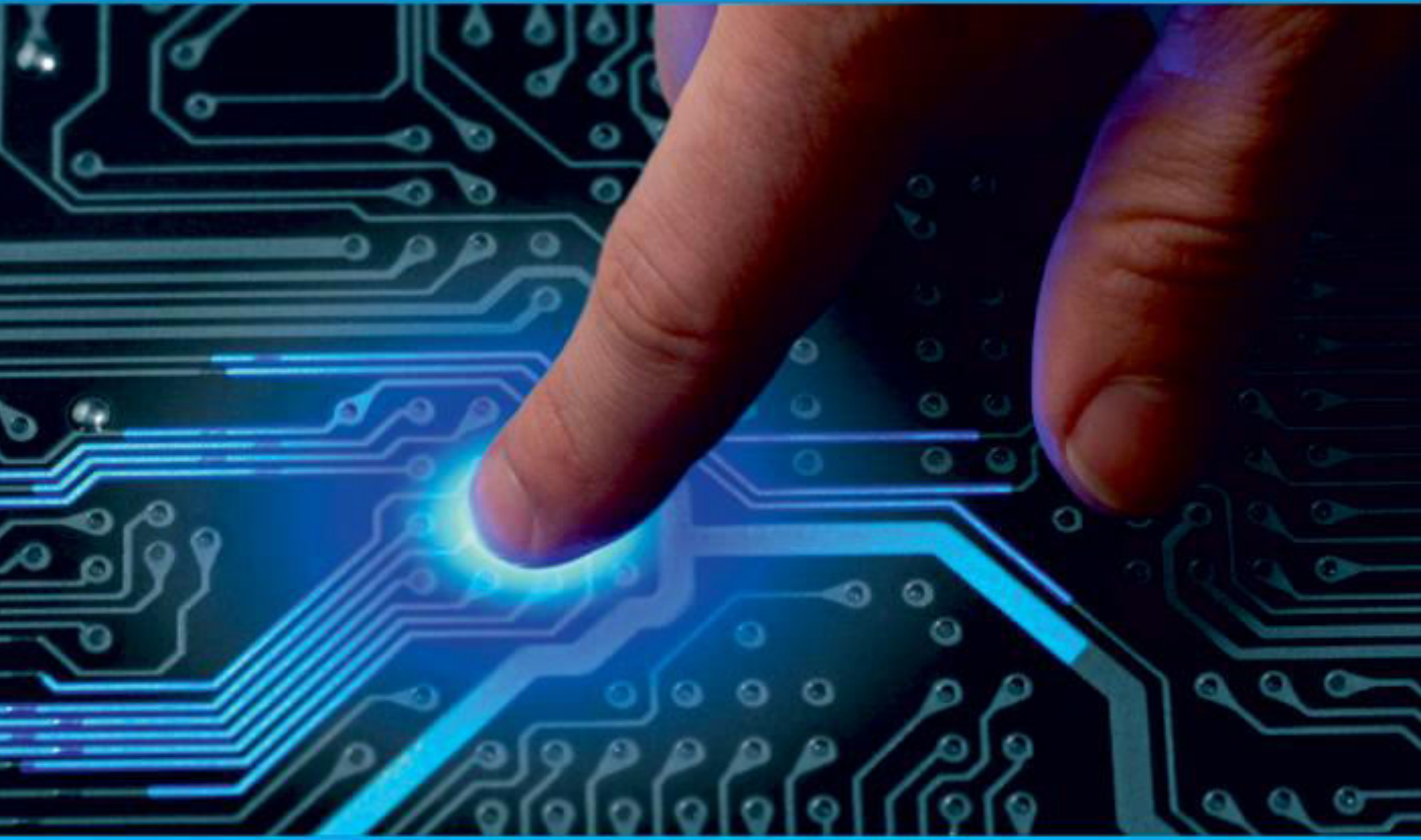
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

| DOI: 10.15680/IJIRCCE.2024.1204141 |

Functional Encryption with Payable Outsourced Decryption

K. Jyothi, Sireesha Dasari, Pavan Gorre, Gudipati Saketh Kasyap

Assistant Professor, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

UG Student, Dept. of CSE, Anurag University, Hyderabad, India

ABSTRACT: Functional encryption (FE) has emerged as a solution for addressing the limitations of public-key encryption (PKE) in various modern applications, such as cloud storage services, where both data storage and sharing are essential. However, many existing FE schemes suffer from inefficiency due to their reliance on expensive bilinear pairings for computation. A common approach to mitigate this inefficiency is to delegate heavy computational tasks to a trusted third party, while the user handles lighter computations. However, assuming this third party, like a cloud service provider, offers its services for free is impractical. Despite the importance of addressing this issue, little attention has been given to the payment process between the user and the third party in a Functional Encryption with Outsourced Decryption (FEOD) scheme, especially under the assumption that neither party can be fully trusted. In this study, we propose a novel approach called Functional Encryption with Payable Outsourced Decryption (FEPOD), leveraging blockchain-based cryptocurrencies for payment transactions. By utilizing the transparency and security features of blockchain technology, FEPOD allows users to compensate a third party for completing outsourced decryption tasks accurately. We establish an adversarial model for FEPOD schemes and present a generic construction for such schemes. Additionally, we conduct performance evaluations of our proposed generic construction by implementing a concrete FEPOD scheme on a blockchain platform. These advancements not only enhance the efficiency and practicality of FE schemes but also contribute to the broader adoption of blockchain technology in cryptographic protocols for secure and decentralized payment mechanisms.

KEYWORDS: Cloud storage, blockchain, outsourced decryption.

I. INTRODUCTION

Consider a scenario where Alice, a privileged user of a cloud storage application, lacks the computational resources on her device to decrypt encrypted data stored in the cloud. This challenge can be addressed through a Functional Encryption with Outsourced Decryption (FEOD) scheme, such as Identity-Based Encryption (IBE) or Attribute-Based Encryption (ABE) with outsourced decryption. In this setup, Alice can delegate the decryption computation to a capable third party, like Bob, and offer payment for the service.

However, ensuring fair payment and verification of Bob's computation presents two key challenges. Firstly, Alice needs a mechanism to verify Bob's correctness before payment, and secondly, Bob requires assurance that Alice won't deny the correctness of his work to evade payment. Traditional solutions involving trusted intermediaries, like banks, compromise user privacy as transactions become transparent to the authority.

Blockchain technology offers a decentralized and self-enforcing alternative for fair payment between Alice and Bob. Platforms like Ethereum enable the execution of smart contracts, facilitating secure transactions without the need for intermediaries. Alice can create a smart contract on the blockchain, detailing the computation task and payment rules. The blockchain acts as a trusted third party, holding funds in escrow until the task is completed satisfactorily. Bob, too, deposits funds as collateral against malicious behavior.

However, a challenge remains in verifying Bob's solution without compromising encryption security. Traditional methods, like setting a "trapdoor" for efficient verification, are not suitable in a blockchain environment due to inefficiencies and trust issues. Therefore, the design of Functional Encryption with Payable Outsourced Decryption (FEPOD) schemes becomes crucial.

In this paper, we propose a generic construction for FEPOD schemes, leveraging blockchain-based cryptocurrencies for payment. Our focus lies in enabling efficient and public verification of outsourced decryption outcomes. An ideal FEPOD scheme allows anyone to verify the correctness of the result solely based on public information. This approach enhances the efficiency and trustworthiness of outsourced decryption, addressing the needs of both users and service providers in cloud storage and similar applications.

II. RELATED WORK

Outsourcing Computation. Outsourcing computation enables a user to delegate the heavy computation, especially pairing operations, to a third entity [4], [10]. Such a method has been widely used in cryptographic primitives such as digital signature schemes (e.g., [9], [10]), identity-based encryption (IBE) schemes (e.g., [11], [13]) and attribute based encryption (ABE) schemes (e.g., [4], [8]) to handle heavy computation workloads for resource-constrained devices.

The main security issue in existing outsourcing computation schemes is the correctness check for the result returned by the third party [10] who is not trusted and have the incentive to provide a false answer. Due to this observation, many outsourcing computation schemes with verification functions have been proposed (e.g., [4], [8], [16]) in terms of either security or efficiency improvement. Unfortunately, these existing solutions fail to work for the scenario of the payable outsourced decryption considered in this paper. In these schemes, the public verification is achieved by making the outsourced computation re-doable by the third party, so they fail to enable the verification to be efficiently conducted by any third party. Zero-Knowledge Contingent Payments.

Through the zeroknowledge contingent payment (ZKCP) [2] protocol, a seller is able to sell any piece of information that can be verified for a payment in Bitcoin. But when it is the service rather than the information to be sold, the ZKCP fails to achieve the fair payment. To remedy this limitation, the ZKCP protocol is extended to the zero-knowledge contingent service payment (ZKCSP) [3] for a seller to sell some assurance to a buyer.

Unfortunately, the ZKCSP scheme based on the witness indistinguishable (WI) proof is insecure [5]. The aim of succinct non-interactive arguments (SNARGs) and SNARGs of knowledge is to provide an efficient way to verify the work that has been outsourced to an untrusted entity, of which the solutions have been proposed in various kinds of settings [1], [6], [7], [15], depending on whether the verification should be public or private. In publicly verifiable computation protocols, to assign a computation task on an input x to a third party, the assigner should be able to generate a verification key vkx such that other entities are able to obtain the answer from the third party and check the correctness of the given answer. There are a few publicly verifiable SNARGs (e.g., [14], [15]) that enables anyone who receives the result to verify the correctness. They can be applied in designing the payment mechanism for the blockchain-based payable outsourced decryption. But their efficiency is undesirable when applying to complex FE schemes due to the generation of ZK proofs. We are seeking a solution to achieve the fair payment without using ZK proofs in this paper.

III. PROPOSED SYSTEM

As shown in Fig. 1, a functional encryption with payable outsourced decryption (FEPOD) scheme over a blockchain involves data owners, users (i.e., devices with restrained resources), a cloud (i.e., a powerful server) and miners. Data owners (note that the data owner could be a user as well if he/she empowers the decryption capability to himself/herself) encrypt their data items via one functional encryption (FE) scheme, and upload the resultant ciphertexts to the cloud to protect the data security and privacy. Users are able to acquire the original messages of the ciphertexts in the cloud in accordance with their access rights. The cloud is in charge of the storage of the data items for data owners, and may help users with the heavy computation (if necessary). The miners play the role of monitoring and processing the transactions over the blockchain.

Assume that a user, say Alice, who has access to an amount of ciphertexts stored on the cloud, intends to obtain the plaintext of a ciphertext CT with the help of the cloud on the heavy computation without compromising the security and privacy of the original data. Alice sends the outsourcing computation task, in the form of a smart contract, to the blockchain along with two public keys (of which one is the transformation key and the other is the verification-key). Alice also deposits a certain amount of cryptocurrencies to this smart contract, and specifies that “whoever posts the correct result can claim the reward (e.g., $1)”.

Anybody is able to check the blockchain and find the outsourced computation task. If a cloud, say Bob, is willing to take this task, he deposits some cryptocurrencies to the smart contract. Bob then executes the computation to produce a transformed ciphertext CTA, and submits the transformed ciphertext CTA to the smart contract. Alice can retrieve and decrypt the transformed ciphertext CTA using her decryption key. Thereafter, Alice indicates whether the given transformed ciphertext CTA is correct as "1" or "0", and releases the proof regarding the correctness of the public keys and the transformation result CTA such that miners are able to determine whether Bob should be paid for his work. If Alice indicates "1", Bob will be returned his deposit and receive the payment immediately. If Alice indicates "0", the miner verifies the correctness of the transformed ciphertext CTA.

If the transformation result CTA passes the verification, the miner will perform the transformation to yield the transformed ciphertext CT A. If CT A equals to CTA, the miner outputs, meaning that Bob will be paid and Alice will pay more transaction fees (than that in a normal situation). Otherwise, the miner outputs ⊥, meaning that Bob (rather than Alice) will lose a part of his deposit to pay for the transaction fee and Alice will be returned her deposit. If the transformation result CTA fails to pass the verification, the miner checks the validity of the public keys. If the public keys are not well formed, the miner outputs "1", meaning that Bob will be paid and returned his deposit. Otherwise, the miner outputs "0", meaning that the transaction fails without any payment. Note that if in a certain time period (e.g., one hour), Alice does not send any proof regarding the correctness of the result CTA, the payment to Bob will be proceeded by default. In general, an FEPOD scheme is composed of the following algorithms: a setup algorithm Setup, a function key generation algorithm PrivKG, a transformation key generation algorithm TranKG, a verification key generation algorithm VeriKG, an encryption algorithm Encrypt, a transformation algorithm Transform, a decryption algorithm Decrypt and a verification algorithm Verify.

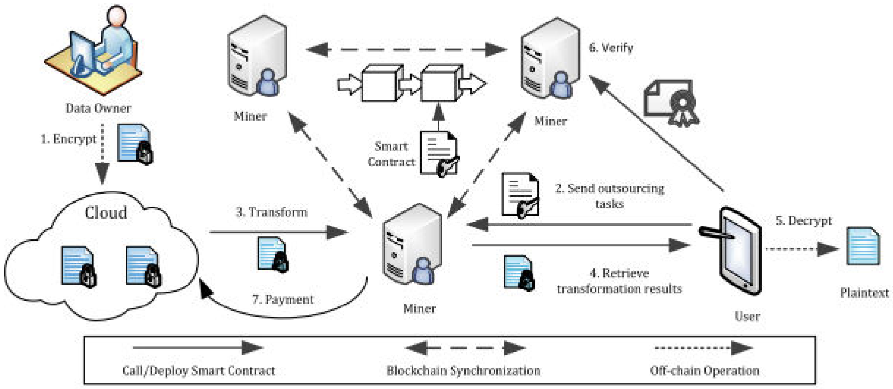


Fig 1: A pictorial architecture for an FEPOD scheme over a blockchain.

IV. SIMULATION RESULTS

Prior to implementing the Blockchain-based Attribute-Based Encryption with Verifiable Outsourced Decryption (BABEPOD) protocol, we first implement the proposed Attribute-Based Encryption with Verifiable Outsourced Decryption (ABEPOD) scheme using Charm (version 0.50), a Python-based rapid prototyping framework for cryptographic schemes. With Charm, we efficiently implement all cryptographic algorithms of ABEPOD using approximately 300 lines of Python code. Additionally, we compose a smart contract in about 150 lines of Solidity, Ethereum's dedicated language for smart contract composition, to realize the BABEPOD protocol. Notably, Ethereum lacks native support for bilinear pairings over elliptic curves, thus requiring modifications to enable miners to verify transformation results using the PBC (Pairing-Based Cryptography, version 0.5.14) library. Interactions with the smart contract on Ethereum are facilitated through the Python library web3, enabling Remote Procedure Call (RPC) interactions. The implementation of BABEPOD protocol itself is also carried out using Python.

Our experimentation environment consists of a laptop with an Intel Core i5-8250U processor @ 1.6GHz*4 and 24GB RAM running 64-bit Ubuntu 18.04 as the miner, a Raspberry Pi 3B device with a Quad-Core 1.2GHz Broadcom 64-bit CPU and 1GB RAM running Raspbian as the user, and an Aliyun ecs.c5.xlarge cloud server equipped with 4 vCPU and 8GB RAM as the cloud. We initially conduct experiments on the laptop to assess the average computation time for each algorithm of the ABEPOD scheme without involving the blockchain. The results show that the computation overheads for generating private attribute-keys, public transformation keys, and public verification keys are roughly linear to the number of attributes, with costs under 1 second for 50 attributes, validating practical feasibility (see Fig. 3).

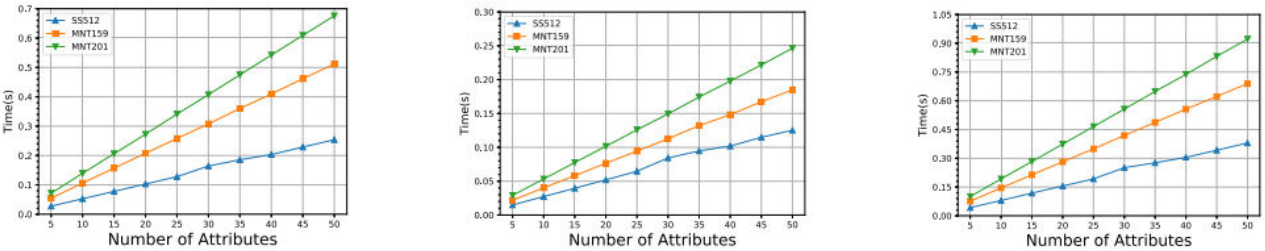
Furthermore, we evaluate the average computation costs for decryption and verification algorithms, which are independent of the number of attributes. For an access structure with 30 attributes, decryption and verification costs are under 1 millisecond across various security curves, affirming efficiency (see Fig. 4). The computation costs for Encrypt and Transform algorithms are found to be linearly proportional to the number of attributes associated with access structures, aligning with the theoretical expectations outlined in Table III. Subsequently, we assess the performance of each step of the BABEPOD protocol. Utilizing Ethereum's fast consensus algorithm Proof of Authority (PoA) and fixed attributes and access structures, we find that setup, task publication, and payment settling processes are highly efficient, taking less than 75 milliseconds for 1000 ciphertexts in total. Transformation processes exhibit longer durations, but decryption is notably faster. The overall protocol delay is approximately 90 milliseconds per ciphertext, demonstrating practical usability (see Fig. 7).

Finally, we test the BABEPOD protocol under scenarios where either the cloud or the user might act maliciously. The average computation costs for miners under both scenarios are presented, along with a summary of gas costs in Ethereum for different protocol steps. Notably, additional computation is required for the miner to regenerate transformed ciphertexts or verification keys to settle payments, ensuring security against malicious behavior (see Fig. 8 and Table IV). These experiments demonstrate the efficiency and practicality of the ABEPOD and BABEPOD protocols in real-world scenarios, highlighting their potential for secure and verifiable outsourced decryption in cloud environments.

TABLE III

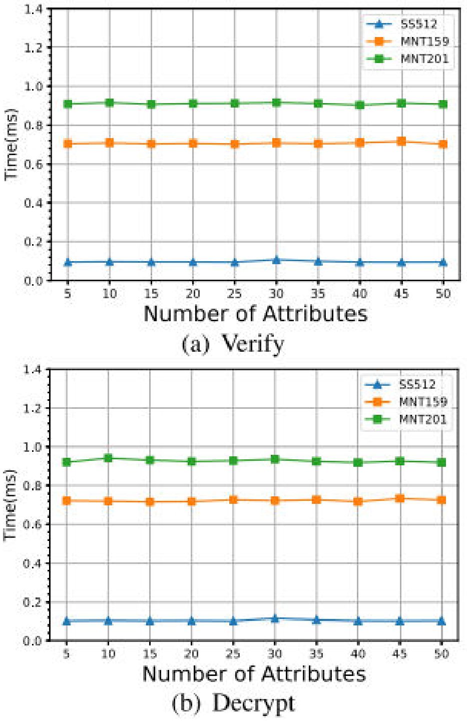
THE COMPUTATION COSTS OF THE UNDERLYING ABE SCHEME [33], THE PA-ABE SCHEME [12] AND THE GIVEN ABEPOD SCHEME, WHERE "-" MEANS "NOT APPLICABLE". k DENOTES THE NUMBER OF ATTRIBUTES ASSOCIATED WITH A PRIVATE ATTRIBUTE-KEY, AND l DENOTES THE NUMBER OF ATTRIBUTES IN AN ACCESS STRUCTURE. "E" DENOTES EXPONENTIATION OPERATION, AND "P" DENOTES PAIRING OPERATION

	TranKG User	VeriKG User	Transform Cloud	Decrypt User	Verify Miner
ABE [33]	-	-	-	$\leq (3k+1)\cdot \mathrm{P} + k\cdot \mathrm{E}$	-
PA-ABE [12]	$(4+3k)\cdot \mathrm{E}$	-	$\leq (3k+1)\cdot \mathrm{P} + k\cdot \mathrm{E}$	E	-
ABEPOD	$(2+2k)\cdot \mathrm{E}$	$(5+5k)\cdot \mathrm{E}$	$\leq (6k+2)\cdot \mathrm{P} + 2k\cdot \mathrm{E}$	E	E



(a) PrivKG (b) TranKG (c) VeriKG

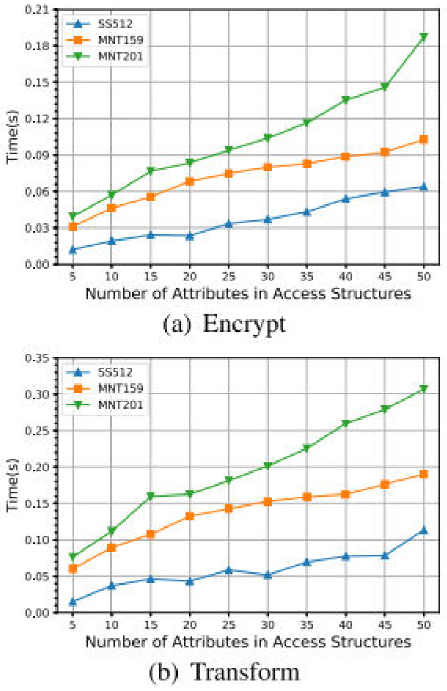
Fig. 3. The average computation costs of the PrivKG, TranKG and VeriKG algorithms on different numbers of attributes for different elliptic curves in the scheme ABEPOD.



(a) Verify

(b) Decrypt

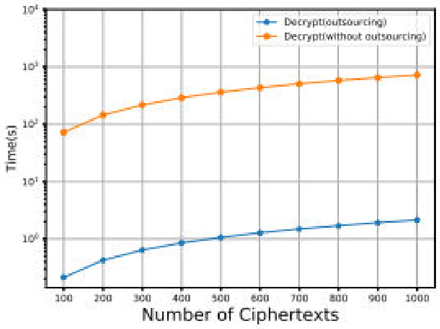
Fig. 4. The average computation costs of the Encrypt, Verify and Decrypt algorithms in terms of different number of attributes, with the access structure with 30 attributes, for different elliptic curves in the scheme ABEPOD.



(a) Encrypt

(b) Transform

Fig. 5. The computation costs of the Encrypt and Transform algorithms in terms of different numbers of attributes in access structures for different elliptic curves in the scheme ABEPOD.



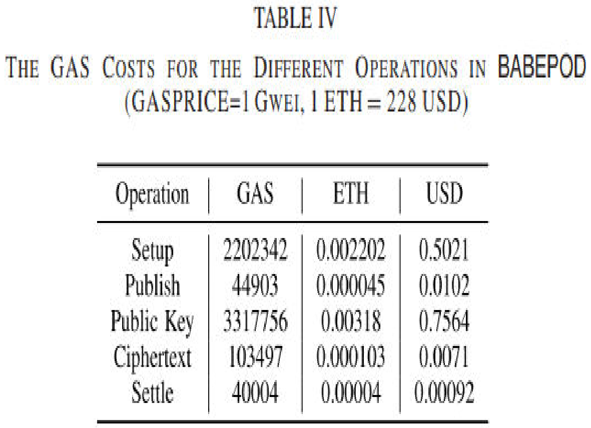


TABLE IV

THE GAS COSTS FOR THE DIFFERENT OPERATIONS IN BABEPOD (GASPRICE=1 GWEI, 1 ETH = 228 USD)

Operation	GAS	ETH	USD
Setup	2202342	0.002202	0.5021
Publish	44903	0.000045	0.0102
Public Key	3317756	0.00318	0.7564
Ciphertext	103497	0.000103	0.0071
Settle	40004	0.00004	0.00092

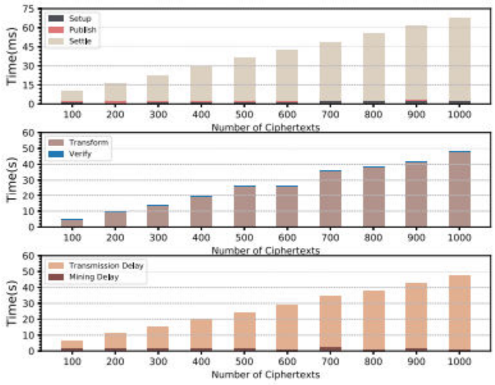


Fig. 6. The average computation cost of the Decrypt algorithm for the Raspberry Pi device (with the help of a laptop) in terms of different numbers of ciphertexts in the scheme ABEPOD, where both the number of attributes and the size of access structures are set to be 30.

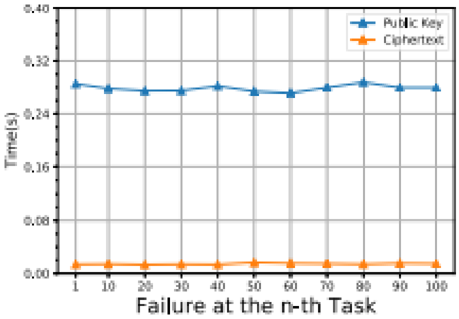


Fig. 7. The average computation cost of the protocol BABEPOD in terms of different numbers of ciphertexts, where both the number of attributes and the size of access structures are set to be 30.

Fig. 8. The average computation cost of the miner under the assumption that the participants in the protocol BABEPOD may be malicious when the termination of the protocol BABEPOD happens, where the number of ciphertexts is set to be 100.

V.CONCLUSION AND FUTURE WORK

Given its superiority over traditional public-key encryption (PKE), functional encryption (FE) has emerged as a promising encryption mechanism for safeguarding data security and privacy, particularly in applications like cloud computing services. However, the practical adoption of FE has been hindered by the inefficiency of existing schemes. To address this challenge, the concept of FE with outsourced decryption (FEOD) has been proposed, aiming to offload computation-intensive tasks to third-party providers. Yet, a crucial aspect often overlooked in previous FEOD schemes is the payment process between the user outsourcing the computation and the third-party executor.

With this gap in mind, our paper introduces a novel approach: a generic FE with payable outsourced decryption (FEPOD) scheme, designed to facilitate payment to third-party service providers through blockchain-based cryptocurrencies while ensuring public verifiability. After establishing the security guarantees of the proposed FEPOD construction, we outline the process of integrating FEPOD into a blockchain framework. Furthermore, we implement a concrete instantiation of FEPOD over a blockchain platform to assess its practical efficiency.This contribution not only addresses the payment challenge in FEOD schemes but also leverages blockchain technology to enhance transparency and security in transaction settlements. Through theoretical analysis and empirical evaluation, we demonstrate the feasibility and effectiveness of FEPOD in real-world scenarios, underscoring its potential to advance the adoption of functional encryption in practical applications.

REFERENCES

[1] B. Applebaum, Y. Ishai, and E. Kushilevitz, "From secrecy to soundness: Efficient verification via secure computation," in Proc. 37th Int. Colloq. Automat., Lang., Program., in Lecture Notes in Computer Science, vol. 6198. Bordeaux, France: Springer, Jul. 2010, pp. 152–163.

[2] W. Banasik, S. Dziembowski, and D. Malinowski, "Efficient zeroknowledge contingent payments in cryptocurrencies without scripts," in Proc. 21st Eur. Symp. Res. Comput. Secur., in Lecture Notes in Computer Science, vol. 9879. Heraklion, Greece: Springer, Sep. 2016, pp. 261–280, doi: 10.1007/978-3-319-45741-3_14.

[3] M. Campanelli, R. Gennaro, S. Goldfeder, and L. Nizzardo, "Zeroknowledge contingent payments revisited: Attacks and payments for services," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Dallas, TX, USA, 2017, pp. 229–243, doi: 10.1145/3133956.3134060.

[4] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attributebased encryption," in Proc. 21st Eur. Symp. Res. Comput. Secur., in Lecture Notes in Computer Science, vol. 9879. Heraklion, Greece: Springer, Sep. 2016, pp. 570–587.

[5] G. Fuchsbauer, "WI is not enough: Zero-knowledge contingent (service) payments revisited," in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., London, U.K., Nov. 2019, pp. 49–62, doi: 10.1145/3319535.3354234.

[6] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in Proc. 30th Annu. Cryptol. Conf., in Lecture Notes in Computer Science, vol. 6223. Santa Barbara, CA, USA: Springer, Aug. 2010, pp. 465–482.

[7] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: Interactive proofs for muggles," J. ACM, vol. 62, no. 4, pp. 27:1–27:64, 2015.

[8] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in Proc. 20th USENIX Secur. Symp., San Francisco, CA, USA, Aug. 2011, pp. 1–16.

[9] M. Jakobsson and S. Wetzel, "Secure server-aided signature generation," in Proc. 4th Int. Workshop Pract. Theory Public Key Cryptogr. (PKC), in Lecture Notes in Computer Science, vol. 1992. Cheju Island, South Korea: Springer, Feb. 2001, pp. 383–401.

[10] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[11] 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.

[12] C. H. Lim and P. J. Lee, "Server (prover/signer)-aided verification of identity proofs and signatures," in Proc. Int. Conf. Theory Appl. Cryptograph. Techn., Saint-Malo, France, May 1995, pp. 64–78.

[13] J. K. Liu, C. Chu, and J. Zhou, "Identity-based server-aided decryption," in Proc. 16th Australas. Conf. (ACISP), in Lecture Notes in Computer Science, vol. 6812. Melbourne, VIC, Australia: Springer, Jul. 2011, pp. 337–352.

[14] C. Papamanthou, R. Tamassia, and N. Triandopoulos, "Optimal verification of operations on dynamic sets," in Proc. 31st Annu. Cryptol. Conf., in Lecture Notes in Computer Science, vol. 6841. Santa Barbara, CA, USA: Springer, Aug. 2011, pp. 91–110.

[15] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in Proc. 9th Theory Cryptogr. Conf. (TCC), Taormina, Sicily, Italy, Mar. 2012, pp. 422–439.

[16] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," IEEE Trans. Inf. Forensics Security, vol. 10, no. 7, pp. 1384–1393, Jul. 2015.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

📱 9940 572 462 ✆ 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details