scientific reports

OPEN

Privacy protection of communication networks using fully homomorphic encryption based on network slicing and attributes

Wei Wang[1]✉, Rong Liu[2] & Silin Cheng[1]

At present, social networks have become an indispensable medium in people's daily life and work. However, concerns about personal privacy leakage and identity information theft have also emerged. Therefore, a communication network system based on network slicing is constructed to strengthen the protection of communication network privacy. The chameleon hash algorithm is used to optimize attribute-based encryption and enhance the privacy protection of communication networks. On the basis of optimizing the combination of attribute encryption and homomorphic encryption,, a communication network privacy protection method using homomorphic encryption for network slicing and attribute is designed. The results show that the designed network energy consumption is low, the average energy consumption calculation is reduced by 8.69%, and the average energy consumption calculation is reduced by 14.3%. During data transmission, the throughput of the designed network can reach about 700 Mbps at each stage, which has a high efficiency.. The above results demonstrate that the designed communication network provides effective privacy protection. Encrypted data can be decrypted and tracked in the event of any security incident. This is to protect user privacy and provide strong technical support for communication network security.

Keywords Communication network, Network slicing, Attribute-based encryption, Fully homomorphic encryption, Privacy protection

Recently, there have been frequent incidents of online privacy infringement, seriously affecting the lives of the parties involved. Privacy security issues have become a hot topic of concern for scholars[1]. With the development of information and communication technology, more people connect to communication networks that facilitate communication with family and friends, business activities, and offer convenience. However, people may inadvertently disclose their personal privacy when using the Internet, causing some negative effects, and even being exploited by illegal elements to harm their own interests. Social communication networks frequently collect a large amount of personal data from users without providing sufficient transparency or control over the collection, storage, use, and sharing of these data. This lack of transparency and control can easily lead to privacy breaches. Moreover, social network sites may have security vulnerabilities that make personal information vulnerable to hacker attacks. These security threats are constantly evolving and increasing in severity. Therefore, encrypting personal privacy has become an important means of privacy protection at present. The commonly used encryption algorithms currently include Attribute-Based Encryption (ABE) and Fully Homomorphic Encryption (FHE). The ABE is proposed based on identity-based encryption algorithms. This can associate attribute sets with access structures, cipher-text, and keys, while also possessing the advantages of identity-based encryption[2]. The FHE was proposed in the twentieth century, allowing data users to directly process the cipher-text of data without having to download and process the data before uploading them[3]. However, the ABE and FHE have some drawbacks. The ABE requires obtaining the public key certificate of the data consumers, which takes up a lot of bandwidth and is inefficient in processing. The key required for using the FHE is too large and prone to cipher-text explosion. Although these traditional encryption technologies ensure the security of data

[1]China Mobile Group Shaanxi Company Limited, Xi'an 710000, China. [2]China Mobile Group Design Institute Company Limited Shaanxi Branch, Xi'an 710000, China. ✉email: wangweijingli34@163.com

transmission in multiple fields, their efficiency in handling massive amounts of data cannot be ignored. These algorithms require a large amount of complex mathematical operations to complete the encryption and decryption processes, especially in the context of big data and cloud computing. This leads to a significant increase in computational latency and consuming a large amount of processor resources. This not only poses a challenge for resource limited devices, but also limits the application of encryption technologies in large-scale data environments. In addition, the limitations of traditional encryption methods in processing speed and energy efficiency become more prominent with the Internet of Things and edge computing. Therefore, existing data protection solutions urgently need to be optimized to adapt to these emerging technological trends. Therefore, this study first constructs a communication network architecture based on network slicing. This architecture can address the limitations of current encryption technologies in protecting user privacy and improve the efficiency and effectiveness of privacy protection mechanisms. Next, the ABE is optimized using chameleon hash function to make attributes traceable for more accurate data processing and transmission in the network slicing environment. Then, a Traceable Attribute-Based Homomorphic Encryption Algorithm (BTA-AHEA) is designed by combining the improved ABE with FHE. BTA-AHEA is applied to the network slice-based communication network to improve the efficiency of encryption and decryption and confidentiality to improve the personal privacy security of the communication network. Users can use the BTA-AHEA to encrypt the private information and send it via the communication network system. In the event of a security incident, it is important to have a traceable mechanism for designing algorithms. This mechanism can ensure that encrypted data are decrypted and tracked to protect user privacy. The tracking mechanism can effectively record and manage user operations, ensuring the stable system operation. At the same time, this mechanism can timely detect and handle potential security threats, such as malicious user attack behavior, thereby improving system security. The tracking mechanism assists users in comprehending their operations, enabling them to make informed decisions and enhance their experience. Tracking mechanisms can monitor the real-time usage of user data, effectively preventing illegal use or leakage of data to protect user privacy. Users can use tracking mechanisms to understand how their data is being used, achieving better privacy protection. At the same time, the tracking mechanism provides detailed reports on the use of data, allowing users to have more complete control over their data and further protecting user privacy. The BTA-AHEA contributes to enhancing the efficiency and security of data encryption and decryption, thereby providing more reliable for network communication security. The contribution of this paper is the development of the BTA-AHEA, which combines the advantages of ABE and FHE, to provide a powerful encryption solution for network communication. BTA-AHEA improves the efficiency and security of data encryption and decryption and introduces traceability features to protect user privacy in the event of security incidents. Meanwhile, BTA-AHEA holds significant importance in the context of network slicing. The characteristic of network slicing is the ability to flexibly provide diverse services, which requires efficient and secure encryption and decryption of data. Therefore, the designed BTA-AHEA can better adapt to these requirements. The research content mainly includes five parts. The first part is the background of privacy protection encryption technology. The second part is a review of the current research status of online privacy protection measures both domestically and internationally. The third part entails designing the communication network system. The first section constructs a communication network based on network slicing. The second section improves the ABE. The improved algorithm is combined with the FHE to form the BTA-AHEA, which is applied to the communication network designed on the basis of network slicing. A communication network based on network slicing and the BTA-AHEA is designed. The fourth part is the performance analysis of communication networks based on network slicing and the BTA-AHEA. The first section is the performance analysis of communication networks and BTA-AHEA. The second section is the performance analysis of communication networks based on network slicing and BTA-AHEA in practical applications. The fifth part is a summary of the entire content. The shortcomings of the research and future prospects are pointed out.

Related works

With the acceleration of the Internet process, the network not only brings convenience to people, but also poses a great threat to user privacy. Relevant professionals have proposed many privacy protection measures to prevent user privacy breaches. Li et al. designed a pervasive intelligent federated learning privacy protection scheme to solve the insufficient computing power and data security threats of machine learning under edge computing. The scheme realized secure transmission by training some models in embedded devices and adding matrix masks. The results showed that the accuracy and efficiency of the scheme were high[4]. Xu et al. designed a cloud-assisted certified public audit scheme for medical wireless sensor networks to address privacy issues in medical data. This scheme enriched wearable sensor functions through cloud storage services and enabled data sharing. The results showed that this scheme achieved a high security[5]. Yang et al. designed an identity-based blockchain aggregation signature scheme to address the shortcomings and security vulnerabilities of existing aggregation signature schemes. This scheme utilized blockchain technology and identity verification mechanisms to achieve data aggregation, signature, and encryption operations. The results showed that this scheme had lower computational and communication costs[6]. Zhao et al. designed a federated learning algorithm based on local differential privacy to avoid privacy threats and reduce communication costs. The algorithm combined three output mechanisms with a suboptimal mechanism. The results showed its capability to ensure practicality whilst guarding privacy[7]. Gope et al. designed a privacy protection authentication scheme for the Internet of Vehicles based on energy Internet to ensure safe, efficient, and reliable operation. This scheme enabled users to securely access the services provided by the provider through symmetric keys established between users. The results showed that the algorithm performed well in detecting electronic intrusions[8]. Xing et al. designed a location privacy protection method based on double K anonymity to protect the location of users in social vehicle networking. This method utilized cloud servers to isolate users and provided services to hide users' location and request information. The results showed that this method maximized the protection of users' location privacy[9]. Shouran et al. designed a privacy and security protection scheme based on the Internet of Things to protect the data of users or enterprises in smart homes. This scheme not only managed personal data, but also provided reliable services to users. The results showed that this scheme effectively protected sensitive data[10]. Lu et al. proposed a secure data sharing scheme utilizing blockchain and federated learning to enhance the security and privacy of private industrial Internet of Things data. This scheme leveraged federated learning to address data sharing constraints and integrated it into the blockchain for licensing. The results showed that this scheme had high accuracy and security[11].

Liu et al. designed a gated recurrent unit neural network algorithm for precise traffic prediction while preserving privacy using federated learning. This algorithm updated the universal learning model through a secure parameter aggregation mechanism. The results showed that the algorithm yielded highly accurate traffic predictions without compromising data prediction[12]. Sun et al. proposed a machine learning-based privacy protection algorithm to prevent privacy leakage in the sixth generation communication network. The algorithm solved the maximum likelihood in the sixth generation communication through machine learning. The results demonstrated its effectiveness in protecting privacy[13]. Gai et al. designed a model licensed blockchain edge model for smart grid networks to address privacy and security issues such as infrastructure mapping attacks in smart grid networks. This model utilized group signature and covert channel authorization techniques to ensure user legitimacy. The experiment showed that this model's security awareness strategy was better[14]. Yao et al. designed a privacy protection method based on convolutional neural networks and the Paillier algorithm to prevent malicious tampering of information. It utilized convolutional neural networks to detect abnormal behavior in metric data, while using the Paillier algorithm to protect energy privacy. The results showed that this method had high accuracy[15]. Qu et al. designed a blockchain-supported federated learning scheme to ensure a reasonable balance in the event of poisoning attacks. It utilized a blockchain-based global learning model to locally learn and update terminal device exchanges. The results showed that this scheme performed well in terms of privacy protection[16]. Lu et al. designed a blockchain-based federated learning approach using digital twin edge networks to improve communication security and safeguard the data privacy of the Internet of Things. The scheme employed authorized federated learning on the blockchain to improve communication security and data privacy protection of digital twin networks. The results indicated that this scheme significantly improved the security of data[17]. Zhang et al. designed an industrial Internet of Things framework based on edge services and blockchain support to address the issue of distributed edge services causing malicious attacks on data. It utilized the edge transaction approval mechanism of cross-domain shared edge resources and credit differentiation. The results showed that the framework had high security in edge service management[18].

In summary, current researchers have proposed many network security technologies. However, there are still many shortcomings in privacy protection and encryption and decryption efficiency. Therefore, the study combines network slicing technology with the optimization of the ABE. Meanwhile, a chameleon hash function is introduced to propose a new BTA-AHEA. This algorithm improves both the efficiency of encryption and decryption as well as the security of communication networks, providing a new solution to solve the problems faced in the current network security.

Communication network based on network slicing and BTA-AHEA

The first section of this paper mainly constructs a communication network architecture designed based on network slicing. The second section mainly designs the BTA-AHEA and sets related functions.

Communication network architecture based on network slicing

All symbols and their meanings in the text are shown in Table 1.

Communication network systems have evolved from analog to digital communication and modern Internet networks. The systems' initial form is point-to-point analog communication. These systems gradually develop into digitalization, incorporating complex modulation and demodulation technologies to improve communication efficiency. In the twenty-first century, the Internet's popularity has driven communication network systems towards more efficient and secure directions, meeting the needs of information transmission speed and security. This study designs a communication network management architecture based on network slicing. This architecture can address the low resource utilization, complex network structure, and high cost of dedicated communication hardware in traditional communication networks. Network slicing is an innovative way of creating virtual network architecture on demand. It enables the sharing of the same infrastructure while dividing it into independent virtual networks, each possessing its unique characteristics. It is crucial to configure the virtual network adequately according to third-party requirements and the diverse functions of each network to cater to user needs and provide personalized service. The main technologies of network slicing include Network Function Virtualization (NFV) and Software Defined Network (SDN)[19]. NFV can virtualize traditional proprietary hardware by installing a network on a virtual server. Utilizing NFV does not require dedicated hardware for each network function, which can improve network operational efficiency. SDN can implement network virtualization, which is typically used to manipulate the configuration of virtual machines on the core cloud or edge. It separates network forwarding and control functions of the network, thus creating a network that can be centrally managed and programmed. It manages network traffic through programming, achieves flexible resource allocation, and achieves the goal of providing one-on-one services to users[20]. The network slicing structure based on NFV and SDN is shown in Fig. 1.

In Fig. 1, the network slicing structure includes five parts: NFV management and orchestration, SDN controller, virtual resource layer, hardware resource layer, and operation and business support system. NFV management and orchestration mainly involves the virtualization of network functions and the management of virtual network

$\boldsymbol{F(x)}$	**Determining factor of migration**
x	Service node
$\sum(\cdot)$	Weighted average of all elements in a column
y	Starting point of migration
$T_d(x)$	Delay during the transmission process
$S_c(x)$	Size of the server capacity
$B_{\max}(x,y)$	Maximum bandwidth between two service nodes
$R_d(x)$	Delay in the return journey between two service nodes
$C_p(x)$	Cost price of the server
β	Return on migration time
n	Number of nodes
C_n	Total amount of migrated data
T	Migration cycle
R_n	Channel transmission rate
t	Migration time
$C_{n-1}(t)$	Amount of data at the current time
v	Mobile terminal speed
$R_n(th)$	Threshold of the minimum migration rate of the current time point node
$\overline{E}$	Average energy consumption of data migration
E_t	Energy consumption of the current node
P	Transmission power
$E_{\min}$	Minimum migration energy consumption
CH_Gen	Key generation function
λ	Safety parameter
P_k	Public key for outputting the chameleon hash
S_k	Private key of the chameleon hash
CH_Hash	Hash generation function
r	A random number
m	Any information
h	Hash value
p	A random value
CH_Ver	Hash collision function
m'	New information
r'	A new random number
$P_k(CH)$	Public key for the temporary threshold of the chameleon hash function
$S_k(CH)$	Private key for the temporary threshold of the chameleon hash function
N_1	A composite number obtained through an asymmetric encryption algorithm
e	A prime number
H_1	Hash function
d_1	Threshold value
$mP_k(ABE)$	Public key of the BTA-AHEA input
$mS_k(ABE)$	Private key of the BTA-AHEA input
p'_1, p'_2,p'_3	Subgroups of the composite group
N	A composite number
CT	Cipher-text
$\overline{CT}$	Set of cipher-text
CT'	Updated cipher-text
s	A random vector
t	A random integer
N_2	A composite number
r	A random number
D	Set of decryption functions
$\overline{K'}$	Set of keys
r'_1,r'_2	The random numbers corresponding to the two threshold values d_1 and d_2

Table 1. Comparison of symbols and their specific meanings.

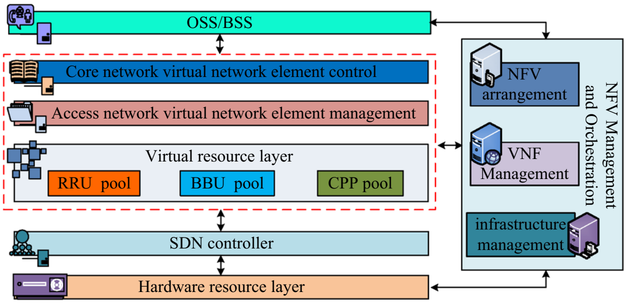

OSS/BSS
Core network virtual network element control
Access network virtual network element management
Virtual resource layer
RRU pool
BBU pool
CPP pool
SDN controller
Hardware resource layer
NFV Management and Orchestration
NFV arrangement
VNF Management
infrastructure management

Figure 1. Network slicing structure diagram based on NFV and SDN.

functional units and infrastructure. The SDN controller is primarily utilized to detect the network's topology structure and control traffic. The SDN controller cam manage the discovered network topology, network security, and traffic in the network. The virtual resource layer consists of three parts: radio frequency far-off pool, a centralized baseband processing pool, and a centralized protocol processing pool. The hardware resource layer consists of physical storage, physical network, and physical computing resources. The operational and business support system primarily manages the virtual network components of the core and access networks, including virtual network elements and resources. The research aims to implement the management and compilation functions of network slicing through a virtual platform. Its structure is shown in Fig. 2.

In Fig. 2, the virtual platform consists of four parts: the service, driver, business management, and blade management systems. The service system is primarily responsible for cataloging and managing user-related data. The business management system monitors and configures. The drive system handles drive management. The blade management system manages all server blades. Server migration is required to enhance network stability, security, and performance. The migration of network services is associated with the network virtual directory system. The migration determining factors can be obtained through the SDN controller. The calculation method for the migration determining factors is shown in Eq. (1).

$$F(x) = \sum \left[(T_d(x), S_c(x), B_{\max}(x, y), R_d(x), C_p(x) \right] \tag{1}$$

In Eq. (1), $F(x)$ represents the determining factor of migration. x represents the service node. $\sum(\cdot)$ represents the weighted average of all elements in a column. y represents the starting point of migration. $T_d(x)$ represents the delay during the transmission. $S_c(x)$ represents the size of the server capacity. $B_{\max}(x, y)$ represents the maximum bandwidth between two service nodes. $R_d(x)$ represents the delay in the return journey between two service nodes. $C_p(x)$ represents the cost price of the server. The matrix serves as the determining factor for migration, and normalizing it results in the lowest communication network cost. This study utilizes mobile edge computing to depict the connection between mobile technology and edge computing. It also uses mobile edge computing to regulate storage and computing resources[21]. The network architecture of mobile edge computing is shown in Fig. 3.

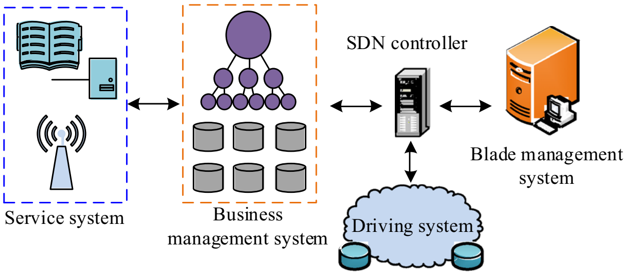


Figure 2. Virtual platform architecture.

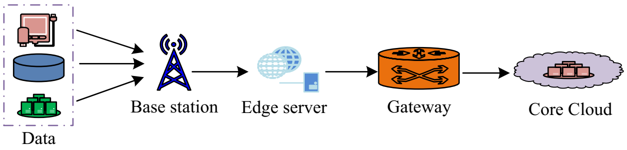


Figure 3. Network architecture of mobile edge computing.

In Fig. 3, the addition of an edge server between the base station and the gateway can provide computation and storage for complex commands. Meanwhile, the transmission time of commands in the network can be reduced. The migration time revenue is introduced to judge whether the current node has revenue to reduce the energy consumption in mobile edge computing. The specific expression is shown in Eq. (2)[22].

$$\beta = \frac{C_{n-1}}{R_{n-1}} - nT - \frac{C_n}{R_n} \tag{2}$$

In Eq. (2), β represents the return on migration time. n represents the number of nodes. C_n represents the total migrated data. T represents the migration cycle. R_n represents the channel transmission rate, that is, the migration rate. The larger the value of migration time benefit, the shorter the migration time, and the lower the energy consumption. The calculation method for migration time is shown in Eq. (3)[23].

$$t = \frac{C_{n-1}(t) + vnT}{R_n - c} \tag{3}$$

In Eq. (3), t represents the migration time. $C_{n-1}(t)$ represents the amount of data at the current time. v represents the mobile terminal speed. The next step is to calculate the threshold for the minimum migration rate of the current node at each time point, as shown in Eq. (4)[24].

$$R_n(th) = [C_n(t) + \frac{cC_{n-1}}{R_{n-1}}]/[\frac{C_{n-1}}{R_{n-1}} - nT] \tag{4}$$

In Eq. (4), $R_n(th)$ represents the threshold of the minimum migration rate of the current time point node. The average energy consumption of the data is calculated based on the total energy consumption during migration, as shown in Eq. (5)[25].

$$\overline{E} = \frac{nE_t + Pt}{C_{n-1}(t) + c(nT + t)} \tag{5}$$

In Eq. (5), $\overline{E}$ represents the average energy consumption of data migration. E_t represents the energy consumption of the current node. P represents the transmission power. The minimum migration energy consumption can be obtained through the average migration energy consumption. This can be achieved by making the migration time benefit greater than 0, as shown in Eq. (6)[26].

$$E_{\min} = \min\left\{E(nE_t) - \overline{E} \times E[C_{n-1}(t) + c(nT + t)]\right\} \tag{6}$$

In Eq. (6), $E_{\min}$ represents the minimum migration energy consumption. The flow of mobile edge computing to reduce energy consumption is shown in Fig. 4.

In Fig. 4, the comparison between the energy consumption necessary for migration and the channel transmission rate determines whether the transmission should proceed. This leads to lower energy usage and increased energy efficiency.

Network privacy protection based on BTA-AHEA algorithm

Chameleon hash algorithm

Network slicing brings convenience and reduces the cost of communication networks, which also brings security risks. On the one hand, the security risks of network slicing include traditional network security risks such as Distributed Denial of Service (DDOS) attacks and routing security policies. These security risks are not much different from the risks faced by traditional network technologies. On the other hand, new security risks are introduced by the technical characteristics of network slicing itself. Traditional communication network security defense mainly adopts boundary security defense architecture. It is assumed that all attack eavesdropping comes from the outside and is completely trustworthy internally. However, this assumption is often not true in architecture based on network slicing. Because the technology in network slicing breaks the traditional physical boundaries of the network or system. Therefore, the protection of user privacy and data security must shift from boundary defense architecture to endogenous security defense architecture for slice-based communication networks. Currently, commonly used encryption algorithms include keyword searchable public key encryption, identity-based encryption, the ABE, etc. Among them, keyword searchable public key encryption searches

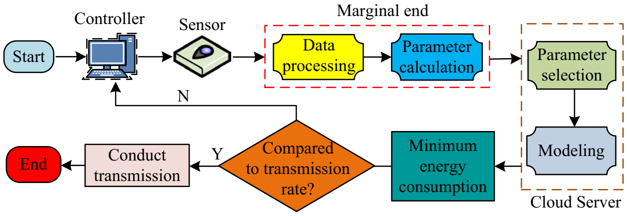


Figure 4. Flow chart of mobile edge computing to reduce energy consumption.

encrypted data based on the corresponding keyword provided by the user, thereby finding the file associated with the keyword[27]. Identity-based encryption is a public key encryption technology that implements identity authentication and message encryption. It uses the user's identity information as the public key and generates a private key by a third party to achieve encryption. ABE is an encryption method that represents a user's identity as a set of multiple attributes, without requiring knowledge of the user's personal information. Only users who meet the attribute requirements can decrypt. In the ABE, the encryptor defines a set of access policies that only users who meet these policies can decrypt the corresponding cipher-text. The Homomorphic Encryption (HE) algorithm is an encryption method implemented to secure data privacy. It involves encryption functions that transform plaintext into cipher-text and decryption functions that reverse the process from encrypted cipher-text to pre-encrypted plaintext. The encryption methods include addition homomorphism and multiplication homomorphism[28]. These methods allow calculations to be performed on encrypted data and generates encrypted results. Unlike FHE, HE provides support for only a limited set of computational operations that include addition, multiplication, and exponential operations. The FHE refers to the ability to simultaneously satisfy two homomorphic operations and perform multiple homomorphic addition and multiplication operations. It allows calculations to be performed on encrypted data and generates encrypted results. In the decryption phase, a private key can decrypt is used and plaintext results are obtained. Then, the equivalence of cipher-text operations and encryption of plaintext operations can be ensured. The security of information transmission between communication parties can be ensured by applying the FHE to a communication network based on network slicing. The chameleon hash algorithm can generate different hash values. However, with a specific private key, these hash values can be mapped to the same value. This mechanism increases the difficulty of cracking and improves the security of encryption. At the same time, it can generate corresponding encryption keys based on the user's attributes. The chameleon hash algorithm allows only users with matching attributes to decrypt and access data, thereby achieving fine-grained control over data access. Therefore, this study introduces the chameleon hash function to optimize the ABE, combining the optimized algorithm with FHE to design the BTA-AHEA. The specific framework of this algorithm is shown in Fig. 5.

In Fig. 5, BTA-AHEA consists of five parts: attribute authorization center, blockchain, cloud server, data owner, and data user. Among them, the main function of the attribute authorization center is to collect the collection of identity attributes uploaded by data users and generate private keys for data users. The blockchain is mainly used as a database for storing information. The cloud server is mainly used to return the content queried by data users. The attribute-based FHE mainly includes five parts: parameter generation, public key extraction, encryption, decryption, and homomorphic operation. Firstly, the security parameter 1^{λ} is input to generate the public parameter P_p and the master key M_k. Then, the public parameter P_p, master key M_k. A user's attribute list L are input to obtain the user's private key SK_L. Next, the public parameter P_p, private key M_k, access restriction parameter W, and plaintext $p \in \{0, 1\}$ are input to generate ciphertext c. The public parameter P_p, private key SK_L, and ciphertext c are entered. If the user's attribute list a meets the access restriction W, plaintext p is output. Otherwise, the output is empty. Finally, under the same access rights and responsibilities, k ciphertext information $c_1, c_2, ..., c_k$ and common parameter P_p are input. A new ciphertext c_f that satisfies $(P_p, c_f, SK_L) = f(p_1, p_2, ..., p_k)$ is calculated and output. Hash functions are primarily utilized in blockchain cryptography to convert strings of varying lengths, types, or sizes to a constant length string. The constant length string is then utilized to digitally sign documents within the network. Generally, this output string is called a hash value or hash digest[29]. Traditional hash functions have collision resistance. Chameleon hash functions can easily find collisions, thereby improving computational efficiency. The chameleon hash algorithm includes four steps: key generation, hash generation, hash verification, and hash collision. The key generation algorithm outputs both the public and private keys of the chameleon hash by utilizing a random seed as input to produce a key of specific length. The key generation function usually uses cryptographically secure pseudo-random number generation algorithms to ensure the randomness and security of the generated key[30]. The specific calculation method is shown in Eq. (7).

$$CH\_Gen(1^{\lambda}) = (P_k, S_k) \tag{7}$$

In Eq. (7), CH_Gen represents the key generation function. λ represents the safety parameter. P_k represents the public key for outputting the chameleon hash. S_k represents the private key of the chameleon hash. The next step is to use the result generated by the key as input and calculate the hashed message to generate a hash value. The specific hash generation algorithm uses keys and messages as inputs and calculates the hash value utilizing a combination of nonlinear functions, bit operations, and permutation operations. The calculation method for hash generation is shown in Eq. (8).

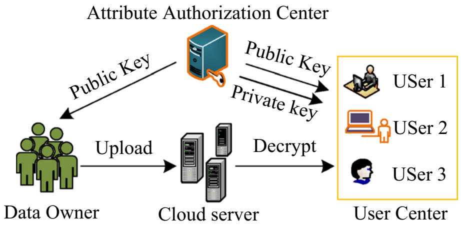


Figure 5. Specific framework of BTA-AHEA.

$$CH\_Hash(P_k, m, r) = (h, p) \tag{8}$$

In Eq. (8), CH_Hash represents the hash generation function. r represents a random number. m represents any information. h represents the hash value. p represents a random value. After receiving a hash value, it performs a hash generation operation using the same key and message. The generated hash value is compared with the received hash value. If the two hash values are the same, it is considered that the hash verification is successful. Otherwise, it is considered that the hash verification failed. The next step is to perform hash verification, as shown in Eq. (9).

$$CH\_Ver[P_k, m, (h, p)] \tag{9}$$

In Eq. (9), CH_Ver represents the hash validation function. The result of hash verification depends on whether (h, p) is correct or not. When the hash value is incorrect, the output is 0. Otherwise, it is 1. The calculation method for hash collision is shown in Eq. (10).

$$\begin{cases} CH\_Cld(sk, m, m', (h, p)) \\ CH\_Ver[P_k, m, (h, p), r] = CH\_Ver[P_k, m', (h, p), r'] = 1 \end{cases} \tag{10}$$

In Eq. (10), CH_Cld represents the hash collision function. m' represents new information. r' represents a new random number. Hash collision occurs when two distinct messages generate the same hash value following a hash generation operation. In the chameleon hash function, there is a possibility of hash collisions. There are two different messages that can generate the same hash value. The probability of a collision occurrence is dependent on the hash generation algorithm and key generation function's properties. The calculation of the chameleon hash algorithm is shown in Fig. 6.

In Fig. 6, the signature obtained through implementation of the chameleon hash algorithm exhibits non-transferability, non-forgeability, and non-repudiation, which can bolster network security.

Attribute based homomorphic encryption algorithm combined with chameleon hash algorithm

Then, the chameleon hash algorithm is applied to attribute-based homomorphic encryption. Firstly, a temporary threshold is set for the chameleon hash function, and its public and private keys are set. The specific expression is shown in Eq. (11).

$$\begin{cases} P_k(CH) = \{\lambda, N_1, e, H_1\} \\ S_k(CH) = d_1 \end{cases}, e < N_1 \tag{11}$$

In Eq. (11), $P_k(CH)$ represents the public key for the temporary threshold of the chameleon hash function. $S_k(CH)$ represents the private key for the temporary threshold of the chameleon hash function. N_1 represents a composite number obtained through an asymmetric encryption algorithm. e represents a prime number. H_1 represents the hash function. d_1 represents the threshold value. The next step is to set the public and private keys for the BTA-AHEA input to ensure the confidentiality, integrity, and ability to track attributes of encrypted data, as expressed in Eq. (12).

$$\begin{cases} mP_k(ABE) = \left\{N, w, u, h, v, g, F_1 = g^{\alpha}, F_2 = g^{b}, e(g, g)^{\alpha}\right\} \\ mS_k(ABE) = \left\{p'_1, a, b, g^{\alpha}, g_3, R_0\right\} \\ N = p'_1 p'_2 p'_3 \end{cases} \tag{12}$$

In Eq. (12), $mP_k(ABE)$ represents the public key of the BTA-AHEA input. $mS_k(ABE)$ represents the private key of the BTA-AHEA input. p'_1, $p'_{2,}$ and p'_3 represent subgroups of the composite group. N represents a composite number, $w, u, h, v \in p_1$. α, a, b belong to integers. $g \in p'_3$. $R_0 \in p'_3$. Next, the properties of the hash function are used to generate cipher-text. The temporary threshold of the chameleon hash function and the public and private keys of the BTA-AHEA input are used to represent the public and private keys of the chameleon hash function. The data users upload their identity attributes to the attribute authorization center, which will result in the generation of a private key for them. Then, they input the public key, information, and access policy matrix through a hash function, parse the temporary threshold, and obtain the cipher-text, which is expressed in Eq. (13)[31].

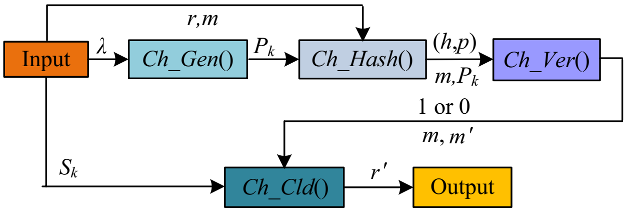


Figure 6. The calculation process of chameleon hash algorithm.

$$\begin{cases} \overline{CT} = \left(C, C_0, C_1, C_2, \left\{C_{i,1}, C_{i,2}, C_{i,3}\right\}_{i\in[l]}, CT'\right) \\ CT' = Enc_{SE}(k, d_2) \\ C = \overline{K} \times e(g,g)^{\alpha,s}, C_0 = g^s, C_1 = (g^\alpha)^s, \\ C_2 = (g^b)^s, \overline{K} = encode(k, r) \\ CT = (\overline{CT}, C', t), C' = (wu^{\overline{h}}v^t)^s \end{cases} \tag{13}$$

In Eq. (13), CT represents cipher-text. $\overline{CT}$ represents the set of cipher-text. CT' represents the updated cipher-text. s represents a random vector. t represents a random integer. N_2 represents a composite number. The next step is to obtain the hash value and random number, as shown in Eq. (14).

$$(z, r) = [(h', N_2, F_2, CT), r'] \tag{14}$$

In Eq. (14), z represents the hash value. r represents a random number. It uploads the obtained hash value and random number (z, r) to the cloud server and inputs the public key of the chameleon hash function and message S. The data user verifies whether the condition is met through hash verification. If the condition is met, the output is 1. Otherwise, the output is 0. The data user calculates the set of decryption functions and the set of keys, as shown in Eq. (15)[32].

$$\begin{cases} D = e(w,g)^{(\alpha+T)r_u s} \\ \overline{K'} = \dfrac{\overline{K'}e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} \end{cases} \tag{15}$$

In Eq. (15), D represents the set of decryption functions. $\overline{K'}$ represents the set of keys. Then, it is validated. If $\overline{C'} = C'$, it represents that CT is valid. Among them, $C' = (u^{\overline{h}}v^t w)^s$. Finally, it calculates a new random number, expressed in Eq. (16).

$$\begin{cases} r_1' = [y_1(x_1'^{-1})]^{d_1} \bmod N_1 \\ r_2' = [y_2(x_2'^{-1})]^{d_2} \bmod N_2 \end{cases} \tag{16}$$

In Eq. (16), r_1' and r_2' represent the random numbers corresponding to the two threshold values d_1 and d_2. If the verification is successful, the data user can successfully use information m' instead of information m. At this point, data encryption for the user is completed. The process is illustrated in Fig. 7.

Figure 7 depicts the utilization of input security parameters for data initialization and processing through asymmetrical and designed encryption algorithms. The next step is to set the temporary threshold key and public key for the chameleon hash algorithm. The public and private keys are set for the BTA-AHEA. The public and private keys are set for the chameleon hash algorithm. The next step is to calculate the cipher-text through the encryption algorithm and use it as input to verify whether the access policy matches the attribute set. If there is a match, the decrypted information will be output. Otherwise, the termination symbol will be output. Finally, the tracking algorithm checks the private key's form in the attribute authorization center. If the form is correct, 1 is output, indicating that the identity code of the data user can be extracted from the private key. Otherwise, it outputs 0. The tracking algorithm outputs the termination symbol. It applies traceable attribute-based FHE to a communication network based on network slicing. Then, a communication network is obtained using FHE methods based on network slicing and attributes. Network slicing technology can divide a physical network into multiple virtual networks. Each virtual network can independently allocate resources and configure policies as needed. This feature improves the customizability and flexibility of the communication network, enabling it to better meet different business and service requirements. At the same time, selecting attributes as the fundamental element of the system is to leverage the advantages of BTA-AHEA. This method can effectively manage and utilize data while ensuring data security. Data users only need to meet the access policies of the data to obtain the corresponding data by associating the identity attributes of data users with the data. Data users need not to

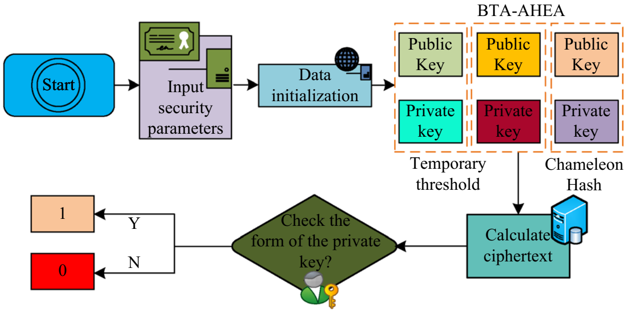


Figure 7. User data encryption flowchart.

know the actual storage location of the data. This not only improves the efficiency of data usage, but also ensures the security of the data.

Analysis of communication network results based on network slicing and BTA-AHEA

The first section of this paper mainly analyzes the performance of the communication network and the BTA-AHEA designed on the basis of network slicing. The second section mainly analyzes the effectiveness of the BTA-AHEA applied to communication networks designed based on network slicing.

Performance analysis of communication networks and BTA-AHEA algorithm based on network slicing

A study was conducted on the Core Xeon processor 4210rA2.40 GHz to verify the performance of the designed BTA-AHEA communication network system. The experiments were conducted on Ubuntu 18 cloud server with 16 GB of running memory and Python 3.6. In Ubuntu 18 cloud server, first, the environment was set up. Then, the BTA-AHEA system was deployed and configured. Experimental parameters were adjusted by writing and running automation scripts. Finally, the Seal database (https://link.zhihu.com/?target=https%3A//github.com/microsoft/SEAL) was used to conduct experiments. Seal is an easy-to-use open-source homomorphic encryption library developed by Microsoft's cryptography and privacy research group. It is written in modern standard C++ and is easy to compile and run in many different environments. The parameters used in the experiment include private key generation time, cipher-text generation time, decryption generation time, and bit error probability. Runtime can directly reflect the total time required for the system to perform specific operations. Shorter runtime means higher efficiency and better user experience. In network communication, the waiting time for users to start using the system is directly affected by the system key generation time. Therefore, the length of key generation time largely determines the quality of the user experience. The speed of processing encryption and decryption directly affects the efficiency and real-time performance of data transmission while ensuring data security. The storage cost of tracking malicious users is a measure of the storage resources required by the system to track malicious user behavior. A lower or zero storage cost indicates that the algorithm can complete tracking tasks with minimal resource overhead. Throughput can describe the amount of data that a system can process per unit of time. High throughput indicates that the system can quickly process large amounts of data, demonstrating high-performance data processing capabilities. High throughput is particularly important for real-time communication needs. The probability of bit errors can intuitively reflect the encryption effect of the system. A lower probability of bit errors means better encryption performance, which can better protect the privacy of user data. Therefore, these indicators can comprehensively reflect the performance and efficiency of the system, as well as the encryption effect and privacy protection ability. Optimizing these indicators can improve user experience and meet real-time communication needs, while ensuring the privacy and security of user data. This study first verified the energy consumption of the communication network designed on the basis of network slicing as users increase. Then, it was compared with the migration strategy using deterministic task nodes and the network that used a random task node migration strategy, as shown in Fig. 8.

In Fig. 8, the average energy consumption of the network designed based on network slicing and the network using random task node migration strategy increases with the increase of users. Then, the average energy consumption gradually stabilizes after reaching a certain value. The network that uses a certain task node migration strategy first increases, then decreases, and gradually stabilizes as the user count grows. After reaching stability, the network designed based on network slicing has an average energy consumption of 0.42 J. The average energy consumption of the network using a random task node migration strategy after reaching stability is 0.46 J. The average energy consumption of the network using a determined task node migration strategy after reaching stability is 0.49 J. Compared with the other two networks, the average energy consumption of communication networks based on network slicing has decreased by 8.69% and 14.3%, respectively. The result suggests relatively low energy consumption. This study further verifies the data migration rates of communication networks designed based on network slicing and compares them with the other two networks. The migration rates for varying data generation rates and different node numbers are shown in Fig. 9.

Figure 9a shows that as the data generation rate increases, the average effective data migration rate of the network created using network slicing decreases uniformly. The average effective data migration rate of the

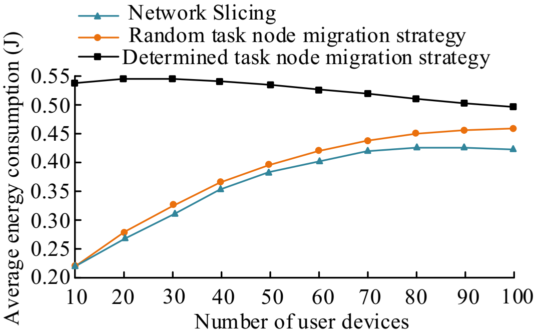
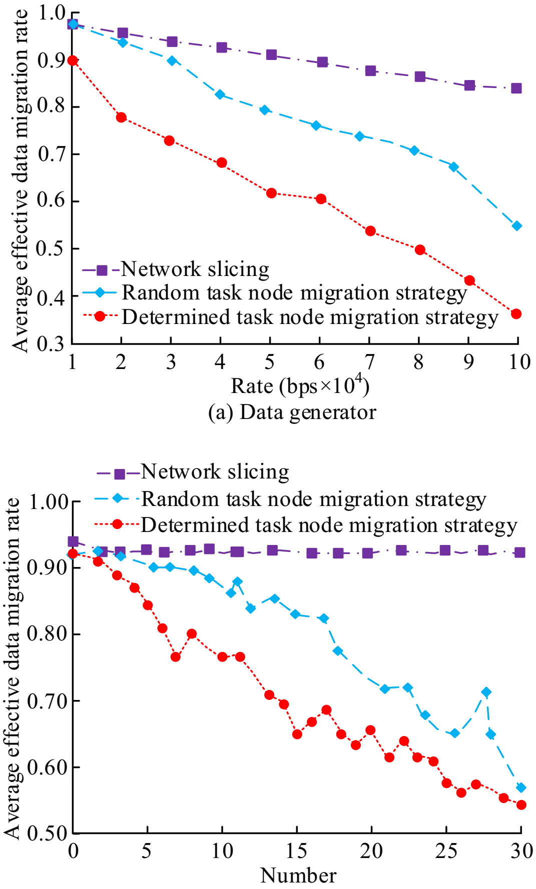


Figure 8. Energy consumption of three communication networks.



(a) Data generator

(b) Number of nodes

Figure 9. Migration rate under different data generation rates and number of nodes.

other two algorithms shows a downward trend. However, the decline speed is greater than that of the network designed on the basis of network slicing. In Fig. 9b, the average effective data migration rate of the communication network designed based on network slicing remains basically unchanged at about 92.5% with the increasing detection nodes. The average effective data migration rate of the other two algorithms gradually decreases. The above results indicate that the employment of network slicing in designing a communication network provides a higher data migration rate. At the same time, the random transfer exhibits similar performance to the proposed method, which can be attributed to the fact that the data was randomly scrambled before training. Randomly shuffling data help to break any patterns that may exist in the data and prevent overfitting. When patterns are absent, the model's learning complexity increases. Therefore, its performance is similar to that of applying random transfer. The next step is to perform performance analysis on the designed BTA-AHEA. The temporary threshold is set to m. The attribute set length of the network system is set to $|l|$. The attribute set length of the user is set to $|S|$. The size of the attribute set that matches the access policy during decryption is set to d. The length of the matrix row of the access policy is set to $|h|$. The storage cost, private key size, and cipher-text length for tracking malicious users are calculated, which are compared with ABE, HE, and FHE. First, the common parameters for each algorithm are determined, which is a constant independent of the length of the message or attribute set. The size of the private key and cipher-text for each algorithm is calculated. This is determined by the length of the attribute set as well as other factors. The storage cost is finally calculated by adding the size of the public parameter, private key, and cipher-text, comparing with ABE, HE, and FHE. The experiment was conducted on a 64 bit Windows 10 operating system and a Core (TM) i7-7700 processor CPU@3.60 GHz (3.60 GHz), running in an environment with 8 GB of memory, as shown in Table 2.

In Table 2, the public parameters of the BTA-AHEA, FHE, and HE algorithms are constants. The public parameters of the ABE vary with the length of the attribute set. The private key size and cipher-text length of the BTA-AHEA are smaller than those of the other three algorithms, whose storage cost for tracking malicious users is 0. The other three algorithms cannot track malicious users. The above results demonstrate that the BTA-AHEA can effectively track malicious users. It has a good privacy protection effect. Next, the throughput of the entire network system encryption using the BTA-AHEA, ABE, FHE, and HE algorithms is calculated, as shown in Fig. 10.

In Fig. 10, during the system initialization phase, the throughput of the BTA-AHEA is about 700 Mbps. The ABE is about 640 Mbps. The FHE is about 680 Mbps. The HE algorithm is about 615 Mbps. During the key generation stage, the throughput of the BTA-AHEA is about 695 Mbps. The ABE is about 600 Mbps. The FHE is

Method	Public parameter size	Private key size	Cipher-text size	Storage cost
BTA-AHEA	12	$4+2\lvert S\rvert$	$3\lvert h\rvert+1$	0
FHE	8	$7+3\lvert S\rvert$	$\lvert h\rvert+4$	None
HE	8	$5+2\lvert S\rvert$	$3\lvert h\rvert+4$	None
ABE	$\lvert l\rvert+6$	$8+\lvert S\rvert$	$2\lvert h\rvert+5$	None

Table 2. Comparison of storage costs of four encryption algorithms. "None" indicates that tracking malicious users is not supported.

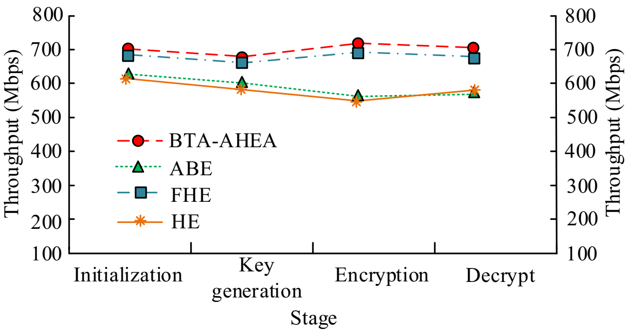


Figure 10. Throughput of the entire network system encryption process.

about 650 Mbps. The HE algorithm is about 590 Mbps. During the encryption phase, the BTA-AHEA achieves a throughput of around 720 Mbps. The ABE, FHE, and HE algorithms achieve approximately 580 Mbps, 675 Mbps, and 585 Mbps, correspondingly. During the decryption phase, the throughput of the BTA-AHEA, ABE, FHE, and HE algorithms is approximately 705 Mbps, 585 Mbps, 660 Mbps, and 595 Mbps, respectively. The analysis shows that the BTA-AHEA has a higher throughput than other algorithms at each stage, indicating its high computational efficiency. Finally, the time required for rating prediction, loss function, weight update, and Epoch in one training cycle of the designed network is calculated to evaluate its overall performance. The method proposed in this study is compared with numerous methods, such as ubiquitous intelligent federated learning privacy protection scheme[4], certificateless public auditing scheme for cloud-assisted medical WSNs[5], location privacy protection method based on double k-anonymity[6], ciphertext policy attribute based encryption[33], and mixed password text key policy attribute based encryption[34]. In particular, in reference[33], an optimized medical data sharing system is proposed based on ciphertext policy attribute encryption. This system achieves lower storage overhead and encryption/decryption time. In reference[34], a hybrid ciphertext key strategy is proposed based on attributes, which has a smaller key size and execution time. The comparison results are shown in Table 3.

Table 3 showcases that the designed network outperforms the methods in references[4–6,33], and[34] in terms of rating prediction, loss function, weight update, and Epoch time. Specifically, the predicted rating time is 55.23×10^3 µs. Compared to the methods in references[4–6,33], and[34], it has decreased by 47.17%, 48.36%, and 70.02%, 9.06%, and 4.59%, respectively. The operation time of the loss function is 13.76×10^3 µs. Compared with the methods in references[4–6,33], and[34], the reduction is 55.26%, 57.54%, 75.66%, 19.63%, and 12.28%, respectively. The weight update time is 6.11×10^3 µs. Compared to references[4–6,33], and[34], it has decreased by 45.68%, 67.34%, 81.07%, 44.25%, and 27.52%, respectively. Finally, the time taken by Epoch is 5.62×10^3 µs. Compared to the methods in references[4–6,33], and[34], it has decreased by 46.78%, 47.48%, 34.58%, and 17.84%, respectively. The above results indicate that the network designed in the study not only has advantages in prediction accuracy, but also shows significant superiority in computational efficiency.

Network	Reference[4]	Reference[5]	Reference[6]	Reference[33]	Reference[34]	This study
Scoring prediction	104.51×10^3 µs	106.87×10^3 µs	183.98×10^3 µs	60.73×10^3 µs	57.89×10^3 µs	55.23×10^3 µs
Loss function	30.75×10^3 µs	32.42×10^3 µs	56.53×10^3 µs	17.12×10^3 µs	15.51×10^3 µs	13.76×10^3 µs
Weight update	11.24×10^3 µs	18.72×10^3 µs	32.26×10^3 µs	10.96×10^3 µs	8.43×10^3 µs	6.11×10^3 µs
Epoch	10.56×10^3 µs	10.70×10^3 µs	20.58×10^3 µs	8.59×10^3 µs	6.84×10^3 µs	5.62×10^3 µs

Table 3. The cost of generating private keys and ciphertext for four algorithms.

Analysis of the actual effect of BTA-AHEA communication network

This study first calculates the costs of private key generation and cipher-text generation. Then, this paper verifies the effectiveness of the designed communication network based on network slicing and BTA-AHEA in practical applications. It is compared with communication networks based on ABE, FHE, and HE. It sets the calculation cost of exponential operation to J_e, linear operation to J_p, multiplication operation to J_m. Other parameters remain unchanged. First, the length of the private key and cipher-text is computed. Based on these lengths, the costs of generating the private key and cipher-text are calculated. The results are shown in Table 4.

In Table 4, the private key generation cost for communication networks based on network slicing and BTA-AHEA is $(4+|S|)J_e+J_m$. The private key generation cost for FHE is $(5+5\,|S|)J_e+(1+2|S|)J_m$. The private key generation cost for HE is$(5+7|S|)J_e+(3+|S|)J_m$. The private key generation cost for ABE is $(4+4|S|)J_e+|S|J_m$. The cost of cipher-text generation in communication networks based on network slicing and BTA-AHEA is $(3+|h \times r|)J_p+(1+|h \times r|)J_m$. FHE has a cipher-text generation cost of $(5+3|h|)J_e+(1+3|h|)J_m$. HE has a cipher-text generation cost of $(5+2|h \times r|+6|h|)J_e+(2+|h \times r|+3|h|)J_m$. ABE has a cipher-text generation cost of $(4+3|h|)J_e+(2+3|h|)J_m$. The computational cost of communication networks based on network slicing and BTA-AHEA is significantly lower than other networks designed on the basis of other algorithms. This proves that the data in the designed network have good fit. The next step is to calculate the running time of the designed communication network and compare it with other networks. At the same time, it calculates the variation of the running time of the communication network designed for tracking malicious users based on bit length. The experiment is conducted using the Java Database Connectivity (JPBC) library on the eclipse platform. The number of settings increases linearly from 10 to 100, as shown in Fig. 11.

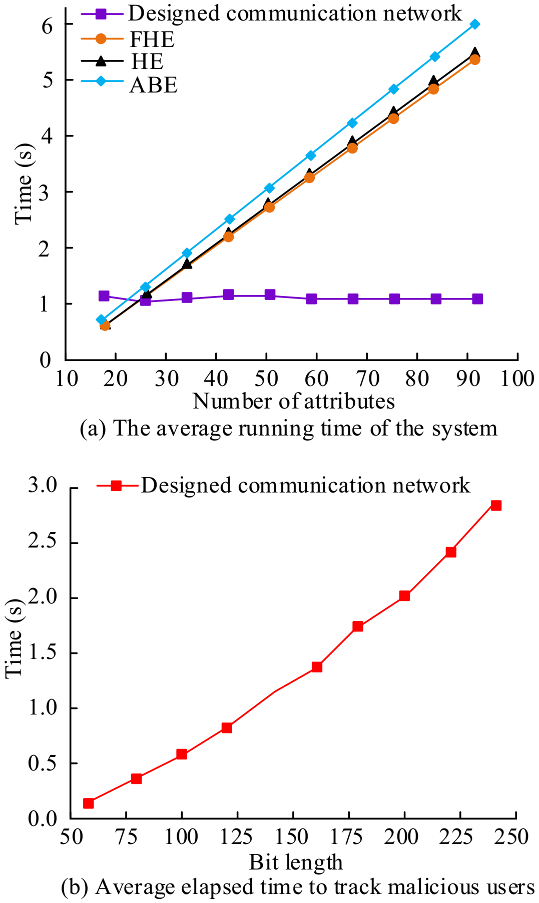
In Fig. 11a, as the amount of data increases, the running time of the designed communication network remains stable, with an average running time of 1.2 s. The running time of the other three networks increases linearly with the amount of data. In Fig. 11b, as the bit length increases, the running time of the designed communication network for tracking malicious users gradually increases. However, the network only takes 2 s to run at a bit length of 200. This suggests that the communication network created using network slicing and the BTA-AHEA is highly efficient. Network slicing can improve the flexibility and scalability of the network, while providing better isolation and security. Network slicing is suitable for the needs of different users and applications. Confidentiality can protect the integrity and confidentiality of data, preventing unauthorized access and leakage. Confidentiality is suitable for various data security scenarios. The initialization run time, private key generation run time, cipher-text generation run time, and decryption generation run time for FHE, HE, and ABE-based networks are calculated. Then, the efficiency of the designed communication network is further verified, as shown in Fig. 12.

In Fig. 12a, the average initialization time of the designed communication network is about 0.03 s. The average initialization time of the network designed on FHE is about 0.08 s. The average initialization time of the network designed on HE is about 0.20 s. The average initialization time of the network designed on the ABE is about 0.05 s. In Fig. 12b, the mean duration for generating private keys in the designed communication network is approximately 0.50 s. The mean duration of the network based on FHE is around 1.00s. The corresponding value for HE-based network is about 1.75 s, and for ABE-based network is about 1.25 s. In Fig. 12c, the average cipher-text generation time for the four network systems is approximately 0.30 s, 0.80 s, 1.10 s, and 0.90 s, respectively. In Fig. 12d, the average decryption generation time of the four network systems is approximately 0.25 s, 0.85 s, 1.65 s, and 0.60 s, respectively. This analysis shows that the average initialization time of the designed communication network is reduced by 0.05 s, 0.17 s, and 0.02 s compared to the other three networks, respectively. The average key generation time is reduced by 0.05 s, 1.25 s, and 0.75 s, respectively. The average cipher-text generation time is reduced by 0.50 s, 0.80 s, and 0.60 s, respectively. The average decryption generation time is reduced by 0.60 s, 1.40 s, and 0.35 s, respectively. This indicates that the designed communication network effectively improves the efficiency of system initialization private key generation, cipher-text generation, and decryption generation. Next, the bit error probability is introduced to verify the confidentiality effect of the designed communication network. The smaller the bit error probability value, the more ideal the effect. Figure 13 presents a comparison of the results with the other three networks.

In Fig. 13, the highest bit error probability value of the designed network is about 0.75, and the lowest is about 0.55. The probability of network bit error using FHE encryption has a maximum value of approximately 0.83 and a minimum value of approximately 0.45. The network encrypted with HE has a maximum probability of bit error of approximately 0.89 and a minimum of around 0.4. The probability of network bit error using ABE encryption has a maximum value of approximately 0.8 and a minimum value of approximately 0.4. The above results indicate that the designed communication network encryption based on network slicing and the BTA-AHEA has better performance. Finally, 7 types of simulation data are selected to test the privacy protection level

Method	Private key length	Cipher-text length	Computation cost	
			Private key generation	Cipher-text generation
BTA-AHEA	$2+\|S\|$	$3+\|h\|$	$(4+\|S\|)J_e+J_m$	$(3+\|h \times r\|)J_p+(1+\|h \times r\|)J_m$
FHE	$3+\|S\|$	$3+2\,\|h\|$	$(5+5\,\|S\|)J_e+(1+2\,\|S\|)J_m$	$(5+3\|h\|)J_e+(1+3\,\|h\|)J_m$
HE	$3+2\,\|S\|$	$4+3\,\|h\|$	$(5+7\,\|S\|)J_e+(3+\|S\|)J_m$	$(5+2\,\|h+r\|+6\,\|h\|)J_e$ $+(2+\|h+r\|+3\,\|h\|)J_m$
ABE	$2+3\|S\|$	$3+3\,\|h\|$	$(4+4\,\|S\|)J_e+\|S\|J_m$	$(4+3\,\|h\|)J_e+(2+3\,\|h\|)J_m$

Table 4. The cost of generating private keys and cipher-text for four algorithms.



(a) The average running time of the system

(b) Average elapsed time to track malicious users

Figure 11. The average running time of the system (**a**) and average elapsed time to track malicious users (**b**).

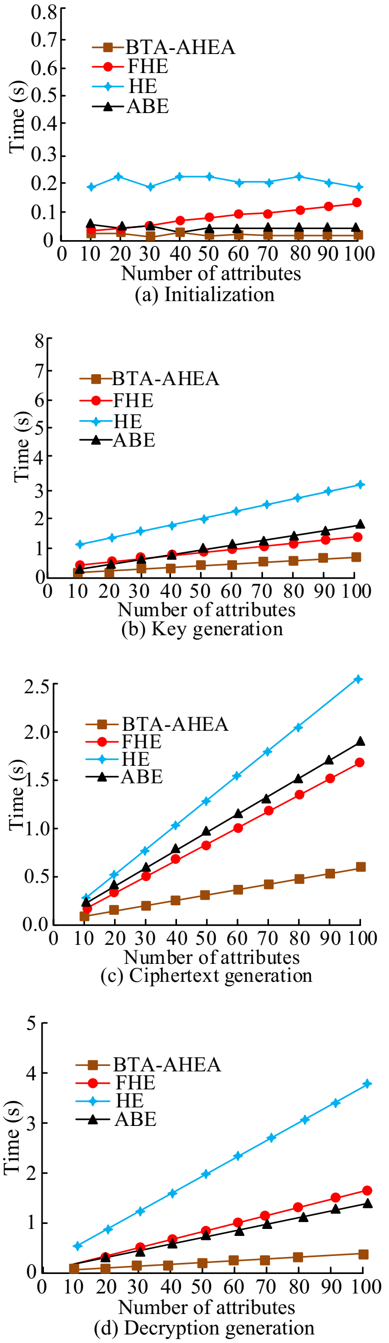
of the designed network during data transmission. BTA-AHEA is compared with networks based on FHE, HE, and ABE. The results are shown in Table 5.

In Table 5, for different types of simulation data, networks based on BTA-AHEA exhibit higher effectiveness in privacy protection. Compared to networks based on FHE, HE, and ABE, BTA-AHEA can effectively prevent the risks of information leakage and Trojan injection during data transmission, ensuring user privacy and security. This indicates that the BTA-AHEA has a high degree of privacy protection when designing the network, which is feasible and reliable for practical applications.

In summary, the designed communication network based on network slicing and BTA-AHEA performs outstandingly in terms of energy consumption, throughput, and runtime. Its computational cost in cipher-text generation is significantly lower than other networks, resulting in lower energy consumption. At the same time, the running time of the BTA-AHEA is stable, with an average running time of 1.2 s. The running time of the network gradually increases when processing large amounts of data, but the increase is small. When processing 200-bit lengths, the running time is only 2 s, demonstrating good throughput and runtime performance. In addition, the running time of this network in key steps such as private key generation, cipher-text generation, and decryption generation is significantly lower than other networks, further verifying its efficiency. From a practical perspective, these results indicate that the designed communication network can effectively improve the efficiency of system initialization private key generation, cipher-text generation, and decryption generation. It also demonstrates the ability to provide higher levels of privacy protection during data transmission, effectively preventing information leakage and Trojan injection risks, thereby ensuring user privacy. It should consider the compatibility between the proposed system and the existing network protocols to integrate this system into the existing network infrastructure. The designed algorithm can be implemented as a software module that can be integrated into the network system to provide the required functions. Furthermore, considering the energy consumption and throughput performance of the designed system, it is feasible to deploy it in environments that are crucial for energy efficiency and high data throughput.

Conclusion

To improve the security level of communication network privacy protection, this study first constructed a communication network system based on network slicing to deeply encrypt people's privacy. Then, the ABE was optimized using a chameleon hash function. The optimized algorithm was combined with the FHE to form the BTA-AHEA. A communication network encryption method is designed based on network slicing and BTA-AHEA.



(a) Initialization

(b) Key generation

(c) Ciphertext generation

(d) Decryption generation

Figure 12. The running time of each step in the decryption process.

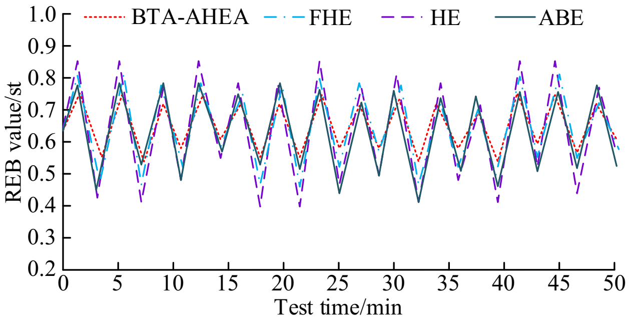


Figure 13. The running time of each step in the decryption process.

Simulation data type	FHE	HE	ABE	BTA-AHEA
1	No leakage	No leakage	No leakage	No leakage
2	There is no leakage, but there is a risk of theft due to Trojan injection	No leakage	No leakage	No leakage
3	No leakage	Complete leakage	Varying degrees of information theft	No leakage
4	Varying degrees of information theft	There is no leakage, but there is a risk of theft due to Trojan injection	No leakage	There is no leakage, but there is a risk of theft due to Trojan injection
5	No leakage	No leakage	No leakage	No leakage
6	User data uploaded, displaying unknown program	Varying degrees of information theft	There is no leakage, but there is a risk of theft due to Trojan injection	No leakage
7	No leakage	No leakage	No leakage	No leakage

Table 5. The degree of privacy protection in different networks.

The experimental results indicate that the designed network effectively prevented risks of information leakage and Trojan injection during data transmission. It was suitable for communication networks that require a high level of privacy protection. Additionally, this network exhibited significantly lower computational costs in cipher-text generation compared to other networks, making it suitable for energy-limited environments due to its low energy consumption characteristics. Furthermore, its stable runtime indicated good throughput and performance when processing large amounts of data, making it suitable for scenarios requiring high data throughput. Although there were limitations in tracking malicious users, the proposed network still possessed strong security, making it suitable for scenarios with high security requirements. Overall, the designed network demonstrates good adaptability and potential applications in handling different types of communication networks or scenarios. However, the designed network still has some limitations when facing the growth of large-scale network data traffic. The deficiencies include limitations in processing power, increased algorithm complexity, and increased management difficulty as the network size expands. Meanwhile, its adaptability in different network protocols and the ever-changing environment of the real world has not been verified. Therefore, future research will focus on improving the scalability of network design to cope with the increasing number of users and data traffic. Experiments will be conducted in different complex environments to ensure the practicality of the network.

Abbreviations

Noun	Abbreviation	Define
Attribute-based encryption	ABE	ABE is an encryption method that represents a user's identity as a set of multiple attributes, without requiring knowledge of the user's personal information
Fully homomorphic encryption	FHE	The FHE refers to the ability to simultaneously satisfy two homomorphic operations and perform multiple homomorphic addition and multiplication operations
Traceable attribute-based homomorphic encryption algorithm	BTA-AHEA	–
Network function virtualization	NFV	NFV can virtualize traditional proprietary hardware by installing a network on a virtual server
Software defined network	SDN	SDN can implement network virtualization
Distributed denial of service	DDOS	–

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 22 November 2023; Accepted: 6 August 2024
Published online: 20 August 2024

References

1. Chen, Z. Research on internet security situation awareness prediction technology based on improved RBF neural network algorithm. *JCCE* **1**(3), 103–108. https://doi.org/10.47852/bonviewJCCE149145205514 (2022).
2. Rasori, M., La Manna, M., Perazzo, P. & Dini, G. A survey on attribute-based encryption schemes suitable for the internet of things. *IEEE Internet Things* **11**(9), 8269–8290. https://doi.org/10.1109/JIOT.2022.3154039 (2022).
3. Chen, J., Li, K. & Philip, S. Y. Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain. *IEEE T. Intell. Transp. 1524-9050 (Print-ISSN)* **23**(8), 11633–11642. https://doi.org/10.1109/TITS.2021.3105682 (2021).
4. Li, D. *et al.* Ubiquitous intelligent federated learning privacy-preserving scheme under edge computing. *Future Gener. Comput. Syst.* **144**, 205–218. https://doi.org/10.1016/j.future.2023.03.010 (2023).
5. Xu, Z., He, D., Vijayakumar, P., Gupta, B. & Shen, J. Certificateless public auditing scheme with data privacy and dynamics in group user model of cloud-assisted medical WSNs. *IEEE J. Biomed. Health Inform.* https://doi.org/10.1109/JBHI.2021.3128775 (2021).

6. Yang, Y., He, D., Vijayakumar, P., Gupta, B. B. & Xie, Q. An efficient identity-based aggregate signcryption scheme with blockchain for IoT-enabled maritime transportation system. *IEEE Trans. Green Commun. Netw.* **6**(3), 1520–1531. https://doi.org/10.1109/TGCN.2022.3163596 (2022).
7. Zhao, Y. *et al.* Local differential privacy-based federated learning for internet of things. *IEEE Internet Things* **8**(11), 8836–8853. https://doi.org/10.1109/JIOT.2020.3037194 (2020).
8. Gope, P. & Sikdar, B. An efficient privacy-preserving authentication scheme for energy internet-based vehicle-to-grid communication. *IEEE T. Smart Grid* **10**(6), 6607–6618. https://doi.org/10.1109/TSG.2019.2908698 (2019).
9. Xing, L., Jia, X., Gao, J. & Wu, H. A location privacy protection algorithm based on double K-anonymity in the social internet of vehicles. *IEEE Commun. Lett.* **25**(10), 3199–3203. https://doi.org/10.1109/LCOMM.2021.3072671 (2021).
10. Shouran, Z., Ashari, A. & Priyambodo, T. Internet of things (IoT) of smart home: Privacy and security. *IJCA* **182**(39), 3–8. https://doi.org/10.5120/ijca2019918450 (2019).
11. Lu, Y., Huang, X., Dai, Y., Maharjan, S. & Zhang, Y. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE T. Ind. Inform.* **16**(6), 4177–4186. https://doi.org/10.1109/TII.2019.2942190 (2019).
12. Liu, Y., James, J. Q., Kang, J., Niyato, D. & Zhang, S. Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet Things* **7**(8), 7751–7763. https://doi.org/10.1109/JIOT.2020.2991401 (2020).
13. Sun, Y., Liu, J., Wang, J., Cao, Y. & Kato, N. When machine learning meets privacy in 6G: A survey. *IEEE Commun. Surv. Tut.* **22**(4), 2694–2724. https://doi.org/10.1109/COMST.2020.3011561 (2020).
14. Gai, K., Wu, Y., Zh, L., Xu, L. & Zhang, Y. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet Things* **6**(5), 7992–8004. https://doi.org/10.1109/JIOT.2019.2904303 (2019).
15. Yao, D. *et al.* Energy theft detection with energy privacy preservation in the smart grid. *IEEE Internet Things* **6**(5), 7659–7669. https://doi.org/10.1109/JIOT.2019.2903312 (2019).
16. Qu, Y. *et al.* Decentralized privacy using blockchain-enabled federated learning in fog computing. *IEEE Internet Things* **7**(6), 5171–5183. https://doi.org/10.1109/JIOT.2020.2977383 (2020).
17. Lu, Y., Huang, X., Zhang, K., Maharjan, S. & Zhang, Y. Communication-efficient federated learning and permissioned blockchain for digital twin edge networks. *IEEE Internet Things* **8**(4), 2276–2288. https://doi.org/10.1109/JIOT.2020.3015772 (2020).
18. Zhang, K., Zhu, Y., Maharjan, S. & Yhang, Z. Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things. *IEEE Netw.* **33**(5), 12–19. https://doi.org/10.1109/MNET.001.1800526 (2019).
19. Wijethilaka, S. & Liyanage, M. Survey on network slicing for Internet of Things realization in 5G networks. *IEEE Commun. Surv. Tutorials* **23**(2), 957–994. https://doi.org/10.1109/COMST.2021.3067807 (2021).
20. Wu, W. *et al.* AI-native network slicing for 6G networks. *IEEE Wirel. Commun.* **29**(1), 96–103. https://doi.org/10.1109/MWC.001.2100338 (2022).
21. Siriwardhana, Y., Porambage, P., Liyanage, M. & Ylianttila, M. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Commun. Surv. Tut.* **23**(2), 1160–1192. https://doi.org/10.1109/COMST.2021.3061981 (2021).
22. Ren, J., Zhang, D., He, S., Zhang, Y. & Li, T. A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet. *CSUR* **52**(6), 1–36. https://doi.org/10.1145/3362031 (2019).
23. Wang, S. *et al.* Dynamic service migration in mobile edge computing based on Markov decision process. *IEEE ACM Trans. Netw.* **27**(3), 1272–1288. https://doi.org/10.1109/TNET.2019.2916577 (2019).
24. Xu, M. *et al.* PDMA: Probabilistic service migration approach for delay-aware and mobility-aware mobile edge computing. *Softw. Pract. Exp.* **55**(2), 394–414. https://doi.org/10.1002/spe.3014 (2021).
25. Ning, Z., Huang, J., Wang, X., Rodrigues, J. J. & Guo, L. Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling. *IEEE Netw.* **33**(5), 198–205. https://doi.org/10.1109/MNET.2019.1800309 (2019).
26. Yuan, Q. *et al.* A joint service migration and mobility optimization approach for vehicular edge computing. *IEEE Trans. Veh.* **69**(8), 9041–9052. https://doi.org/10.1109/TVT.2020.2999617 (2020).
27. Hidayat, I., Ali, M. Z. & Arshad, A. Machine learning-based intrusion detection system: An experimental comparison. *JCCE* **2**(2), 88–97. https://doi.org/10.47852/bonviewJCCE2202270 (2022).
28. Li, F., Liu, K., Zhang, L., Huang, S. & Wu, Q. Ehrchain: A blockchain-based ehr system using attribute-based and homomorphic cryptosystem. *IEEE Trans. Serv. Comput.* **15**(5), 2755–2765. https://doi.org/10.1109/TSC.2021.3078119 (2021).
29. Wang, F. *et al.* An experimental investigation into the hash functions used in blockchains. *IEEE Trans. Eng. Manag.* **67**(4), 1404–1424. https://doi.org/10.1109/TEM.2019.2932202 (2019).
30. Thanalakshmi, P. & Anitha, R. A quantum resistant chameleon hashing and signature scheme. *IETE J. Res.* **68**(3), 2271–2282. https://doi.org/10.1080/03772063.2019.1698323 (2019).
31. Gupta, R., Gupta, I., Saxena, D. & Singh, A. K. A differential approach and deep neural network based data privacy-preserving model in cloud environment. *J. Amb. Intell. Hum. Comp.* **14**(5), 4659–4674. https://doi.org/10.1007/s12652-022-04367-x (2023).
32. Wang, K., Wang, X. & Lu, X. POI recommendation method using LSTM-attention in LBSN considering privacy protection. *Complex Intell. Syst.* **9**(3), 2801–2812. https://doi.org/10.1007/s40747-021-00440-8 (2023).
33. Mishra, A. K. & Mohapatra, Y. Hybrid blockchain based medical data sharing with the optimized CP-ABE for e-Health systems. *Int. J. Inf. Tech.* **16**(1), 121–130. https://doi.org/10.1007/s41870-023-01625-9 (2024).
34. Siva Swaroopa Rani, A. *et al.* Hybrid cipher-text key policy attribute-based encryption (HCKP-ABE): The performance analysis and scalability in virtual machines. *IJCDS* **15**(1), 1–9. https://doi.org/10.12785/ijcds/xxxxxx (2024).

Author contributions

Wei Wang processed the numerical attribute linear programming of communication big data, and the mutual information feature quantity of communication big data numerical attribute was extracted by the cloud extended distributed feature fitting method. Wei Wang and Rong Liu Combined with fuzzy C-means clustering and linear regression analysis, the statistical analysis of big data numerical attribute feature information was carried out, and the associated attribute sample set of communication big data numerical attribute cloud grid distribution was constructed. Wei Wang and Silin Cheng did the experiments, recorded data, and created manuscripts. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to W.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.