when he gives me the accurate result of this task". Abbey may send this computation task to the cloud. The computation task is visible to every other user of the

cloud. A user say David received the message and is willing to perform the computation and send Abbey the result.

RELATED WORK

J. A. Akinyele et al[1] illustrates a charm framework for swift prototyping the cryptographic system. This framework provides various properties which supports the updated protocols which incorporates aids composition of building blocks of cryptography, architecture for new protocols and large library of code. Such architecture is built over as many as 40 cryptographic system using charm which have never been implemented in practice. This charm framework also includes a built-in benchmarking to compute the execution of earlier charm which uses c implementation. In many cases this infrastructure results in drop of code size which impact the acceptable performance. However, this framework is open source to research community. The major drawback of using charm framework is that it is unable to schedule urgent correction. It lacks to fix dependencies among shift and change request. The user would have to do a abundance of urgent corrections in order to accomplish the above computations.

However, it include some of the drawbacks like:

- Use of C language which needs constant run time checking and which contains lack of exception handling.
- Urgent calls which result in emergency pause for whole execution.
- No dependencies which results in improper maintainability and testibility.

Applebaum and E. Kushilevitz, [2] shows the issues of verifiable computation where a week client request the computationally strong server to perform computation of function (f) on an input x. However this computationally strong server is untrusted. To overcome this issue here a new approach for implementing verifiable computation and also providing the solution for program verification is done. Hence this notion helps decrease the job of verifiable computation to alternatives of multiparty computation protocols. Here this protocol shows how to convert multiple party protocols to durable verifiable computation protocols through message authentication code.

In practice, the following applications are obtained:

1. The verifiable computation protocol uses black-box operation for automatic computation.
2. A non-iterative verifiable computation protocol is used to carry out boolean operations in the preprocessed design.

Unfortunately it also includes some of the drawbacks:

1. Since MPC encryption is inefficient in practice and requires the server to perform heavy computation, the verifiable computation protocols cannot be implemented in practice.
2. Verifiable computation protocols can only be based on weaker cryptographic hardness assumption.

W. Banasik and D. Malinowski[3] uses A well organized zero-knowledge contingent payments in blockchain. One of the attractive innovations given by bitcoins are smart contract, which is digital contract between two parties. This execution is carried out by the mechanism of digital currency in the blockchain. The protocols are drafted in the form of "scripts" which is a segment of code in "scripting language". However smart contracts have bright futures but momentarily it is not used in practice. Bitcoin miners only use non-trival transaction in blockchain. But the non-trival transactions are difficult to be created in blockchain. Motivated by this, the following question arises:

"Can non-trival transaction be created using standard transaction only in the smart contracts?". The answer is the usage of zero knowledge contingent payment protocol which is used in huge classes of NP relation. This protocol sells a factorization (a,b) of RSA module where n=ab and n=ba which is constructed and verified efficiently in practice.

This framework includes some of the drawbacks, some of them are outlined below:

1. First is the strike to allow consumer to learn the information about the item being sold without the payment. Also here the customer is allowed to select the common parameters that actually should be selected by trusted third party.
2. This protocols are only suited for purchase of goods but not for buying of the digital services. Due to this limitations the seller do not receive appropriate payment after proving that services have been provided.

D. Boneh, and B. Waters[4] uses functional encryption and explains its challenges. In functional encryption plain text is ciphered by appending a secret key which does not allow a common person to have knowledge about the plaintext. Security of the plaintext is initiated precisely by encrypting the data. This encryption has restricted secret key which allows the user to learn only the restricted function, without learning anything else about it. However the main limitation of using such encryption is that it can be satisfied on random Oracle design but cannot be satisfied in the standard design. But it explain how to plan many existing system to formalize functional encryption and also includes several interesting open concepts in this young area:

Drawbacks of using functional encryptions are:
1. As complexity of the access data increases its size and decryption time increases.
2. The main drawback of function encryption is that the decryption is “all or nothing”.

PROPOSED SYSTEM

We represent the concept of function encryption with payable outsource decoding(FE-POD) which allows anybody to examine the the accuracy of the solution provided by the outside organisation such that the payment is carried out by the blockchain based smart contracts. We describe FE-POD scheme ensuring the security and constructed a generic model .We generate a secure model of FE-POD system that is obtained from the general model, over the blockchain platform to calculate its viability and practicality. To enable the execution of agreement the smart contracts are built on the blockchain in the form of cryptocurrencies. The first blockchain that support the smart contract is Ethereum. Briefly speaking a smart contract is a segment of program code executing on blockchain. This executions are activated by events eg; the transactions are included in blockchain only if the accuracy is assured via the agreement protocol of the blockchain. Undertaking the the notion of digital outsourcing the smart contracts are helpful in avoiding traditional methods of outsourcing. In an ideal way, the smart contract states that user can claim a proposed award once he provides the correct solution.

SYSTEM ARCHITECTURE

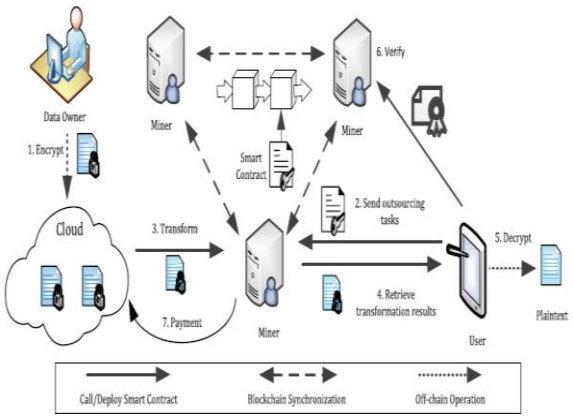

Figure 1: System Architecure

METHODOLOGY

The FE-POD framework consists of various types of principles, parameters and methods each plays a specific role:
Data User: It describes client's access right. The user makes request to perform certain computation and pays for the requested computation in the form of cryptocurrencies. The user promises that he will pay certain amount of cryptocurrencies if any data owner provides the accurate result for his requested task.
Data owner: Data owner is the one who process the data. Once the request from the user is uploaded to the cloud, data owner provides the requested content to the data user.

- Cloud: Agreement are used to check the operation on the cloud displays in a natural way. The cloud operator uses IAAS services to establish organisation which is private to various users of the application.
- Cloud: Agreement are used to check the operation on the cloud displays in a natural way. The cloud operator uses IAAS services[7] to establish organisation which is private to various users of the application.
- Miner: Miner is the trusted third party which act as an intermediary between content provider and data user . It provides a fair payment between the participant with the help of an emerging technology called blockchain.

SHA-256 ALGORITHM

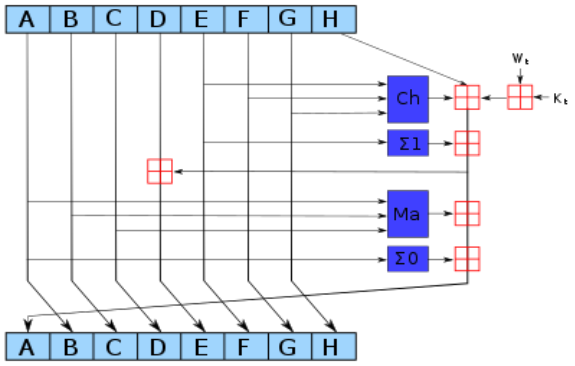

Figure 2: SHA-256 Algorithm

General explaination:
SHA-256 algorithm is a keyless cryptographic hash function where the message is 512 bits wide and each block possesses 64 iterations.
General operations:

1. Boolean operations: OR, AND and XOR operations indicated as ∨, ∧ and ⊕.
2. Bitwise complement indicated as –

Functions:
The functions used in algorithm are:
Ch(A,B,C) = (A ∧ B) ⊕ (AB∧ C),
M a(A,B,C) = (A ∧ B) ⊕ (A ∧ C) ⊕ (B ∧ C),
Σ0(A) = Rotate_Right(A, 2) ⊕ Rotate_Right(A, 13) ⊕ Rotate_Right(A, 22),
Σ1(X) = Rotated_Right(A, 6) ⊕ Rotate_Right(A, 11) ⊕ Rotate_Right(A, 25).

IMPLEMENTATION

Figure 3: Home Page

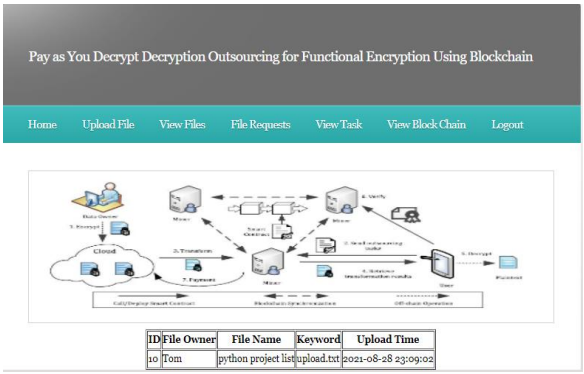

Figure 4: Uploading File

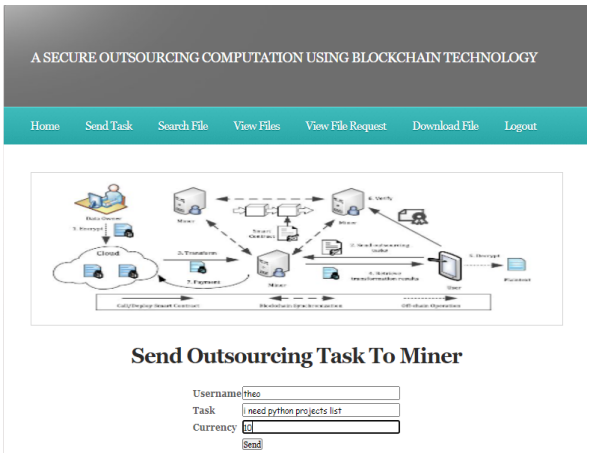

Figure 5: Requesting file to Data Owner

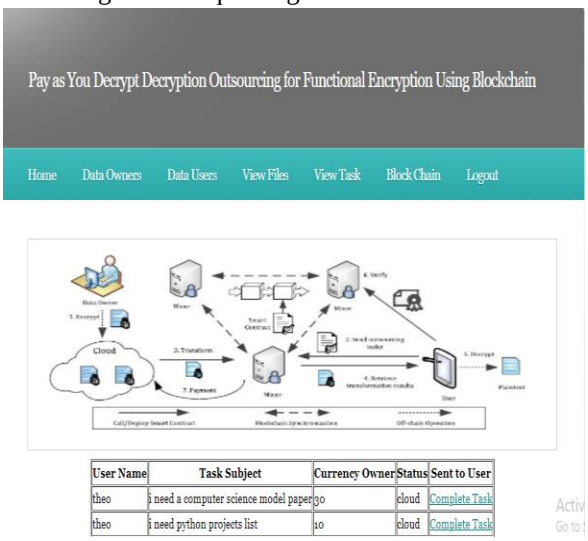

Figure 6: Miner's Task

CONCLUSION

Useful encryption has been recognised as an encryption tool to ensure information security and protection in a number of new applications, such as distributed computing administrations, due to its

advantages over open key encryption. Regrettably, the majority of current encryption plans are not competent enough to be executed in practise. The encryption with re-evaluated decoding (FE-OD'S) is designed to alleviate clients of their significant duties by entrusting most of the calculations to a outside party. Also, one problem that has been overlooked in prior encryption designs is the mechanism to held the payment between the client who communicates the re-appropriating operation duties and the outside party who conducts the re-evaluating calculation operation. In light of the above impression, we propose an openly verifiable nonexclusive functional encryption payable rethought decoding (FEP-OD'S) system that allows an outsider to be reimbursed for their assistance utilising a block-chain-based digital currency.

REFERENCES

[1] J. A. Akinyele et al., "Charm: A framework for rapidly prototyping cryptosystems", J. Cryptograph. Eng., vol. 3, pp. 111-128, Jun. 2013.

[2] Applebaum, Y. Ishai and E. Kushilevitz, "From secrecy to soundness: Efficient verification via secure computation", Proc. 37th Int. Colloq. Automat. Lang. Program., vol. 6198, pp. 152-163, Jul. 2010.

[3] W. Banasik, S. Dziembowski and D. Malinowski, "Efficient zero-knowledge contingent payments in cryptocurrencies without scripts", Proc. 21st Eur. Symp. Res. Comput. Secur., vol. 9879, pp. 261-280, Sep. 2016.

[4] D. Boneh, A. Sahai and B. Waters, "Functional encryption: Definitions and challenges", *Proc. 8th Theory Cryptogr. Conf. (TCC)*, vol. 6597, pp. 253-273, Mar. 2011.

[5] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in Proc. 8th Theory Cryptogr. Conf. (TCC), in Lecture Notes in Computer Science, vol. 6597. Providence, RI, USA: Springer, Mar. 2011, pp. 253–273.

[6] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," IEEE Trans. Services Comput., vol. 10, no. 5, pp. 785–796, Sep. 2017, doi: 10.1109/TSC.2016.2520932.

[7] Y. Zhang, R. H. Deng, X. Liu, and D. Zheng, "Blockchain based efficient and robust fair payment for outsourcing services in cloud computing," Inf. Sci., vol. 462, pp. 262–277, Sep. 2018, doi: 10.1016/j.ins.2018.06.018.

[8] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute-based encryption with continuous leakage-resilience," Inf. Sci., vol. 484, pp. 113–134, May 2019, doi: 10.1016/j.ins.2019.01.052.

[9] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," IEEE Trans. Comput., vol. 64, no. 2, pp. 425–437, Feb. 2015.