International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

FFPBO_KEYGEN: FLOWER FERTILIZATION PAPER-BASED OPTIMIZATION ENABLED KEY GENERATION FOR BLOCKCHAIN-ASSISTED TRANSPARENCY FRAMEWORK FOR PRIVACY PRESERVATION OF THIRD-PARTY SERVICES

Pathakamuri Srinivasulu[1], Dr.B.Venkata Ramana Reddy[2], Dr.A.P.Siva Kumar[3]

[1]Research Scholar, J N T University, Anantapur, mail id: srinivasulu.p@visvodayata.ac.in

[2]Professor, Dept. of CSE, Golden Valley Integrated Campus, Madanapalli, mail id: busireddy100@gmail.com

[3]Professor, Dept. of CSE, J N T University, Anantapur, mail id: sivakumar.AP@gmail.com

Abstract

Preserving privacy in third-party services has become increasingly crucial as reliance on external platforms for data processing, storage, and analytics grows. While classical privacy-preserving models, such as access control, encryption, and anonymization techniques provide foundational protection, and they are not effective in dynamic and complex service environments. Typically, these traditional approaches lack adaptability, computationally intensive, and failed to prevent indirect privacy leaks through inference attacks or data correlation. To address the complexities, this work introduces a novel hybrid key generation approach based on Flower Fertilization Paper-Based Optimization enabled key generation model (FFPBO_KeyGen) within a Blockchain-assisted transparency framework for privacy preservation in third-party services. The generation of secure keys is handled by third-party authority using FFPBO, which combines Paper Publishing-Based Optimization (PPBO) and the Flower Fertilization Optimization (FFO) algorithm.

Keywords*:* Privacy preservationof third-party services, Key generation, Blockchain, Paper Publishing-Based Optimization, Flower Fertilization Optimization.

1. INTRODUCTION

In a blockchain technology, transactions are recorded within blocks that contain details, such as transaction details, timestamps, and additional metadata. This structure ensures that the blockchain offers transparency, decentralization, user anonymity, and immutability [1]. Due to its appealing features, such as transparency, anonymity, autonomy, and tamper-resistance, blockchain technology has found extensive applications in voting systems, supply chains, healthcare, and more. Within a blockchain, all transactions are verified via a consensus mechanism in an untrusted environment, ensuring that no participant can arbitrarily alter the data. Blockchain is resilient against failures and malicious attacks. Their security relies on the majority of computing power within the network, rather than on a central authority [3].The

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
blockchain utilizes cryptographic algorithms to guarantee data integrity, security, standardized auditing, and support for smart contracts to control data access. Additionally, encryption techniques can be employed to protect the confidentiality and security of the data [4]. Its security depends on mechanisms, such as data encryption, timestamping, distributed consensus, and incentive structures, rather than relying on a Trusted Third Party (TTP) [5].

As data volumes continue to grow rapidly, users find it challenging to store, manage, and process all their data independently without relying on third-party services. Therefore, maintaining a high level of trust in these third-party providers particularly those handling sensitive user informationbecomes essential [7]. blockchain solutions face various challenges, including compliance with legal regulations like the General Data Protection Regulation (GDPR), scalability and response time limitations, security threats, and privacy concerns. Key concerns include transaction traceability, adherence to data protection regulations, on-chain data privacy, and the threat of malicious smart contracts. To address these issues, numerous researchers in both academia and industry have used a variety of solutions aimed at enhancing the privacy and security of blockchain systems [9].

Privacy-preserving authentication approaches are employed to establish user and system verification mechanisms, including methods like single sign-on, federated identity, and key management [13]..

The main goal of this research is to develop a novel key generation approach, called FFPBO_KeyGen for privacy preservation of third-party services. The system architecture is composed of key entities, including smart contracts, Blockchain, third-party authority, data owners, and users. The proposed FFPBO_KeyGen framework is constructed by integrating the strengths of PPBO and FFO. By combining these two advanced optimization techniques, FFPBO_KeyGen aims to generate cryptographic keys for ensuring secure and reliable communication in privacy-sensitive applications.

2. MOTIVATION

In the evolving landscape of data privacy, ensuring secure and verifiable interactions between users and third-party services is critical. Despite the increasing adoption of third-party services for data processing and storage, users face significant risks related to data privacy, unauthorized access, and lack of transparency. Furthermore, existing key management solutions are centralized or lack scalability, this leads to security breaches. This section shows the review of the existing works related to privacy preservation in a Blockchain-assisted transparency framework, this helps to design secure key generation and management scheme with privacy regulations.

2.1 Literature review

R. Xu, *et al.*[16] presented a Trust, Accountability, and Balance(TAB) framework for Privacy Preserving Third-party Services. TAB offered better data integrity authenticity, and scalability. However, as the volume of transactions increased, it resulted in slower processing speeds and higher transaction costs. R. Vijay Anand, *et al.*[17] developed a Federated

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
learning and Long-Short Term Memory (LSTM) autoencoders for secure and transparent blockchain network transactions. The model's performance was enhanced while ensuring maximum transparency and integrity. Additionally, differential privacy and homomorphic encryption were utilized to achieve high accuracy with minimal impact on performance. However, the model did not incorporate optimized encryption techniques to further improve computational efficiency.U. Islam, *et al.* [18] designed a Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) units for enhancing Blockchain Security Against Data Tampering. This approach ensured efficient training and inference times. Additionally, the model achieved high accuracy while safeguarding individual privacy. However, the integration of blockchain with multi-party computation (MPC) in this model demanded substantial computational resources, resulting in increased energy consumption. P. Thantharate and A. Thantharate, [19]introduced a ZeroTrustBlock protocols for Enhancing Security, Privacy, and Interoperability of Sensitive Data. In this approach, the distributed ledger architecture removed the vulnerability of a single point of failure in conventional systems, thereby strengthening data privacy and integrity. However, it did not perform comprehensive testing across various infrastructure setups and heterogeneous blockchains, limiting its scalability potential.

To address these challenges, a hybrid optimization-based key generation method is developed to enhance privacy preservation in blockchain systems while minimizing computational complexity.

3. PROPOSED FLOWER FERTILIZATION PAPER-BASED OPTIMIZATION ENABLED KEY GENERATION MODEL FOR PRIVACY PRESERVATION OF THIRD-PARTY SERVICES

This work develops the novel FFPBO_KeyGenfor privacy preservation of third-party services in a Blockchain-assisted transparency framework. Here, entities like smart contracts, Blockchain, Transparent third-party authority, monitors, data owners, and users are considered. The transparency of third-party authority ensures its operations in such a way that it should be visible, verifiable, and understandable. Moreover, the steps like initialization, registration, authorization, key request, key generation, data protection, and access control is performed for privacy preservation. Here, FFPBO is devised by the integration of PPBO [24], and FFO [25]. Figure 1 shows the block diagram of FFPPBO-based key generation for privacy preservation in a Blockchain-assisted transparency framework.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

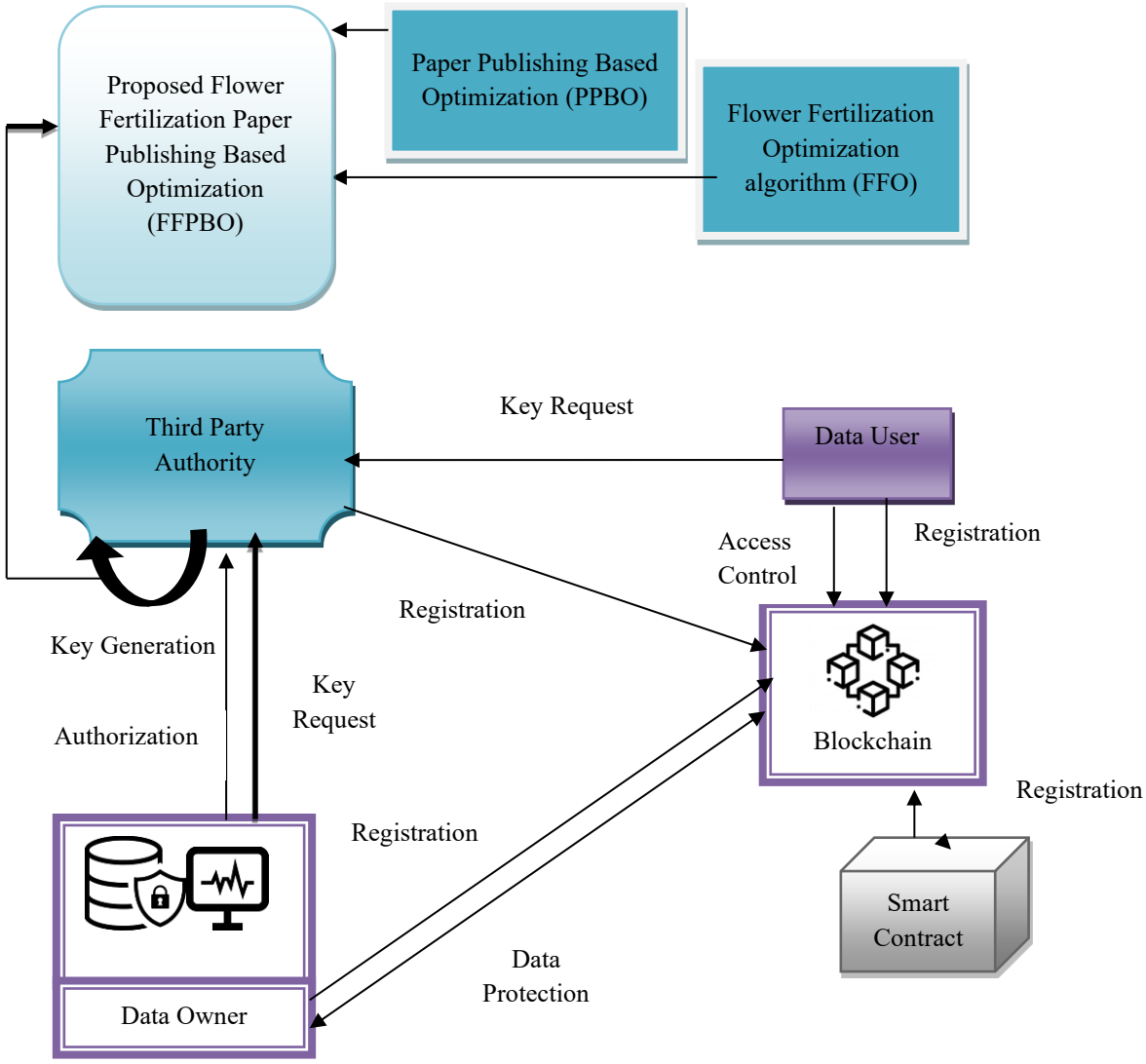


Figure 1. Block diagram of FFPBO_KeyGen for privacy preservation in Blockchain

The symbols and description used for key generation and privacy preservation in Blockchain is detailed in Table 1.

Table 1. Symbols and description

Symbol	Description
P_k	Private Key
M_1, M_2	Message
a, b, c, d	Security parameters
p, q, r, s	Random parameters

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

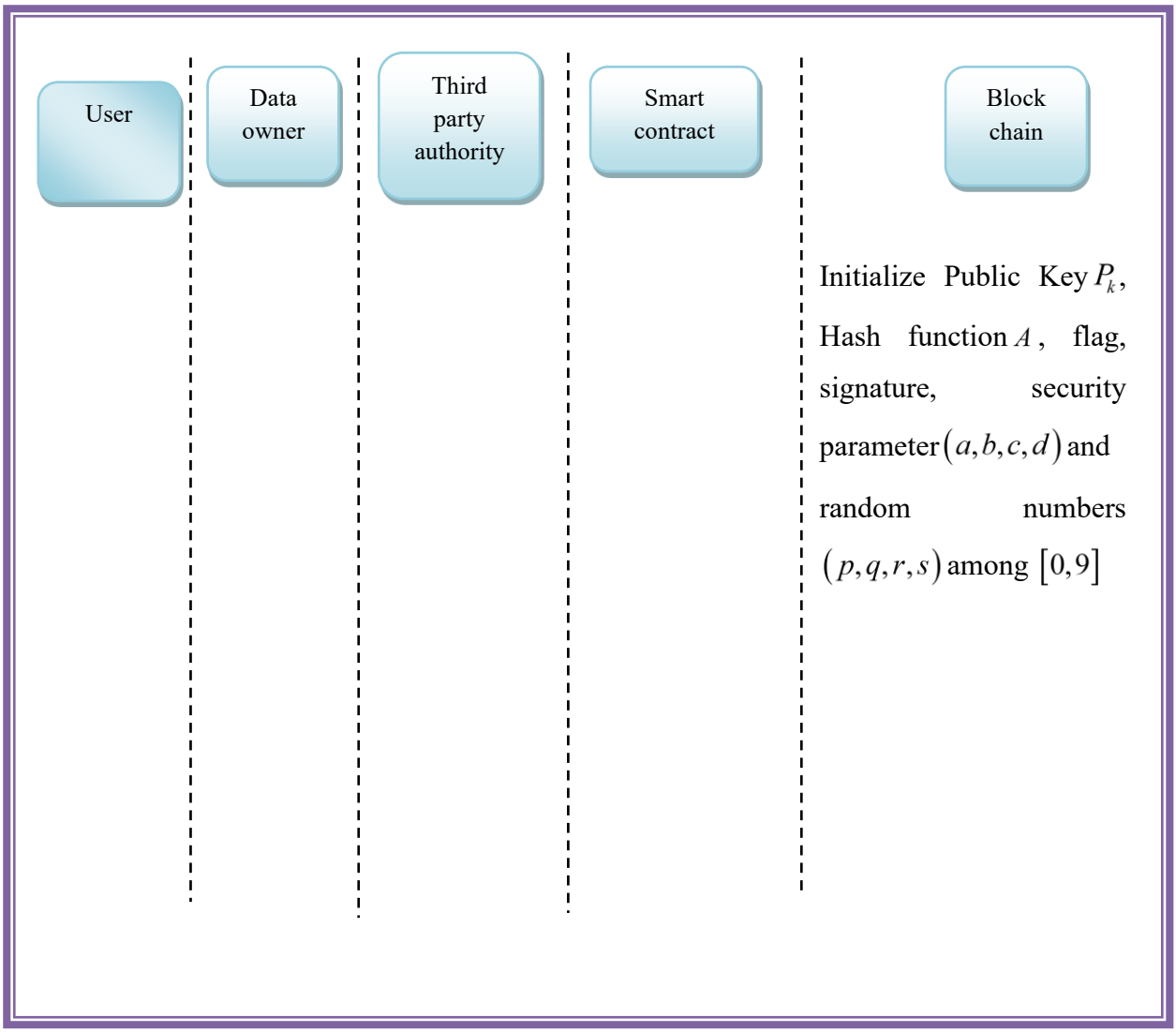
ID_e	Edge ID
pwd_e	edge password
T_{id}, T_{pwd}	ID and password of third-party authority
O_{id}, O_{pwd}	User ID and password of data owner
S_{id}, S_{pwd}	ID and Password of smart contract
U_{id}, U_{pwd}	ID and Password of user
$\oplus$	XOR function
mod	Modulation function
OTP	One Time Password
AC	Access control
En, De	Encryption and decryption
Ke, Ke_1, Ke_2	Key
RE	Authorization request
K	Key request
AM	Authorization message
$flag$	Registered entity
$\|\|$	Concatenation
$\mathrm{H}(.)$	Hash function

3.1 Initialization

The initialization phase involves setting up the necessary protocols and systems to ensure that sensitive data remains secure when a third party is granted permission to act on behalf of another. The entities involved in initialization process is user, data owner, third party authority, smart contract and blockchain. At first, the initialization ofPublic Key P_k, Hash function A, flag, signature, security parameter (a,b,c,d) and random numbers (p,q,r,s) among $[0,9]$ is done for achieving efficient privacy preservation and key generation by using the entities, like user, data owner, third party authority, smart contract and block chain. The process of initialization is depicted in Figure 2.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



3.2 Registration phase

Registration refers to the process by which a user creates an identity with other entities while minimizing the disclosure and retention of personal information. The goal is to enable access to the service without compromising user privacy. Privacy-preserving registration ensures that only essential data is collected and stored, and that users retain control over how their information is shared and used.

3.2.1 Registration between smart contract and block chain

Registration between a smart contract and the blockchain is a crucial step that establishes a secure and verifiable link between the contract and the blockchain network. During this process, the smart contract ID S_{id} and password S_{pwd} is stored in block chain as S_{id}^{*} and S_{pwd}^{*}. This registration process ensures that the smart contract becomes an integral, tamper-proof

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
part of the blockchain, capable of executing predefined functions automatically when triggered by specific conditions. This is shown in Figure 3.

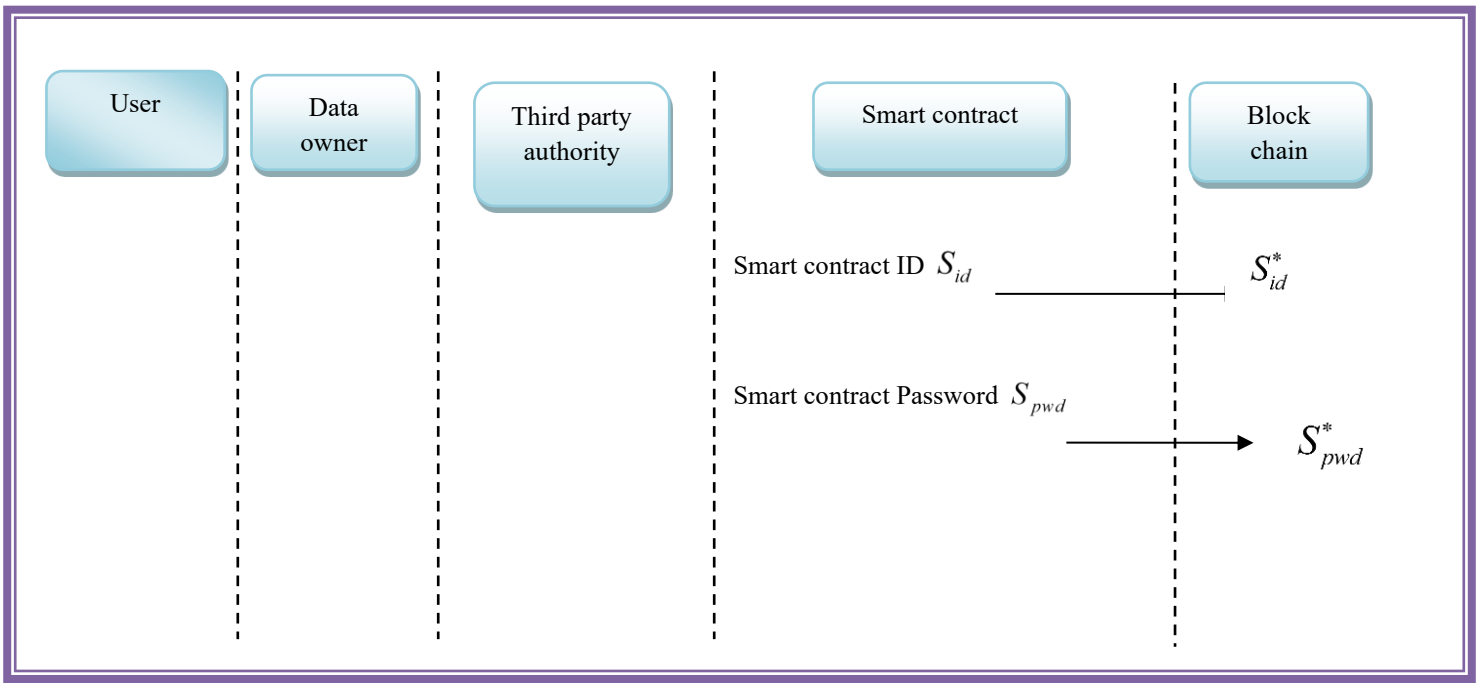


Figure 3. Registration between smart contract and block chain

3.2.2 Registration between data owner and blockchain

Registration between a data owner and a blockchain refers to the process by which the data owner establishes a verifiable and secure identity or data reference on the blockchain network. During registration, the data owner ID O_{id} and password O_{pwd} is stored in block chain and it is represented as O^*_{id} and O^*_{pwd}. On receiving data owner ID and password for registration, blockchain generates a message (M_1) and send to data owner. Here, M_1 is generated by XOR-ing the stored ID of data owner O^*_{id} and security parameter a and it is concatenated with random variable p. Thus, the generated message is given below.

$$M_1 = A\left(O^*_{id} \oplus a\right) \| p \tag{1}$$

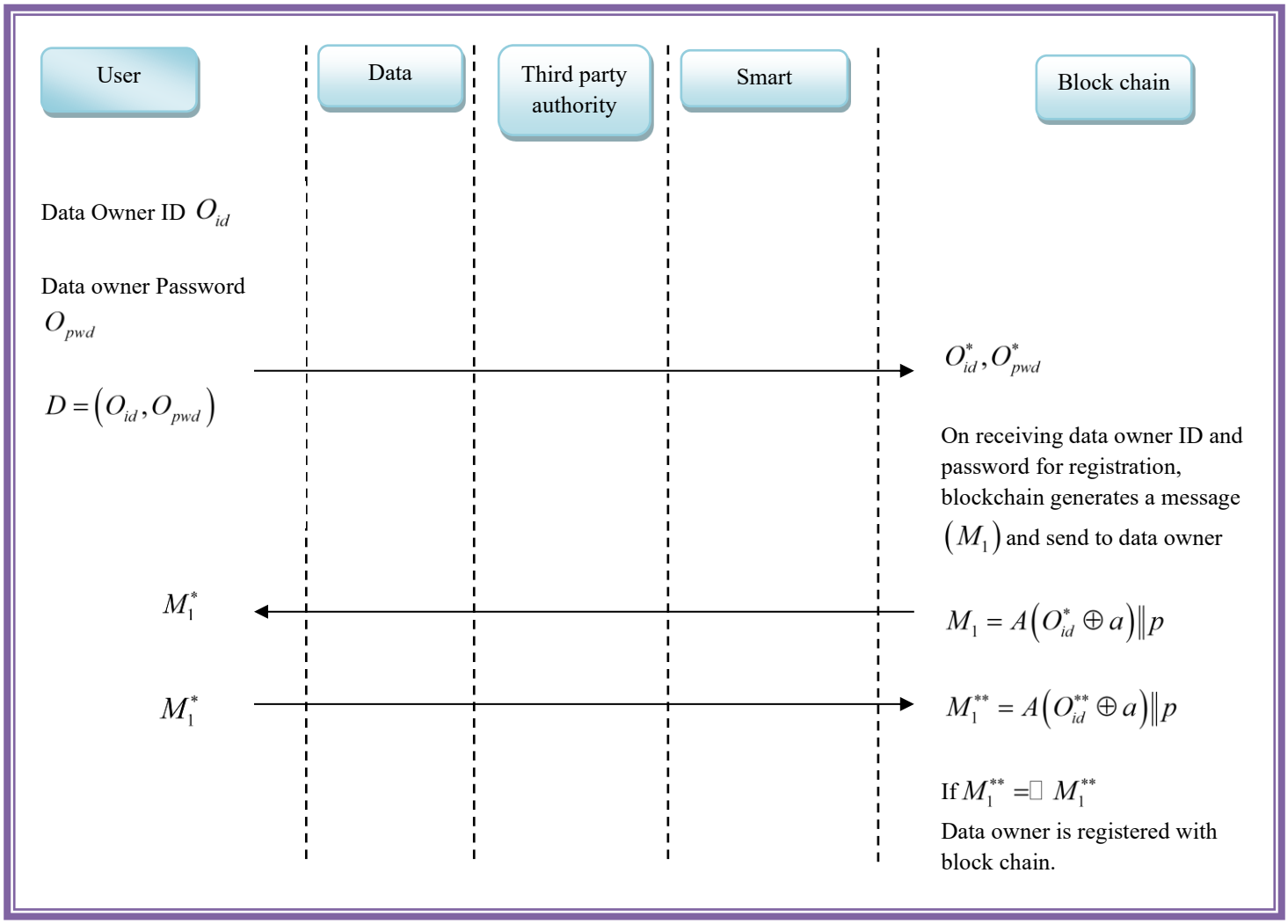
Where $A(.)$ denotes hash function, a implies security parameter, and p represents random number. In user, the generated message is stored as M^*_1 and it is restored in block chain as,

$$M^{**}_1 = A\left(O^{**}_{id} \oplus a\right) \| p \tag{2}$$

If $M^{**}_1 =$ M^{**}_1 Data owner is registered with block chain. The Registration between data owner and blockchain is displayed in Figure 4.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)



3.2.3 Registration between third party authority and block chain

Third-party authority is the legal power granted by an individual or organization to an external person or entity, allowing them to act on their behalf in specific matters. Registration between a third-party authority and a blockchain for privacy preservation in third-party services enables secure and verifiable identity or data validation without exposing sensitive information. In this registration, third party authority ID T_{id} and password T_{pwd} is stored in block chain as T_{id}^{*} and T_{pwd}^{*} .The Registration between a third party authority and blockchain is shown in Figure 5.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

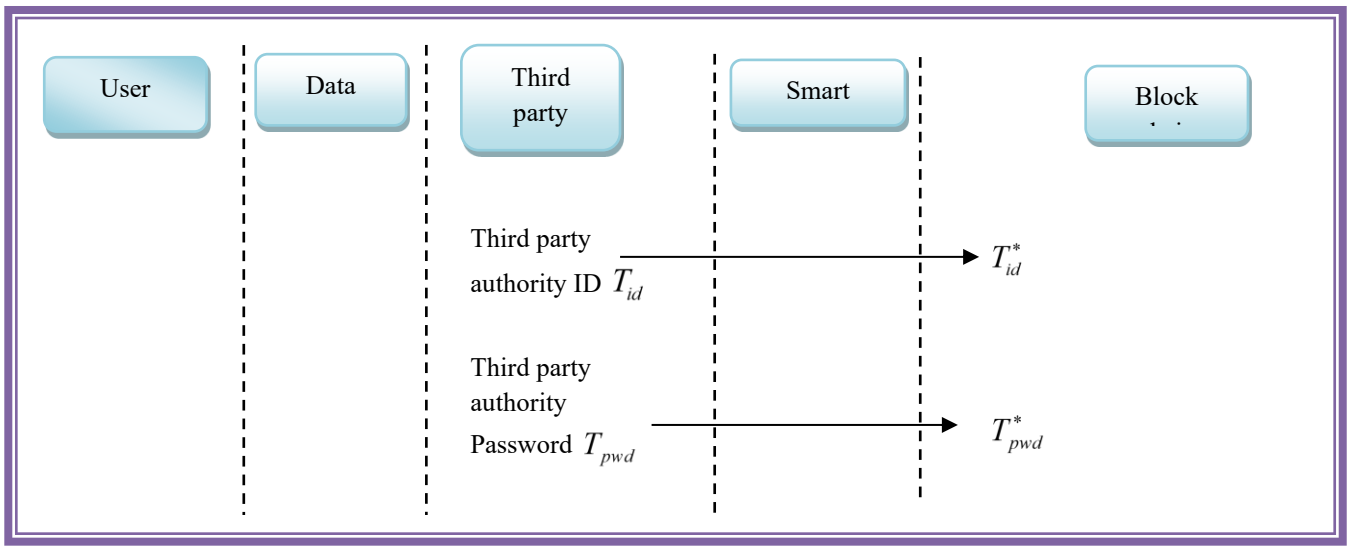


Figure 5. Registration between a third-party authority and blockchain

3.2.4 Registration between user and blockchain

Registration between a user and the blockchain involves creating a secure and verifiable identity record for the user within the blockchain network. here, the entities involved for registration is user, data owner and blockchain. In this process, the user ID U_{id} and password U_{pwd} is stored in data owner as U^* and it is restored in blockchain as U^{**}. Then, data owner generates a OTP for verification and it is sent to the user and it is stored in user as OTP^*. Here, the OTP is generated by concatenating stored user ID U_{id}^* and security parameter b and hash function and modulus of random variable q is applied.The generated OTP is given below.

$$OTP = A\left(U_{id}^* \| b\right) \bmod q \tag{3}$$

where OTP implies one time password, which generated based on modulus of random number $\bmod q$, and hash function by concatenating user ID and security parameter v. Moreover, OTP^* is restored in data owner as follows,

$$OTP^{**} = A\left(U_{id}^* \| b\right) \bmod q \tag{4}$$

If $\left(OTP = OTP^{**}\right)$ user is registered with data owner. Figure 6 illustrates Registration between a user and the blockchain.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

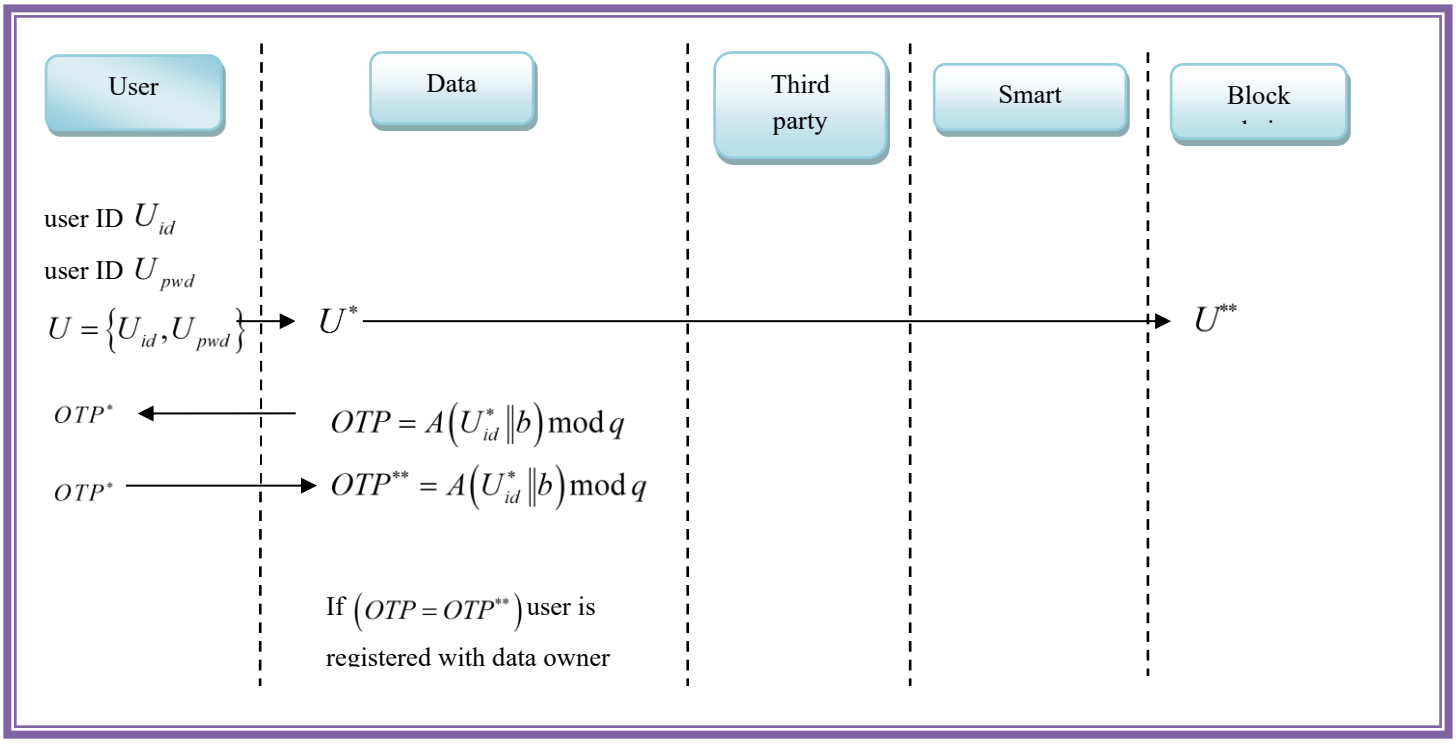


Figure 6. Registration between a user and the blockchain

3.3 Authorization between data owner and third-party authority

Authorization between a data owner and a third-party authority refers to the process by which the data owner grants specific access rights or permissions to the third party to use, view, or process their data. This ensures that the third-party authority can only interact with the data within the defined limits set by the owner, preserving control and privacy. At first, data owner sends an authorization request message to third party authority. The authorization request is developed by the concatenation of hash functions $A(O_{id} \oplus i)$ and $A(P_k \oplus r)$. Here, $A(O_{id} \oplus i)$ denotes hash function of XOR-ing user ID O_{id} and security variable i as well as $A(P_k \oplus r)$ represents hash function of private key P_k and random variable r and it is expressed as,

$$RE_1 = A(O_{id} \oplus i) \| A(P_k \oplus r) \tag{5}$$

Then, the RE_1 is stored in third party authority as RE_1^*,

$$RE_1^* = A(O_{id} \oplus i) \| A(P_k \oplus r) \tag{6}$$

$$\square\ RE_1^* = A(O_{id} \oplus i) \| A(P_k \oplus r) \tag{7}$$

If $(\square\ RE_1^* = RE_1^*)$ third party authority send a verification message to smart contract. The verification message designed by XOR-ing $A(O_{id}^* \| RE_1^* \| d)$ and $(c \| T_{id})$ and it is includes the

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
details about stored user ID O_{id}^{*}, authorization request RE_1^{*}, ID of third party authority T_{id}, and security parameters d, c. The generated verification message is given below.

$$VM_1 = A\left(O_{id}^{*} \| RE_1^{*} \| d\right) \oplus \left(c \| T_{id}\right) \tag{8}$$

In smart contract, the generated message is stored as VM_1^{*} and it is expressed as,

$$VM_1^{*} = A\left(O_{id}^{**} \| RE_1^{**} \| d\right) \oplus \left(c \| T_{id}^{*}\right) \tag{9}$$

$$\square\ VM_1 = A\left(O_{id}^{**} \| RE_1^{**} \| d\right) \oplus \left(c \| T_{id}^{*}\right) \tag{10}$$

If $\left(VM_1^{*} = \square\ VM_1^{*}\right)$ Smart contract generates verified message and send to third party authority, in which the verified message is stored as V_1^{*}. The generated verified message is given below.

$$V_1 = A\left(T_{id}^{*} \oplus b\right) \| A\left(r \oplus flagi\right) \bmod c \tag{11}$$

$$\square\ V_1^{*} = A\left(T_{id}^{*} \oplus b\right) \| A\left(r \oplus flagi\right) \bmod c \tag{12}$$

Here, $flagi$ indicates the data user is registered entity. If $\left(V_1^{*} = \square\ V_1^{*}\right)$ Third party authority authorizes data owner and send authorization message to data owner. The expression of authorization message is given below.

$$AM_1 = A\left(s \| flagi_q \| d\right) \bmod r \tag{13}$$

where $flagi_q$ represents that the data owner is authorized, AM_1 denotes authorization message. In addition, the third-party authority is said to be transparent since every action of third-party authority is saved in blockchain which can be monitored. The authorization among data owner and third-party authority is shown in Figure 7.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

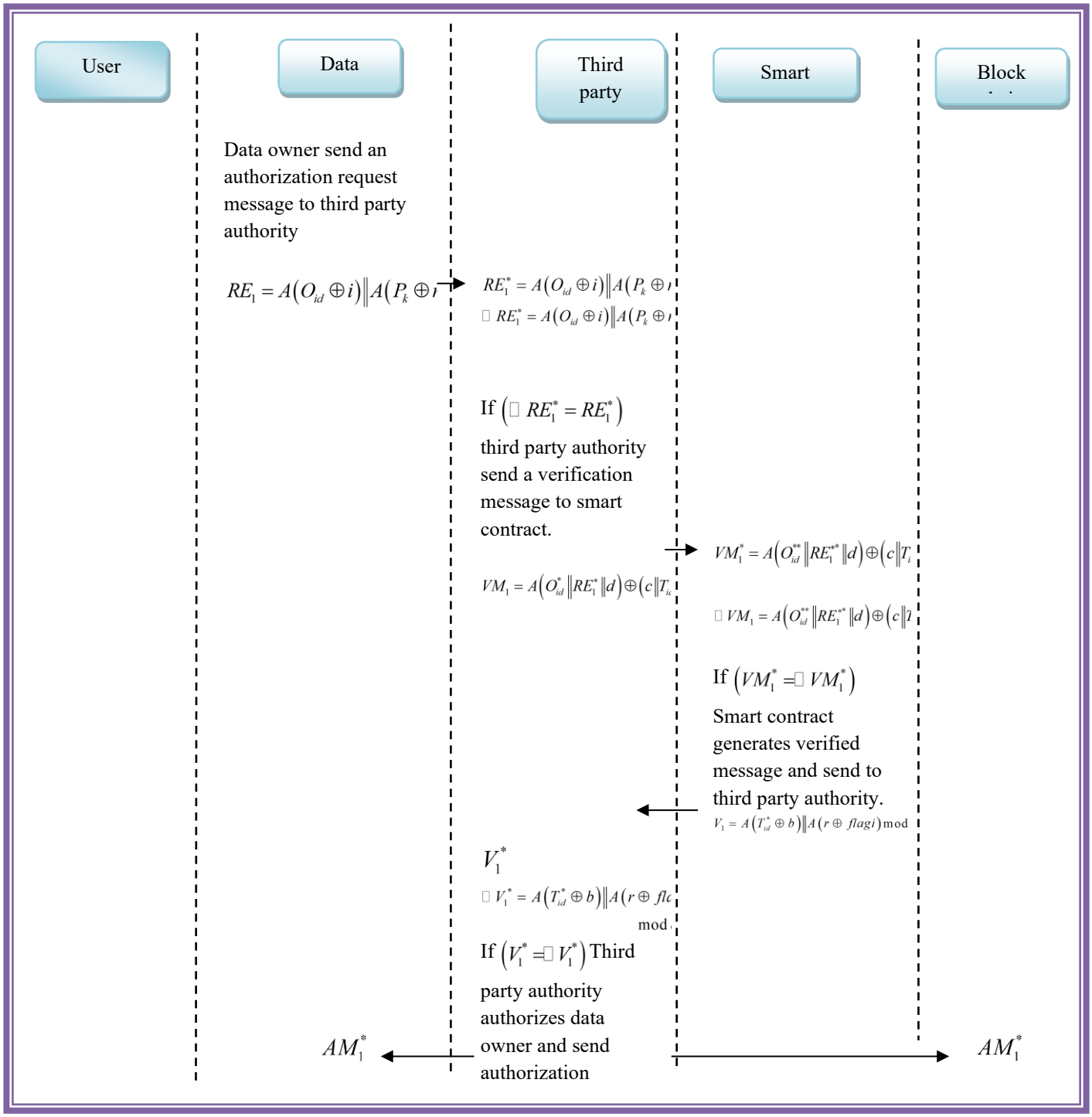


Figure 7. Process of authorization among data owner and third-party authority

3.4 Key request and Key generation

Key generation is a fundamental process in cryptography that involves creating a pair of keys used for secure communication. The primary goal of key generation is to produce keys that are unpredictable and unique, ensuring the security of the cryptographic system. This process utilizes a FFPBO_KeyGen for effective key generation, which is crucial because the strength

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
of encrypted data relies heavily on the secrecy and randomness of the keys produced. The key generation process is optimized to produce robust keys that safeguard sensitive information and facilitate secure authentication and data exchange across networks, especially in resource-constrained environments like IoT devices.

3.4.1 Key request from data owner to third party authority

At first, the Key request is sent from data owner to third party authority. Here, the key request is generated based on the hash function of concatenation of user ID of data owner O_{id} and private Key P_k and s represents random variable. The expression of key request K_1 is given below.

$$K_1 = A\left(O_{id} \| P_k\right) \oplus s \tag{14}$$

Then, the generated key request is stored in third party authority as K_1^*,

$$K_1^* = A\left(O_{id}^* \| P_k\right) \oplus s \tag{15}$$

$$\square\ s = K_1^* \oplus A\left(O_{id}^* \| P_k\right) \tag{16}$$

If $(\square\ s = s)$ third party authority generates a key $\left(Ke = Ke_1 \| Ke_2\right)$ and send to the data owner. Here, the Ke_1 is generated optimally using FFPBO_KeyGen, which is developed by the combination of PPBO and FFO. The mathematical development of FFPBO is explained in the following section. Thus, the attained key is stored in data owner and block chain as Ke^*. In addition, the Ke_2 is generated by XOR-ing security variable g with the concatenation of Quintic polynomial H private Key P_k and it is given below.

$$Ke_2 = g \oplus \left(H \| P_k^*\right) \tag{17}$$

where g represents extra polarization and it is expressed as,

$$g = g_1 + \frac{x - x_1}{x_2 - x_1}\left(g_2 - g_1\right) \tag{18}$$

By Quantic polynomial,

$$H = ex^5 + fx^4 + gx^3 + hx^2 + ix + j \tag{19}$$

Where x_1, x_2 denotes random value ranges from $[0,15]$, e, f, g indicates constants, h and i, j implies random value with range $[0,5]$ and $[0,9]$. Here, x is developed by XOR-ing private key P_k^* with the random parameter s and hash function is used. Moreover, modulus function of security variable b is applied and it is expressed as,

$$x = A\left(P_k^* \oplus s\right) \bmod b \tag{20}$$

Figure 8 shows Key request from data owner to third party authority.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

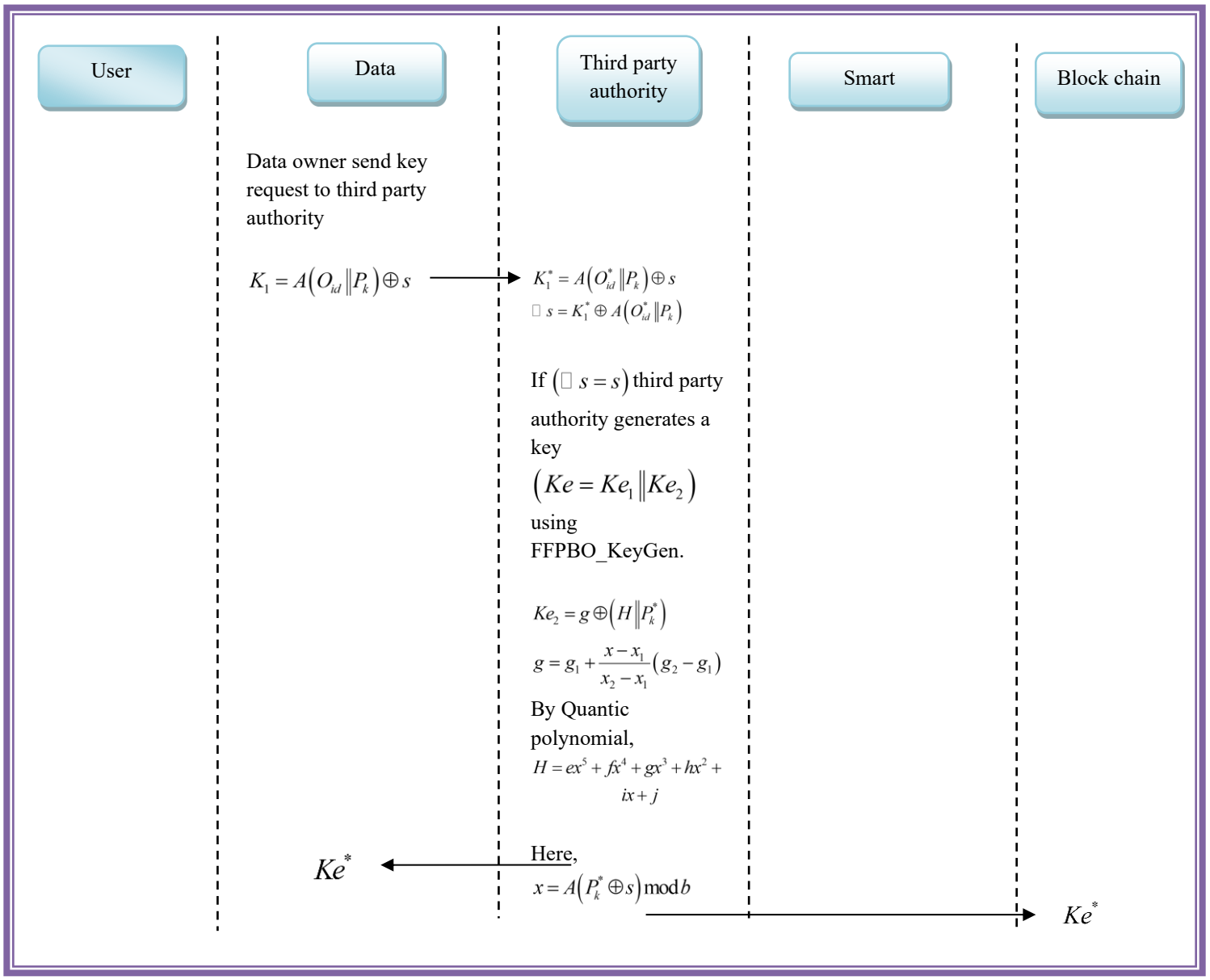


Figure 8. Process of Key request from data owner to third party authority

3.4.2 Key request from user to third party authority

The user sent a key request to third-party authority for accessing protected services or data. Here, the third-party authority, acting as a trusted entity, verifies the user's identity and authorization before issuing the key. This process ensures that only legitimate and authenticated users receive access, enhancing security and privacy. User send a key request K_2 to third party authority, which is given below.

$$K_2 = A(U_{id} \oplus d) \| A(q \oplus P_k) \tag{21}$$

Then, the request is stored in third party authority and block chain as K_2^*,

$$K_2^* = A(U_{id}^* \oplus d) \| A(q \oplus P_k) \tag{22}$$

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

If $(\square K_2^* = K_2^*)$ third party authority generates a key Ke_1 optimally and send to the user. User needs an access key, it requests to third party for access key. The key request generated by the user is given below.

$$K_3 = A(Ke_1 \oplus s) \| A(U_{id} \oplus b) \bmod a \tag{23}$$

Then, the key request K_3 from the user is stored in third party authority as follows,

$$K_3^* = A(Ke_1^* \oplus s) \| A(U_{id}^* \oplus b) \bmod a \tag{24}$$

If $(K_3 = K_3^*)$ third party verifies the user with data owner by sending a verification message, this is expressed as,

$$VM_2 = A(T_{id} \| a) \oplus A(U_{id}^*) \bmod r \tag{25}$$

Then, it is stored in third party authority as $\square VM_2^*$,

$$\square VM_2^* = A(T_{id}^* \| a) \oplus A(U_{id}^*) \bmod r \tag{26}$$

If $(\square VM_2^* = VM_2^*)$ data owner generates a verified message and send to third party authority. The generated verified message is given below.

$$V_2 = A(flag_p \| U_{id}^*) \bmod q \tag{27}$$

where $flag_p$ indicates that user is a registered entity and the generated message is stored in third party authority as follows,

$$V_2^* = A(flag_p \| U_{id}^*) \bmod q \tag{28}$$

If $(V_2^* = \square V_2^*)$ third party provides a key $(Ke = Ke_1 \| Ke_2)$ to the user and blockchain. Figure 9 shows steps involved in Key request from user to third party authority.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

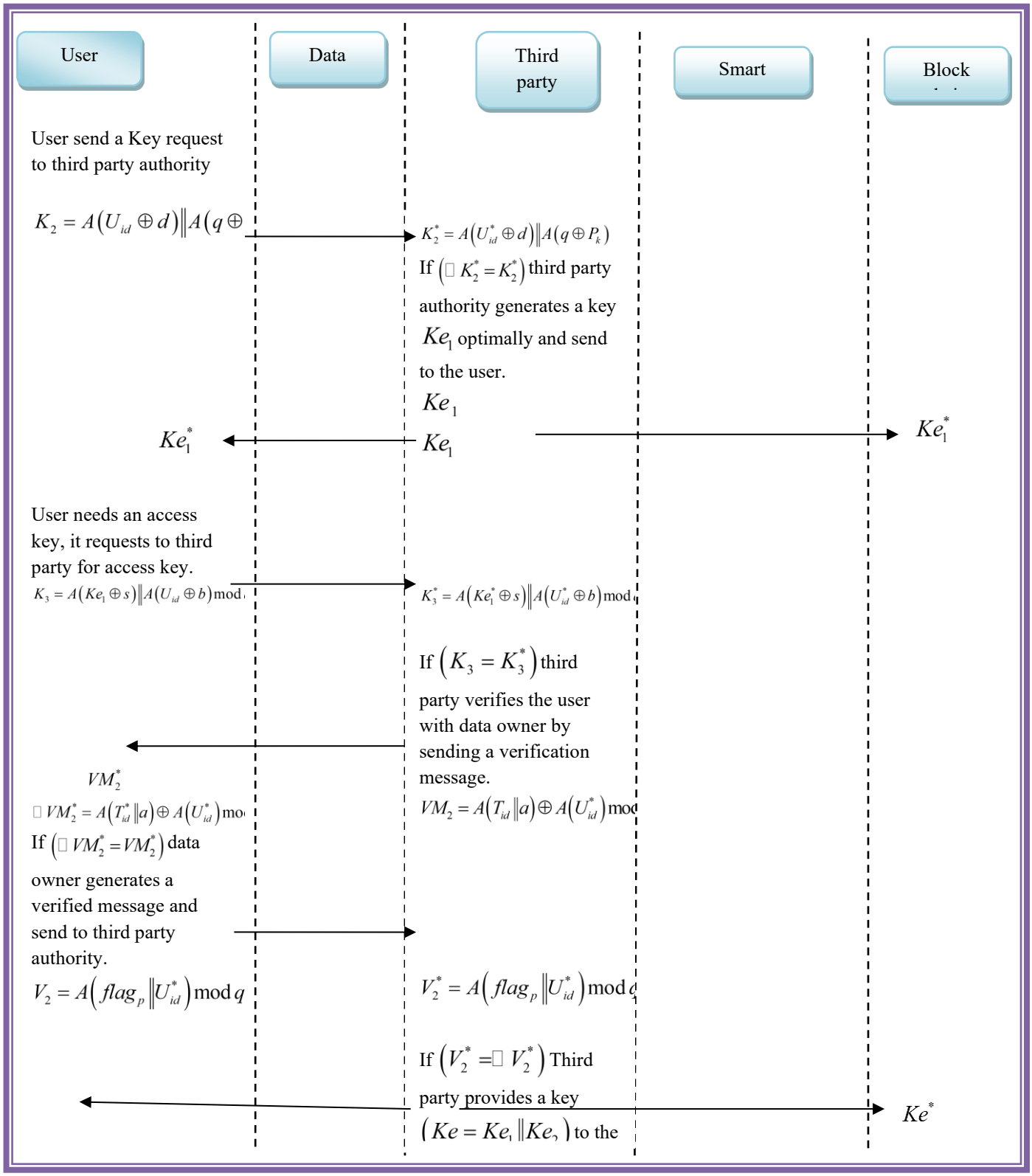


Figure 9. Steps involved in Key request from user to third party authority

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.5 Data protection

Data protection ensures the privacy and security of data during collection, storage, and transmission. This process is crucial to minimize data exposure and prevent misuse. Here, encrypted data En is stored in blockchain as En^*.

$$En = E(D, Ke) \quad (29)$$

Here, En contains the details about data owner termed as D and Key Ke is designed by the concatenation of Ke_1 and Ke_2, this is represented by $Ke = (Ke_1 \| Ke_2)$. Before encryption data owner acts as a monitor to check whether the key provided by third party authority is correct or not. If correct, decryption is done as follows.

$$De = D(En^*, Ke^*) \quad (30)$$

where $D(.)$ implies decryption function. The data protection process is shown in Figure 10.

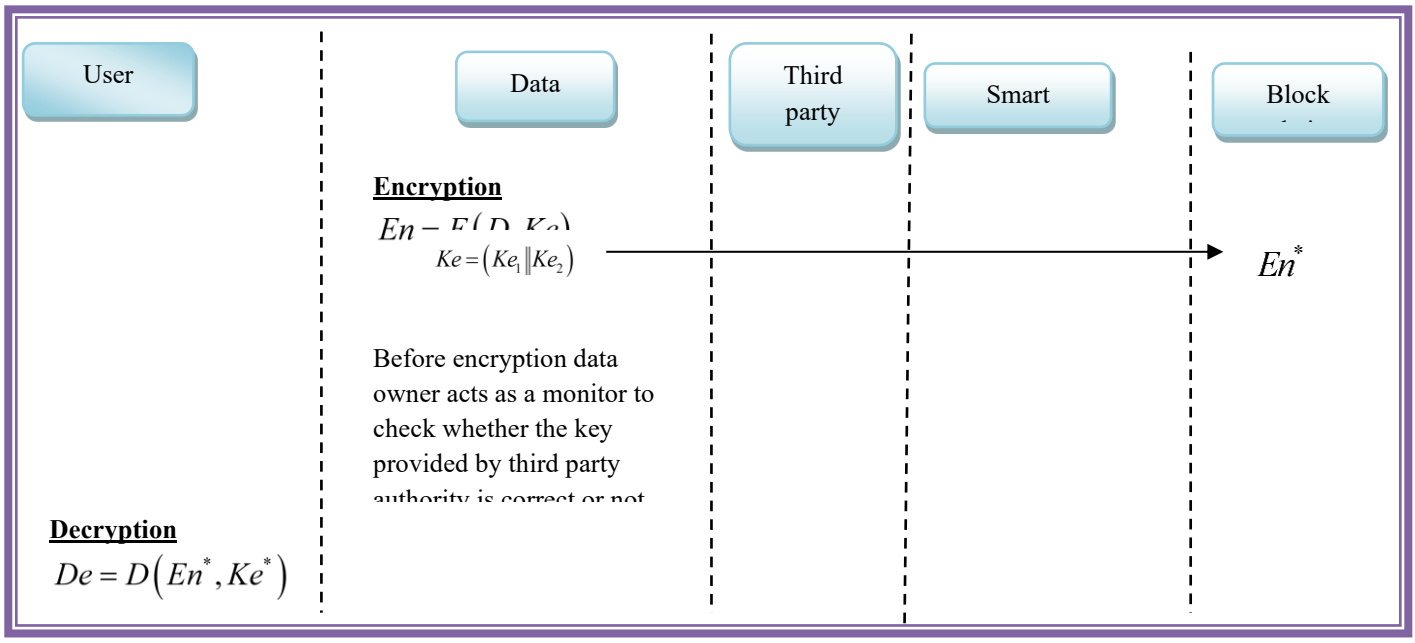


Figure 10. Data protection process

3.6 Access control

Access control is a security technique ensures that only authorized individuals or systems can access specific data.At first, user send an access control request to blockchain through data owner. The access control request is termed as AC and it is generated by XOR-ing user ID U_{id} and private Key P_k and hash function is used, and modulus function of random variable $\bmod a$ is applied. Here, the AC is represented as,

$$AC = A(U_{id} \oplus P_k) \bmod a \quad (31)$$

The generated access control request is stored in data owner as AC^*,

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$AC^* = A\left(U_{id}^* \oplus P_k\right) \bmod a \tag{32}$$

If $\left(AC^* = AC^*\right)$ data owner generates a message using the access control message from the user and send to the smart contract. The message is generated by XOR-ing stored U_{id}^* and access conrol AC and hash function is employed. Likewise, the Id of data owner O_{id} and security parameter s is XOR-ed, then hash function is applied and $\bmod p$ is used. Finally, attained two hashing functions are concatenated. Thus, the generated message is represented as,

$$M_2 = A\left(U_{id}^* \oplus AC\right) \| A\left(O_{id} \oplus s\right) \bmod p \tag{33}$$

Here, the generated message is stored in smart contract as M_2^*,

$$M_2^* = A\left(U_{id}^{**} \oplus AC\right) \| A\left(O_{id}^* \oplus s\right) \bmod p \tag{34}$$

If $\left(M_2^* = M_2^*\right)$ smart contract verifies the user identity and accept the access control from the user and send the encrypted data. This process is shown in Figure 11.

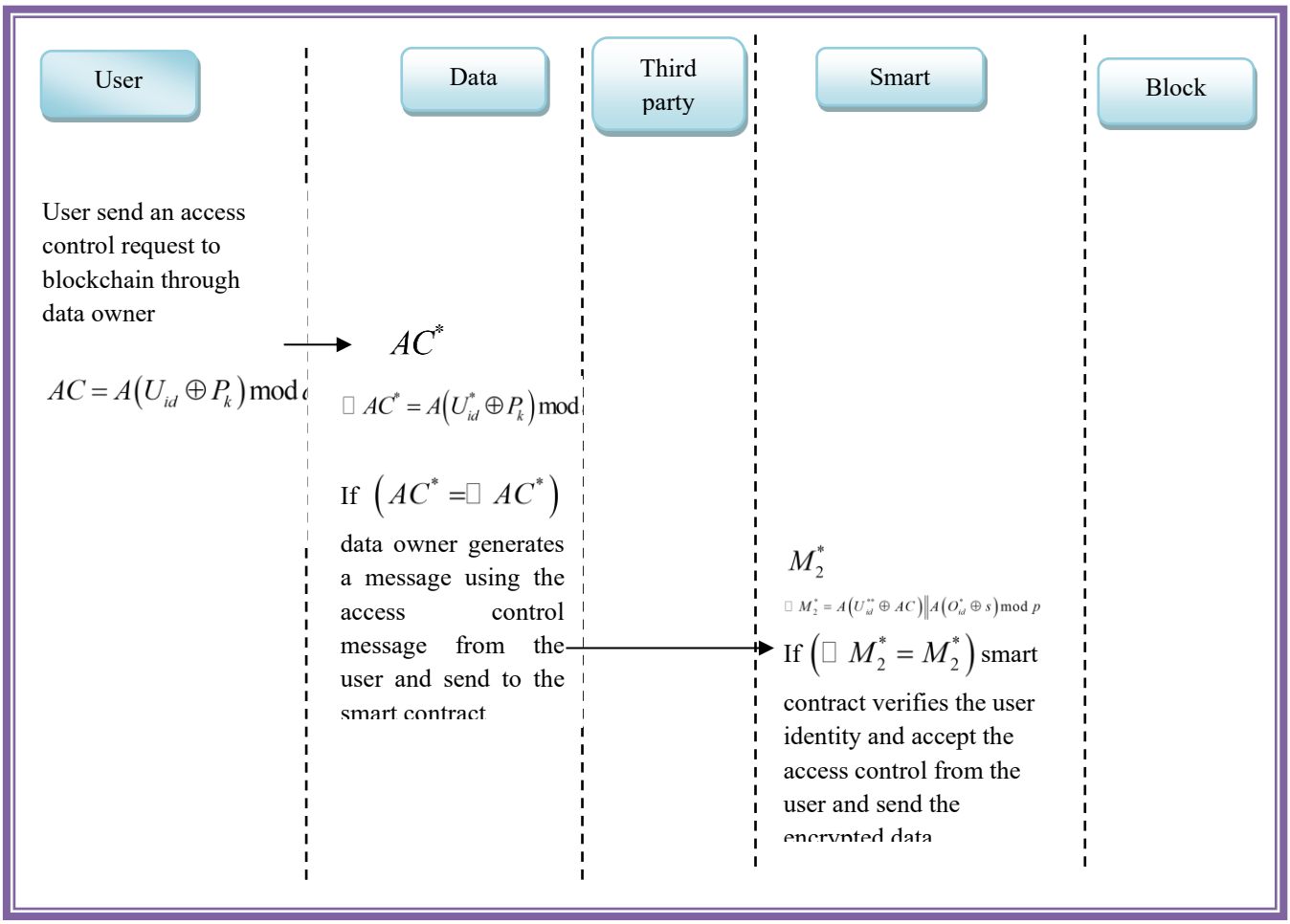


Figure 11. Access control

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.7 Key generation using hybrid FFPBO

Key generation using hybrid optimization combines the strengths of multiple optimization techniques to enhance the security, efficiency, and robustness of cryptographic keys. This approach typically integrates classical methods, such as genetic algorithms, particle swarm optimization, or simulated annealing with other computational strategies to explore the key space more effectively. By leveraging the global search capabilities of one method and the fine-tuning ability of another, hybrid optimization ensures the generation of highly secure and unpredictable keys that are resistant to various cryptanalytic attacks. This process involves iteratively refining candidate keys based on specific fitness criteria, such as entropy, randomness, and resistance to brute-force attempts, until an optimal key is identified. Overall, hybrid optimization techniques provide a powerful framework for generating cryptographic keys that are both secure and adaptable to evolving security requirements.

3.7.1 Solution encoding

Solution encoding is used to achieve potential solution for key generation in a specific region Ψ such that $\Psi \in [1 \times q]$, where q implies Key size.

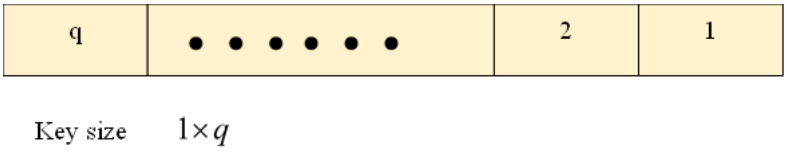


Figure 8. Solution encoding of the hybrid FFPBO

3.7.2 Fitness function

A fitness function is used to evaluate the efficiency of a candidate solution for optimal key generation. To determine the optimal solution, fitness is estimated from normalized variance and conditional privacy metrics. The solution with the maximum fitness score F is then selected, as described as follows,

$$F = \frac{N + C}{2} \tag{35}$$

In this context, N epitomizes the normalized variance, and the conditional privacy is signified as C.

Normalized Variance:

Normalized variance [27] assesses how much the perturbed data differs from the original data in terms of dispersion and is based on the concept of statistical variance. The normalized variance N is formulated as,

$$N = \frac{\alpha^2\left(\chi^* - \mathrm{A}\right)}{\alpha^2\left(\chi^*\right)} \tag{36}$$

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

wherein, the perturbed data is specified as A, χ^* delineates the actual data, and α^2 represents the statistical variance.

Conditional privacy:

Conditional privacy quantifies privacy using conditional entropy, which reflects the uncertainty of a random variable χ when another related variable A is observed. Conditional privacy C is mathematically articulated as,

$$C = 2^{O(\chi|\mathrm{A})} \tag{37}$$

Here, o characterizes the entropy.

3.7.3 Algorithm of FFPBO

The mathematical modeling of FFPBO is given below.

Step 1: Initialization

The FFO algorithm is a new biological optimization technique inspired by the natural process of flower fertilization in plants. The amount of plants in the specified location is termed as,

$$F = \left\{F_1, F_2, ..., F_c, ...F_m\right\} \tag{38}$$

Where F denotes input population, F_m indicates total member in the population, and F_c is considered as candidate solution.

Step 2: Fitness function

The fitness function used to assess the efficiency of accomplished result for effective key generation.

Step 3: Searching strategies

At first, all pollen grains are assigned random positions and velocities, allowing them to disperse throughout the searching location. The global search process employs Levy flights to simulate the long-range movement of pollen grains, facilitating exploration of distant regions within the landscape. This process is expressed as,

$$Y_l^{r+1} = V\left(Y_l^r - Z_l^r\right) \tag{39}$$

where Y_l^r implies solution and Z_l^r represents velocity.

Let us consider,

$$Y_l^{r+1} = Y_l\left(r+1\right) \tag{40}$$

$$Y_l^r = Y_l\left(r\right) \tag{41}$$

$$Z_l^r = z(r) \tag{42}$$

Thus, the expression of Y_l^{r+1} is rewritten as,

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$Y_l(r+1) = V\left[Y_l(r) - Z_l(r)\right] \tag{43}$$

The standard expression of PPBO is given below,

$$E_l^1 = E_l + rand\left(F_l - J_l \cdot E_l\right) \tag{44}$$

Let us assume,

$$E_l^1 = Y_l(r+1) \tag{45}$$

$$E_l = Y_l(r) \tag{46}$$

$$Y_l(r+1) = Y_l(r) + rand\left(F_l - J_l \cdot Y_l(r)\right) \tag{47}$$

$$Y_l(r) = \frac{Y_l(r+1) + rand\, F_l}{\left(1 - rand \cdot J_l\right)} \tag{48}$$

By applying Eq. (48) in Eq. (43),

$$Y_l(r+1) = V\left[\frac{Y_l(r+1) - rand\, F_l}{\left(1 - rand \cdot J_l\right)}\right] - V \cdot Z_l(r) \tag{49}$$

$$Y_l(r+1) - \frac{VY_l(r+1)}{\left(1 - rand \cdot J_l\right)} = V\frac{-rand \cdot F_l}{\left(1 - rand \cdot J_l\right)} - V \cdot Z_l(r) \tag{50}$$

$$-\frac{Y_l(r+1)\left[1 - rand \cdot J_l - V\right]}{\left(1 - rand \cdot J_l\right)} = \frac{-Vrand \cdot F_l - V \cdot Z_l(r)\left(1 - rand \cdot J_l\right)}{\left(1 - rand \cdot J_l\right)} \tag{51}$$

The updated expression of FFPBO is given below.

$$Y_l(r+1) = \frac{-V\left(Y_l + rand\, F_l + Z_l(r)\right)\left(1 - rand \cdot J_l\right)}{\left(1 - rand \cdot J_l - V\right)} \tag{52}$$

where F_l denotes random numbers uniformly selected, J_l indicates position of reviewer, and V presents levy function and it is expressed as,

$$V = \frac{\delta}{\left|\Upsilon\right|^{\frac{1}{\rho}}} \tag{53}$$

where δ, Υ implies random variables.

Step 4: Local search through velocity reduction

As time increases, the velocities of the pollen grains reduced, which directs the search towards a more localized area. This action is represented by,

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$Z_l^{r+1} = Z_l^r e^{\left(\frac{-1}{\tau^{r+1}}\right)} \tag{54}$$

where τ^{r+1} implies reduction coefficient, Z_l^{r+1} and Z_l^r indicates velocity of solution.

Step 5: Revaluation of fitness function

The fitness of obtained solution is revaluated to achieve efficient key generation for privacy preservation in third party services.

Step 6: Termination

The above procedure is iterated for allowing effective key generation. Algorithm 1 shows pseudocode of FFPBO_KeyGen.

Algorithm 1. Pseudocode of FFPBO

SL. No	Pseudocode of FFPBO
1	**Input:** $Y_l(r)$, velocity vector, upper and lower bounds
2	**Output:** $Y_l(r+1)$
3	Start
4	Initialize new pollen population
5	Compute Fitness for effective key generation
6	**Searching strategies:**
7	Use Levy flights to model long distance movement using Eq. (36)
8	Update the location of pollen using Eq. (49)
9	**Local search through velocity reduction**
10	Perform local search based on velocity reduction using Eq. (52)
11	Update best solution
12	Re-evaluate the fitness
13	Terminate

4. RESULT AND DISCUSSION

This section presents a comparative analysis of FFPBO_KeyGen. Here, the effectiveness of devised scheme is assessed by varying block size based on key evaluation metrics.

4.1 Experimental setup

The devised privacy preservation of third-party services using FFPBO_KeyGen is implemented in Python tool.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.2 Experimental results

Figure 12 shows implementation steps of FFPBO_KeyGen. Here, the privacy protection model includes multiple phases, such as Initialization, Registration, Authorization, Key request and Key generation, data protection and access control.

```
Initialization Phase
   >>>> Parameters are initialized
Registration Phase
   >>>> registration between smart contract and block chain is done
   >>>> Data owner is registered with blockchain
   >>>> User is registered with Data owner
Authorization Phase
   >>>> Authorization between data owner and Third party authority is done
Key Request and Key generation
   >>>> key request from Data owner to Third party authority
   >>>> key request from User to Third party authority
   >>>> Key generated using Proposed FFPBO_KeyGen
Data Protection
   >>>> Data is encrypted
Access control
   >>>> User is verified and smart contract sends the data to the User
```

Figure 12. Implementation steps of FFPBO_KeyGen

4.3 Evaluation metrics

This section discusses the key evaluation parameters employed to assess the effectiveness of designed privacy preservation model, named FFPBO_KeyGen.

4.3.1 Authorization time

Authentication time[26] refers to the total duration required to verify the identity of users or entities before granting access to sensitive data or services. The authentication time is termed as κ_1.

4.3.2 Privacy rate

Privacy rate [27] quantifies the proportion of data transactions that successfully maintain confidentiality, anonymity, and compliance with privacy policies. The privacy rate is denoted as κ_2.

4.3.3 Memory

Memory[28]indicates the amount of computational storage resources required to securely handle, process, and protect sensitive data during authentication, encryption, key management, and other privacy-preserving operations. Here, κ_3 implies memory.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.3.4 Encryption time

Encryption time[29] refers to the duration required to convert sensitive data into a secure, encoded format using cryptographic algorithms before it is transmitted, stored, or processed by external entities. The encryption time is denoted by κ_4.

4.3.5 Computational time

Computational time[30] refers to the total time required by a system to perform all necessary cryptographic and privacy-preserving operations. The computation time is specified as κ_5.

4.4 Comparative techniques

The effectiveness of FFPBO_KeyGen is evaluated by varying the key size, and its performance is compared with existing approaches such as TAB framework [16], FL+LSTM autoencoder [17], LSTM-GRU [18], and ZeroTrustBlock [19].

4.5 Comparative evaluation

This section elucidates the comparative evaluation of FFPBO_KeyGen by varying block size.The efficiency of FFPBO_KeyGen is assessed using key performance metrics and its efficiency is compared with conventional modules.

4.5.1 Evaluation ofFFPBO_KeyGen for block size 2

Figure 13depicts evaluation of FFPBO_KeyGen with block size of 2. Figure 13 a) presents the assessment of FFPBO_KeyGen using authorization time. When Key size=256 bits, the presented scheme gained authorization time as0.0089sec and its existing schemes gained authorization time as 0.0103sec, 0.0099sec, 0.0096sec, and 0.0093sec.Figure 13b) illustrates assessment ofFFPBO_KeyGen with respect to privacy rate. The FFPBO_KeyGen achieved a privacy rate as 0.934 and the conventional approaches, like TAB framework, FL+LSTM autoencoder, LSTM-GRU, and ZeroTrustBlock gained privacy rate as 0.788, 0.814, 0.862, and 0.910 with key size=256 bits. Figure 13 c) displays evaluation ofFFPBO_KeyGen based on memory. The FFPBO_KeyGenand its existing models gained memory as 159.33MB, 269.75MB, 268.68MB, 226.03MB, and 171.80MB by utilizing key size as 256 bits. Figure 13 d) shows assessment ofFFPBO_KeyGen using encryption time. Whileconsidering key size= 256 bits, theproposed model utilized encryption time as 0.0071sec and its classical modules, like TAB framework, FL+LSTM autoencoder, LSTM-GRU, and ZeroTrustBlock achieved encryption time as 0.0096sec, 0.0093sec, 0.0084sec, and 0.0077sec. The evaluation of FFPBO_KeyGen using computation time is depicted in Figure 13 e). The FFPBO_KeyGen and its prior schemes accomplished computation time as 8.65sec, 10.56sec, 10.38sec, 10.21sec, and 9.93sec with key size as 256 bits.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

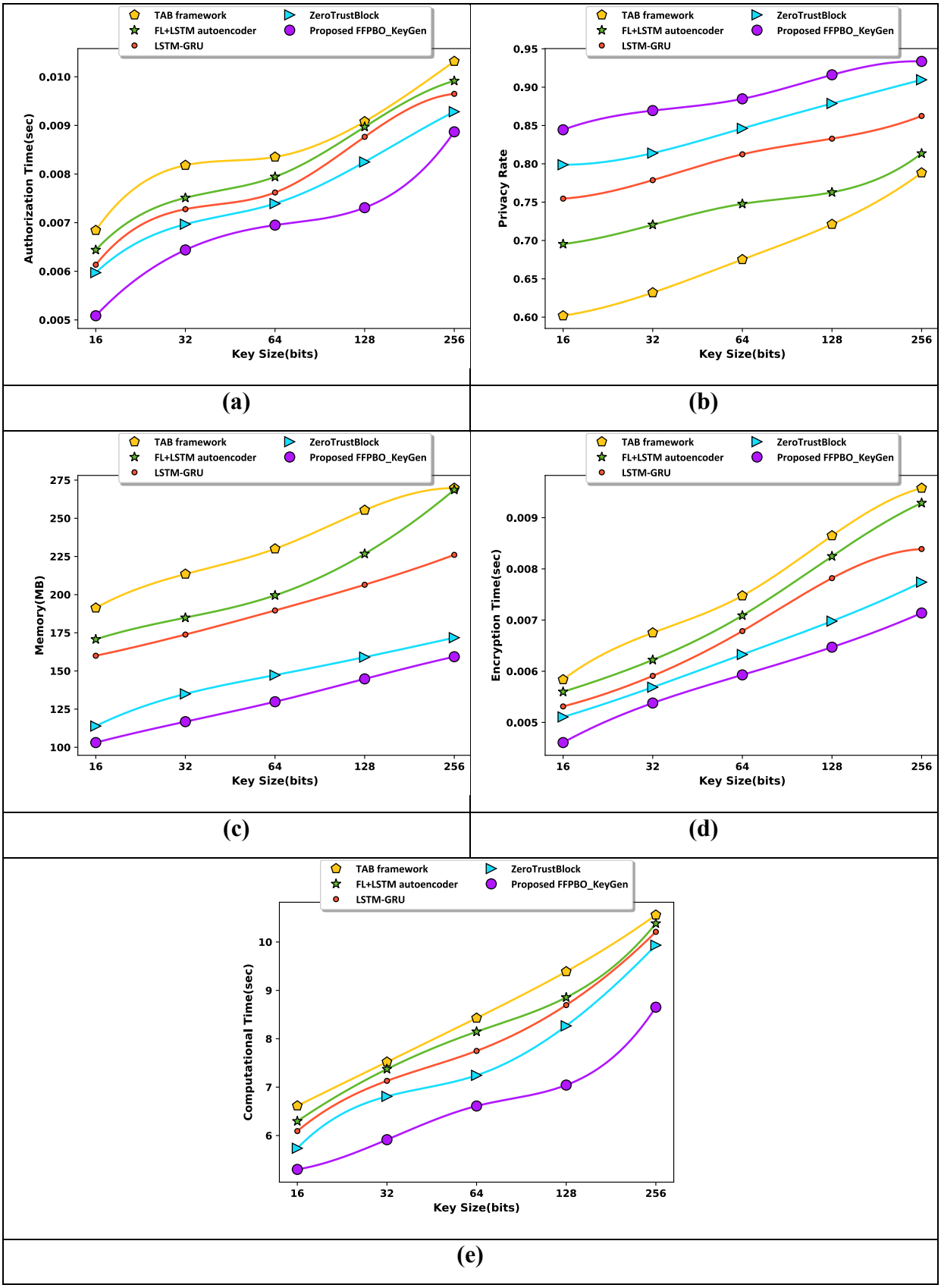


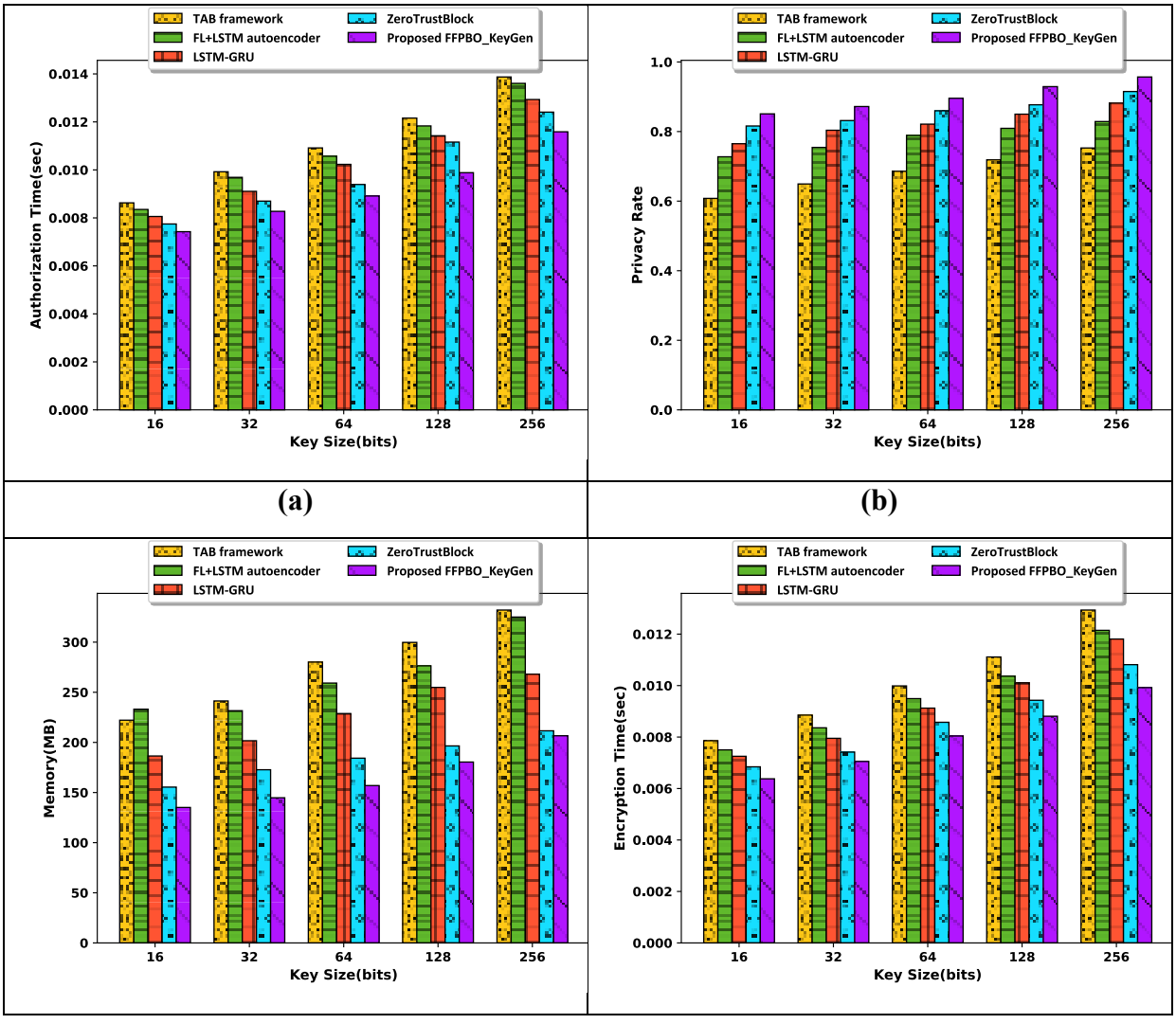
Figure 13. Evaluation of FFPBO_KeyGen with block size= 2, a) Authentication time, b) Privacy rate, c) Memory, d) Encryption time, e) computational time

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.5.2 Evaluation of FFPBO_KeyGen for block size 4

Figure 14 depicts evaluation of FFPBO_KeyGen with block size of 4. Figure 14 a) presents the assessment of FFPBO_KeyGen using authorization time. With key size of 256 bits, the FFPBO_KeyGen attained authorization time as 0.0116 sec and its classical models attained authorization time as 0.0139sec, 0.0136sec, 0.0129sec, and 0.0124sec. Figure 14 b) illustrates assessment of FFPBO_KeyGen based on privacy rate. While using Key size as 256 bits, the FFPBO_KeyGen and its prior modules achieved a privacy rate of 0.957, 0.753, 0.829, 0.882, and 0.915. The evaluation of FFPBO_KeyGen based on memory is presented in Figure 14 c). When Key size= 256 bits, the FFPBO_KeyGen achieved memory as 206.62MB and the traditional methods gained memory as 332.02MB, 325.04MB, 267.98MB, and 211.57MB. Figure 14 d) shows evaluation of FFPBO_KeyGen using encryption time. The FFPBO_KeyGen and its conventional schemes gainedencryption time as 0.0099sec, 0.0129sec, 0.0121sec, 0.0118sec, and 0.0108sec. Figure 14 e) illustratesevaluation of FFPBO_KeyGen using computation time. By utilizing key size as 256 bits, the proposed model consumes computational time as 11.85 sec and TAB framework, FL+LSTM autoencoder, LSTM-GRU, and ZeroTrustBlock obtained computational time as 14.50sec, 13.74sec, 13.26sec, and 12.52sec respectively.



Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

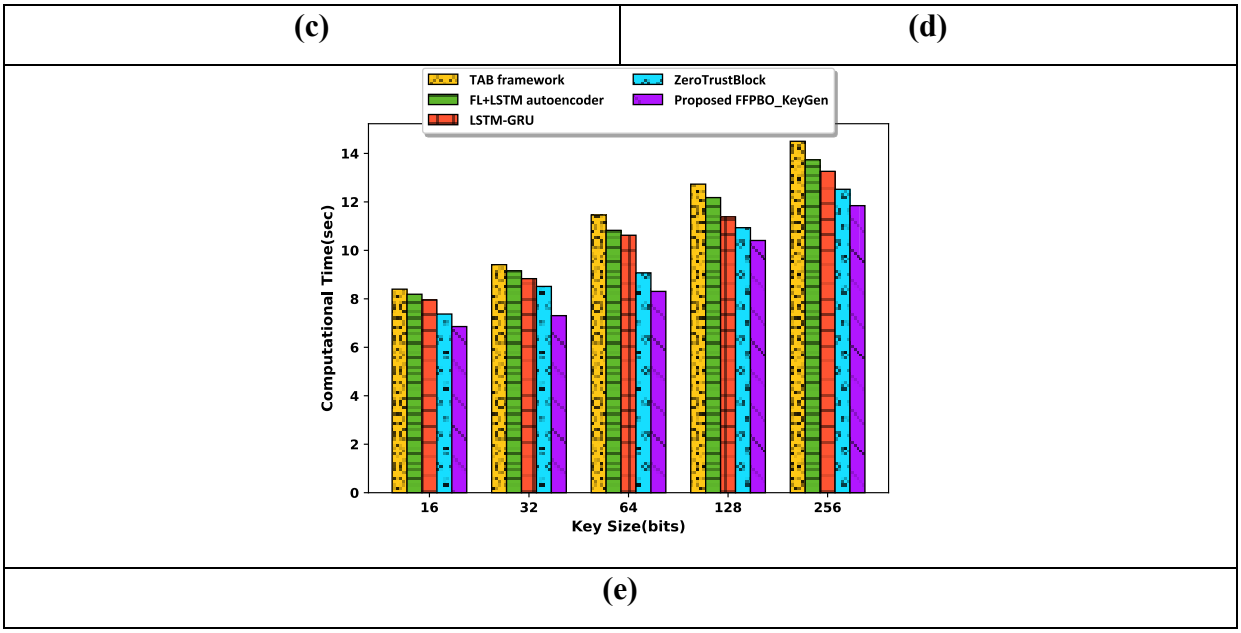


Figure 14. Evaluation of FFPBO_KeyGen with block size= 4, a) Authentication time, b) Privacy rate, c) Memory, d) Encryption time, e) computational time

4.6 Comparative discussion

The comparative discussion ofFFPBO_KeyGen is detailed in Table 2. The presented scheme gained authorization time as 0.0089sec and its existing schemes gained authorization time as 0.0103sec, 0.0099sec, 0.0096sec, and 0.0093sec. A reduced authorization time supports real-time decision-making processes and secure communication channels, making it harder for attackers to exploit vulnerabilities during authentication exchanges. The FFPBO_KeyGen achieved a privacy rate as 0.934 and the conventional approaches, like TAB framework, FL+LSTM autoencoder, LSTM-GRU, and ZeroTrustBlock gained privacy rate as 0.788, 0.814, 0.862, and 0.910. A high privacy rate indicates that user data remains well-protected throughout processing, storage, and transmission, even when accessed or managed by external entities. The FFPBO_KeyGen and its existing models gained memory as 159.33MB, 269.75MB, 268.68MB, 226.03MB, and 171.80MB. Attaining low memory usage is significant for privacy preservation in third-party services because it reduces the footprint of sensitive data within a system, thereby minimizing potential exposure and attack surfaces. The proposed model utilized encryption time as 0.0071sec and its classical modules, like TAB framework, FL+LSTM autoencoder, LSTM-GRU, and ZeroTrustBlock achieved encryption time as 0.0096sec, 0.0093sec, 0.0084sec, and 0.0077sec. Low encryption time strengthens the overall privacy framework by ensuring that data is protected instantly, reducing the risk of interception or unauthorized access during processing and transmission in third-party service scenarios. The FFPBO_KeyGen and its prior schemes accomplished computation time as 8.65sec, 10.56sec, 10.38sec, 10.21sec, and 9.93sec with key size as 256 bits. Lower computational time enhance overall security. In addition, the FFPBO_KeyGen attained authorization time, privacy rate, memory, encryption time, and computational time as 0.0089sec, 0.934, 159.33 MB, 0.0071sec, and 8.65sec respectively.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s 2025,
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 2. Comparative discussion

Metrics/Methods		TAB framework	FL+LSTM autoencoder	LSTM-GRU	ZeroTrustBlock	Proposed FFPBO_KeyGen
Block size=2	*Authorization time (sec)*	0.0103	0.0099	0.0096	0.0093	**0.0089**
	Privacy rate	0.788	0.814	0.862	0.910	**0.934**
	Memory (MB)	269.75	268.68	226.03	171.80	**159.33**
	Encryption time(sec)	0.0096	0.0093	0.0084	0.0077	**0.0071**
	Computational time(sec)	10.56	10.38	10.21	9.93	**8.65**
Block size=4	*Authorization time (sec)*	0.0139	0.0136	0.0129	0.0124	0.0116
	Privacy rate	0.753	0.829	0.882	0.915	0.957
	Memory (MB)	332.02	325.04	267.98	211.57	206.62
	Encryption time(sec)	0.0129	0.0121	0.0118	0.0108	0.0099
	Computational time(sec)	14.50	13.74	13.26	12.52	11.85

5. CONCLUSION

In today's digital ecosystem, vast amounts of personal and sensitive data are routinely shared with third-party services, ensuring privacy preservation has become a critical challenge. Third-party platforms often require access to user data for processing and analytics, yet this opens the door to potential data breaches, unauthorized access, and misuse of information. Traditional encryption methodsoften insufficient in dynamic and resource-constrained environments, particularly when fine-grained control over data access and usage is required. To overcome these challenges, this work proposes a novel hybrid key generation approach based on FFPBO_KeyGen within a blockchain framework. The system architecture involves key components, such as smart contracts, blockchain, a third-party authority, data owners, and users. The third-party authority plays a crucial role in safeguarding privacy. The key generation process encompasses various stages, including initialization, registration, authorization, key request, key generation, data protection, and access control, all designed to ensure robust privacy standards. Secure keys are generated by the third-party authority utilizing the FFPBO, which is devised by integrating PPBO and FFO. Moreover, the FFPBO_KeyGen obtained authorization time, privacy rate, memory, encryption time, and

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
computational time as 0.0089sec, 0.934, 159.33MB, 0.0071sec, and 8.65sec respectively. Additionally, integrating homomorphic encryption or secure multi-party computation techniques will enable privacy-preserving operations on encrypted data and it will allow the third-party services to perform computations without accessing raw user data.

REFERENCES

[1] P. Sharma, S. Namasudra, N. Chilamkurti, B. Ung-Gy U Kim and R. Gonzalez Crespo, "Blockchain-based privacy preservation for IoT-enabled healthcare system," ACM Transactions on Sensor Networks, vol. 19, no. 3, pp. 1-17, 2023.

[2] J. Liu, X. Li, L. Ye, H. Zhang, X. Du and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," IEEE, pp. 1-6, 2016.

[3] J. Wang, M. Li, Y. He, H. Li, K. Xiao and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," IEEE Access, vol. 6, pp. 17545-17556, 2018.

[4] S. Zhang, T. Yao, V. Koe Arthur Sandor, T.-H. Weng, W. Liang and J. Su, "A novel blockchain-based privacy-preserving framework for online social networks," Connection Science, vol. 33, no. 3, pp. 555-575, 2021.

[5] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou and S. Shimizu, "Privacy preservation in permissionless blockchain: A survey," Digital Communications and Networks, vol. 7, no. 3, pp. 295-307, 2021.

[6] G. Ayoade, V. Karande, L. Khan and K. Hamlen, "Decentralized IoT data management using blockchain and trusted execution environment," IEEE, pp. 15-22, 2018.

[7] R. Xu and J. Joshi, "Trustworthy and transparent third-party authority," ACM Transactions on Internet Technology (TOIT), vol. 20, no. 4, pp. 1-23, 2020.

[8] J. Bernal Bernabe, J. Luis Canovas, J. L. Hernandez-Ramos, R. Torres Moreno and A. Skarmeta, "Privacy-preserving solutions for blockchain: Review and challenges," IEEE Access, vol. 7, pp. 164908-164940, 2019.

[9] A. Satybaldy and M. Nowostawski, "Review of techniques for privacy-preserving blockchain systems," In Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure, pp. 1-9, 2020.

[10] W. Mingyue, G. Yu, Z. Chen, W. Cong, H. Hejiao and J. Xiaohua, "MedShare: A privacy-preserving medical data sharing system by using blockchain," IEEE , vol. 16, no. 1, pp. 438-451, 2021.

[11] D. Cezane Gomes Valadares, A. Perkusich, A. Falcao Martins, M. B. Alshawki and C.

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
Seline, "Privacy-preserving blockchain technologies," Sensors, vol. 23, no. 16, pp. 7172, 2023.

[12] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang and W. Luo, "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive," IEEE , vol. 18, no. 5, pp. 2438-2455, 2019.

[13] M. Xu, Z. Zou, Y. Cheng, Q. Hu, D. Yu and X. Cheng, "Spdl: blockchain-secured and privacy-preserving decentralized learning," arXiv preprint arXiv:2201.01989, 2022.

[14] A. Julian, G. Immaculate Mary, S. Selvi and M. Rele, "Blockchain based solutions for privacy-preserving authentication and authorization in networks," 2024.

[15] N. Afrin and A. Pathak, "Blockchain-Powered Security and Transparency in Supply Chain: Exploring Traceability and Authenticity through Smart Contracts," International Journal of Computer Applications, vol. 85, pp. 5-15, 2023.

[16] R. Xu, C. Li and J. Joshi, "Blockchain-based transparency framework for privacy preserving third-party services," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2302-2313, 2022.

[17] R. Vijay Anand, G. Magesh, I. Alagiri, M. Guru Brahmam, B. Balusamy, C. Pon Selvan, H. Mesfer Alshahrani, M. Getahun and B. Othman Soufiene, "Design of an improved model using federated learning and LSTM autoencoders for secure and transparent blockchain network transactions," Scientific Reports, vol. 15, no. 1, pp. 1-18, 2025.

[18] U. Islam, A. Alshammari, Z. Alzaid, S. Abdullah, S. Iftikhar, S. Bawazeer and M. Izhar, "Enhancing Blockchain Security Against Data Tampering: Leveraging Hybrid Model in Multimedia Forensics and Multi-Party Computation for Supply Chain Data Protection," IEEE Access, 2024.

[19] P. Thantharate and A. Thantharate, "ZeroTrustBlock: Enhancing security, privacy, and interoperability of sensitive data through ZeroTrust permissioned blockchain," Big Data and Cognitive Computing, vol. 7, no. 4, pp. 165, 2023.

[20] Y. Xue and J. Wang, "Design of a Blockchain-Based Traceability System with a Privacy-Preserving Scheme of Zero-Knowledge Proof," Security and Communication Networks, no. 1, pp. 5842371, 2022.

[21] Z. Mahmood and V. Jusas, "Blockchain-enabled: Multi-layered security federated learning platform for preserving data privacy," Electronics, vol. 11, no. 10, pp. 1624, 2022.

[22] S. Malik, V. Dedeoglb, S. S. Kanhere and R. Jurdak, "PrivChain: Provenance and privacy preservation in blockchain enabled supply chains," IEEE, pp. 157-166, 2022.

[23] A. Alabdulatif, "Blockchain-Based Privacy-Preserving Authentication and Access

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
Control Model for E-Health Users," Information, vol. 16, no. 3, pp. 219, 2025.

[24] T. Hamadneh, B. Batiha, G. Mousa Gharib, Z. Montazer, M. Dehghani, W. Aribowo, G. Dhiman, H. Monadhe, R. Kareem Jawad, I. Kasim Ibraheem and K. Eguchi, "Paper Publishing Based Optimization: A New Human-Based Metaheuristic Approach for Solving Optimization Tasks," International Journal of Intelligent Engineering & Systems, vol. 18, no. 2, 2025.

[25] H. Albedran, S. Alsamia and E. Koch, "Flower fertilization optimization algorithm with application to adaptive controllers," Scientific Reports, vol. 15, no. 1, pp. 6273, 2025.

[26] . L. Lopez-Fernandez, M. Gallego, B. Garcia, D. Fernandez-Lopez and F. Javier Lopez, "Authentication, Authorization, and Accounting in WebRTC PaaS Infrastructures," 2014.

[27] L. Philippe Sondeck, M. Laurent and V. Frey, "Discrimination rate: an attribute-centric metric to measure privacy. Annals of Telecommunications," vol. 72, pp. 755-766, 2017.

[28] A. Zhou, J. Yang, Y. Gao, T. Qiao, Y. Qi, X. Wang, Y. Chen, P. Dai, W. Zhao and C. Hu, "Brief industry paper: Optimizing memory efficiency of graph neural networks on edge computing platforms," IEEE, pp. 445-448, 2021.

[29] S. Farah, M. Younas Javed, A. Shamim and T. Nawaz, "An experimental study on performance evaluation of asymmetric encryption algorithms," Information Science, pp. 121-124, 2012.

[30] T. Chen, K. Tang, G. Chen and X. Yao, "Analysis of computational time of simple estimation of distribution algorithms," IEEE Transactions on Evolutionary Computation, vol. 14, no. 1, pp. 1-22, 2010.

Received: August 08, 2025