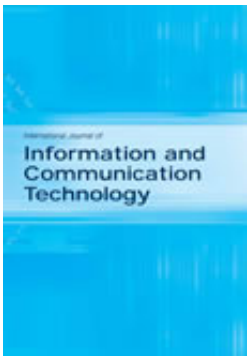
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Fine-grained data cross-domain access control policy based on ciphertext policy attribute encryption

Ying Xue, Gang Wang, Qian Zhang

Article History:

Received:	10 February 2025
Last revised:	19 February 2025
Accepted:	19 February 2025
Published online:	15 April 2025

Fine-grained data cross-domain access control policy based on ciphertext policy attribute encryption

Ying Xue*, Gang Wang and Qian Zhang

Department of Information Technology,
Shaanxi Police College,
Xi'An 710021, China
Email: jcx202411@163.com
Email: wangzixi1976@sohu.com
Email: zq_bfwl1010@163.com
*Corresponding author

Abstract: As the big data technique rapidly develops, the demand for inter-agency cross-domain data sharing is growing, but there is a risk of unauthorised access in cross-domain data sharing. To this end, this paper first improves the ciphertext policy attribute-based encryption (MCACP-ABE), which achieves fine-grained protection of cross-domain data by authorising cross-domain third parties and attribute authority centres, and introduces the accountability tracking module. On this basis, fine-grained data cross-domain access control (AC) policies are designed. The policy designs a cross-domain AC structure based on MCACP-ABE, which realises fine-grained data access protection through a cross-domain negotiation component, a rule mapping component, and a cross-domain encryption component. The security analysis and simulation outcome imply that the offered policy not only satisfies indistinguishable security under chosen ciphertext attack (IND-CCA) but also has high cross-domain communication efficiency, which improves the security and usability of data cross-domain access.

Keywords: cross-domain access control; ciphertext policy attribute-based encryption; accountability tracking; cross-domain negotiation; rule mapping.

Reference to this paper should be made as follows: Xue, Y., Wang, G. and Zhang, Q. (2025) 'Fine-grained data cross-domain access control policy based on ciphertext policy attribute encryption', *Int. J. Information and Communication Technology*, Vol. 26, No. 7, pp.63–78.

Biographical notes: Ying Xue received her Master's degree from Xi'an Jiaotong University in 2007. She is currently an Associate Professor at the Shaanxi Police College. Her research interests include network security technology and big data applications.

Gang Wang received his Master's degree from Xi'an University of Technology in 2014. He is currently a Professor at the Shaanxi Police College. His research interests are included network security and information system architecture design.

Qian Zhang received her PhD degree from the Rocket Force University of Engineering in 2014. She is currently an Associate Professor at the Shaanxi Police College. Her research interests include cybersecurity and law enforcement, as well as network intelligence acquisition and analysis.

1 Introduction

As the cloud computing and big data technology rapidly growing, cross-domain data sharing among different organisations and institutions has increased the value of data utilisation (Singh et al., 2021). However, the risk of unauthorised access in cross-domain data sharing increases with it. The access control (AC) mechanism constitutes one of the key technologies in the realm of information security, which aims to protect the three main security attributes of resources (Salehi et al., 2023). Faced with the rapid increase in the demand for data sharing, the dramatic increase in business transactions between enterprises and organisations under different management domains, and the need to exchange confidential data information, in this scenario requires a data cross-domain AC mechanism to ensure the security of data resources. At present, various organisations have established AC systems for their own domains. On the basis of having their own management domains, realising secure data interaction between different AC systems is a hot issue in realising data cross-domain AC (Sun and Fang, 2009). For such scenarios above, cross-domain AC mechanism will be more effective and practical than traditional single-domain AC methods, and there is an urgent need to design an efficient cross-domain AC policy to protect cross-domain sensitive data resources, so as to guarantee the safety of cross-domain data (Yang and Wang, 2016).

Traditional AC strategies include the autonomous AC model (Qi and Zheng, 2019) and the mandatory AC model (Jiang, 2017). Xue et al. (2022a) introduced the concepts of subject, object, and access privilege for the AC mechanism, which concretised the abstract AC mechanism. Cruz et al. (2018) proposed the role-based AC model (RBAC), which assigns subjects to roles so that the subjects have access rights to the corresponding roles. Liu et al. (2020) proposed the attribute-based AC model (ABAC). The continuous innovation of these models has led to the use of AC in various fields such as IoT, big data, and cloud computing. Karimi et al. (2021) proposed an optimised ABAC model that secures resource sharing between potentially untrustworthy tenants with attributes, roles, and tasks, and achieves secure access by supporting different access rights for the same user in the same session.

The traditional AC model mainly solves the service AC in a single domain, but the systems in different domains need to realise communication interconnection and data sharing, which requires a more flexible and intelligent cross-domain AC model to guarantee the safety and high efficiency of data exchange between systems. Zhang and Liu (2020) proposed a multi-attribute based cross-domain trusted AC model. The model combines heuristic algorithms to realise dynamic multi-dimensional trust evaluation. Chen et al. (2021) realised cross-domain AC for IoT by means of identity authentication. Xue et al. (2022b) proposed to formulate cross-domain ACL method by identity set, but the identity information is easy to leak. The above cross-domain authentication methods based on trust level and identity are coarse-grained in the protection of data resources and have some potential security risks. Meanwhile, the heterogeneity between different domains in these approaches reduces the reliability of trust evaluation and the security of inter-domain data exchanging.

To improve the security of inter-domain data AC, the researcher extends the single-domain AC method by rule mapping approach to protect the security of inter-domain data. Abdelfattah et al. (2022) proposed an inter-domain role mapping mechanism which maps the roles involved in inter-domain participation by mapping the roles in the source domain in the target domain and granting the user the appropriate

privileges to access the resources through the roles the user has in the target domain. Zhao et al. (2022) proposed a secondary assignment of user roles so that users are assigned permissions to achieve relatively fine-grained authorisation for cross-domain users, but suffers from poor scalability. To further optimise the access policy, Banerjee et al. (2020) proposed an approach integrating symmetric encryption and ciphertext policy attribute-based encryption (CP-ABE), which uses a trusted agent to store the key, enhance the system's effectiveness, and achieve cross-domain AC, but suffers from cross-domain inefficiency. Das and Namasudra (2023) proposed CP-ABE with multiple permissions and attribute revocation to realise cross-domain collaboration, but there are still problems such as sensitive data leakage due to mapping mechanism.

In summary, existing cross-domain AC methods for data have the problems of complex access policies, difficult cross-domain responsibility traceability and low confidentiality. In order to solve the above problems, this paper optimises the existing cross-domain AC methods through the ideas of multi-authorisation traceability and simplified access policy. Firstly, CP-ABE is optimised based on the idea of multi-authority traceability (MCACP-ABE), which ensures the effectiveness and availability of cross-domain encryption when there are multiple attribute authority centres in the process of accessing resources through the authentication and authorisation of cross-domain third parties and attribute authority centres. Meanwhile, an accountability tracking module is introduced into the algorithm to trace malicious users across domains, realising fine-grained protection of cross-domain data. On this basis, the legitimacy of parties sharing data across domains is ensured by authenticating the identities of data owners and data users across domains, and the mapping of attribute names and attribute value spaces within each domain solves the problem of heterogeneity of cross-domain access management. In addition, a cross-domain AC structure based on MCACP-ABE is designed to ensure fine-grained protection of data access. Finally, it is proved in the standard model that the designed policy satisfies indistinguishable security under chosen ciphertext attack (IND-CCA). And through experimental simulation, the offered strategy not just reduces the computational overhead but also enhances the cross-domain communication efficiency compared with the existing strategies, realising secure and efficient data cross-domain AC.

2 Relevant theoretical foundations

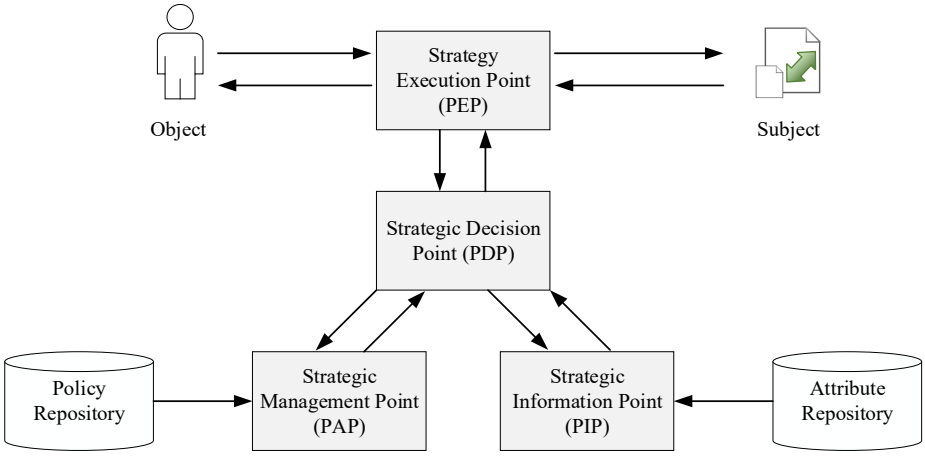
2.1 Attribute-based AC model

ABAC is an AC method that authorises or denies a subject's request to operate on an object based on the subject-object attributes, environmental situations, and a collection of strategies developed based on these attributes and situations (Hu et al., 2015). Compared with the traditional AC model, ABAC is able to achieve fine-grained AC based on multiple attributes and dynamically adjusts the access policy based on the entity's attributes, thus adapting to different AC needs. The authorisation block diagram of ABAC is shown in Figure 1.

The architecture encompasses a policy decision point (PDP), a policy enforcement point (PEP), a policy information point (PIP), and a policy administration point (PAP). The PDP determines the outcome of the access decision by assessing the AC strategy. The PEP responds to the access demand from the AC subject and executes the access

decision given by the PDP. The PIP evaluates the retrieval source of the required data and provides the message required through the PDP to compute the access decision. The PAP creates and manages the AC policy by using an external interface and stores the AC policy in the policy repository.

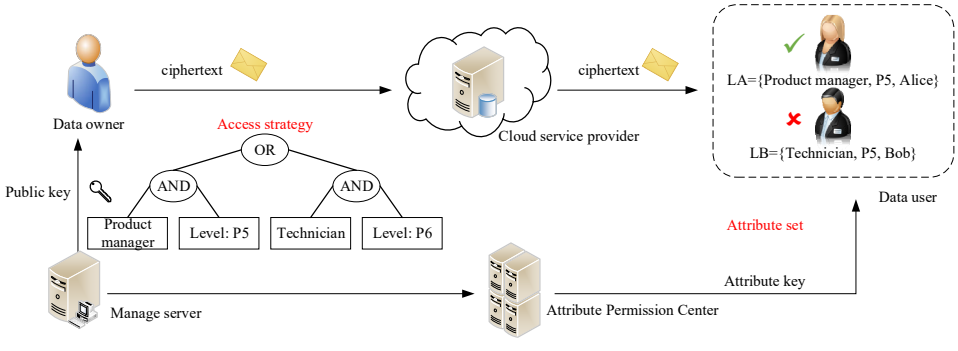
Figure 1 The authorisation block diagram of ABAC (see online version for colours)



2.2 Ciphertext policy attribute-based encryption

CP-ABE is an attribute-based encryption method, which is more appropriate for one-to-many message distribution and scenarios requiring fine-grained AC than the key-policy-based attribute-based encryption (KP-ABE) algorithm due to its property of embedding the access strategy into the ciphertext. CP-ABE distributes attribute keys relied on the accessing user's attributes, and the user is only able to decrypt the ciphertext if the user's attributes meet the requirements of the AC strategy (Nguyen et al., 2018). The system model of CP-ABE is shown in Figure 2, which mainly consists of attribute authority centre (AA), management server (MS), data owner (DO), cloud service provider (CSP), and data user (DC).

Figure 2 The system model of CP-ABE (see online version for colours)



Define an AC strategy tree τ with root node R and an internal subtree τ_r, whose subtree root nodes are r. If a set of attributes $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$ satisfies τ_r, it is denoted by $\tau_r(\omega) = 1$. Verification for each attribute in the attribute group Perform recursion. Based on the set AC policy, CP-ABE is realised as follows.

1 *Setup*(): select the tuple G_0 of order p, generate the element g, and randomly select α, $\beta \in Z_p$. Then compute the output public key *pk* and master key *msk*.

$$\begin{cases} pk = \left(G_0, g, h = g^{\beta}, e(g, g)^{\alpha}\right) \\ msk = \left(\beta, g^{\alpha}\right) \end{cases} \tag{1}$$

2 *Encrypt*(*pk*, *m*, τ): first select a polynomial $q(x)$ of order d_x (and $d_x = k_x - 1$) for every node x in τ, starting from the root node R in a hierarchical fashion. Beginning with R, randomly select $s \in Z_p$ and set $q_R(0) = s$. For other nodes x, set $q_x(0) = q_{parent(x)}(index(x))$. If Y is the set of all leaf nodes in τ, the ciphertext CT is as follows.

$$CT = \left(\tau, C = Me(g, g)^{\alpha z}, C = h^s, \forall y \in Y : C_y = g^{q_y(0)}, C_y' = H\left(att(y)\right)^{q_y(0)}\right) \tag{2}$$

3 *KeyGen*(*msk*, *S*, *pk*): randomly select $r \in Z_p$, select ergodic numbers $r_j \in Z_p$ for every attribute $j \in S$, input *pk* and *msk*, a group of attributes *S*, compute and output the user private key *SK*.

$$SK = \left(D = g^{(r+\alpha)/\beta}, \forall j \in S : D_j = g^r \cdot H(i)^{r_j}, D_j' = g^{r_j}\right) \tag{3}$$

4 *Decrypt*(*CT*, *SK*): first define the recursive function *DecryptNode*(*CT*, *SK*, *x*), with *CT*, *SK* and τ as inputs. If x is a leaf node and set the attribute $I = att(x)$, then having equation (4).

$$\begin{aligned} DecryptNode(CT, SK, x) &= \frac{e\left(D_j, C_j\right)}{e\left(D_j', C_j'\right)} = \frac{e\left(g^r \cdot H(j)^{r_j}, h^{q_j(0)}\right)}{e\left(g^{r_i}, H(j)^{q_x(0)}\right)} \\ &= e(g, g)^{rq_x(0)}, i \in S \end{aligned} \tag{4}$$

If x is a non-leaf node, all the child nodes belonging to x are denoted by z. If S_x is an aggregation of k_x nodes, then $F_x = e(g, g)^{r_i q_j(0)}$. When $\varphi_u \in T$, then the root node R of τ is called and $A = e(g, g)^{r_i s}$ is computed, then the encrypted data can be decrypted.

$$\frac{\tilde{C}}{e(C, D)/A} = \frac{m \cdot e(g, g)^{\alpha s}}{e\left((g^{\beta})^s, g^{(\alpha+r_i)/\beta}\right)/e(g, g)^{r_i s}} = m \tag{5}$$

3 Cross-domain based CP-ABE algorithm design for multi-authority accountability

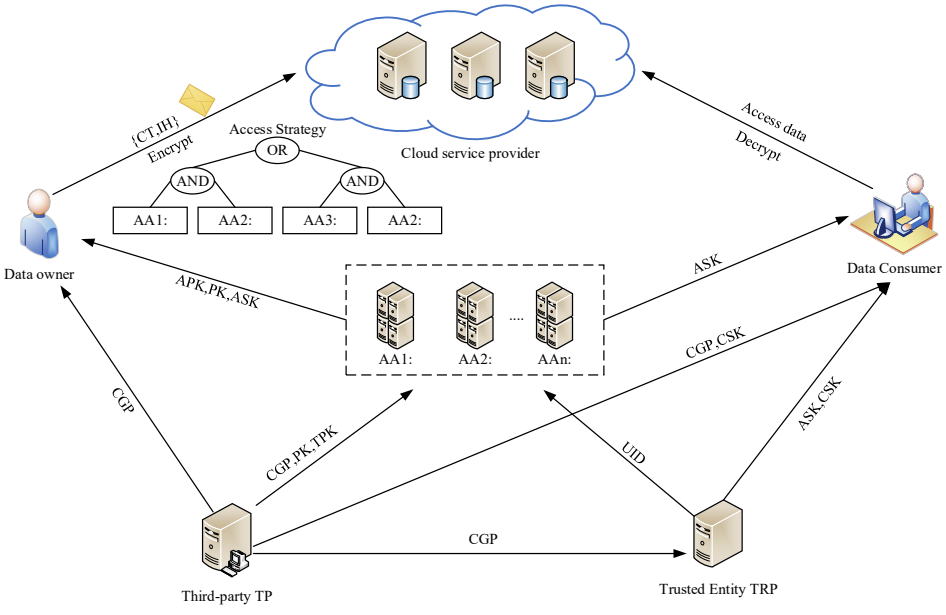
3.1 Improved CP-ABE algorithm architecture

Intending to the issues of constrained cross-domain encryption and decryption authentication of CP-ABE algorithm and the difficulty of cross-domain accountability

traceability, we propose the CP-ABE method based on multi-authority traceability (MCACP-ABE). The method ensures the validity and usability of cross-domain encryption through the authentication and authorisation of cross-domain third parties and attributes authority centres to meet the cross-domain access requirements of multi-attribute authority. In addition, the method improves the security of cross-domain data resources through accountability tracking and cross-domain traceability.

The structure of MCACP-ABE algorithm is implied in Figure 3. The algorithm chiefly contains data owner (DO), data user (DC), cloud service provider (CSP), multi-attribute authority centres (AAs), third party (TP), and trusted entity (TRP). The TP, CSP, and DC are semi-trusted, and the AAs and TRP are fully trusted.

Figure 3 The system model of MCACP-ABE (see online version for colours)

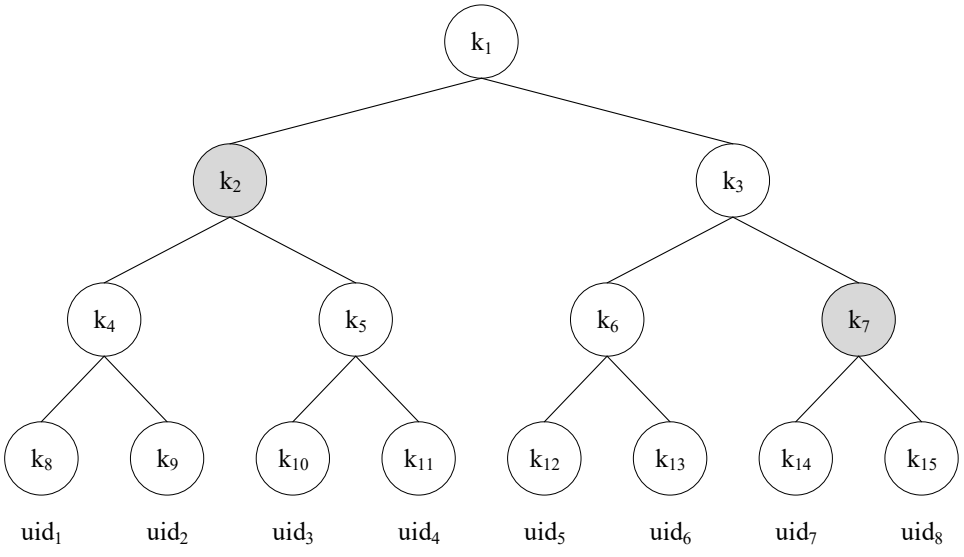


First the TP generates the global parameters (GP) and the central key (CSK) through the system setup and sends the GP to the participating entities. Next, the AAs provide the attribute public key (APK) and attribute master key (ASK) to the DOs and the attribute master key (ASK) to the DCs, and the DOs utilise the predefined access strategy to encrypt the message and then transmit the resultant ciphertext (CT) alongside the message header (IH) to the CSP. The DC then generates a key from the ASK key set gained by the AAs and performs a decryption function if its attributes meet the criteria specified in the CT's access strategy. The input to the accountability procedure is the suspected decryption key, APK, and GP, and the output is the u_{id} of the malicious user as, and it is reported to AAs.

The MCACP-ABE access policy utilises the linear secret sharing scheme (LSSS) (Jia et al., 2022). The AC tree is represented as LSSS and new attributes are added to meet the system requirements by repositioning the nodes and attributes in the tree as shown in Figure 4. Cross-domain AC trees are primarily designed by DOs to match sets of attributes with access rights to control access to data by visitors. Meanwhile, the

cross-domain attribute certificate represents the visitor's identity information and attribute set information, and obtains the attribute key by interacting with the AAs.

Figure 4 The designed AC tree



3.2 Multi-permission cross-domain CP-ABE algorithm

MCACP-ABE includes four algorithms: cross-domain system setup, multi-privilege attribute key, multi-privilege encryption and decryption, and cross-domain accountability. Firstly, input the program identifier and public parameters, and AAs generates the corresponding public and private keys. Secondly, the attribute key generation algorithm generates the ASK by inputting the public parameters, the master key and the public and private keys generated by the AAs. Then the encryption approach encrypts the information against the access strategy to generate a ciphertext (CT), and the decryption algorithm generates a decrypted message that satisfies the access policy by using the ASK, CSK, and common parameter (CGP). Finally, by obtaining the decryption key, public key and CGP of the DC, we can output the suspicious user information.

1 Cross-domain system settings include global settings and authorisation settings. The global setting $CD\_Setup(\alpha) \rightarrow (CGP)$, the input to the algorithm is the security parameter α, and the output is $CGP = \{G, G_1, p, e, g, U, R, Sign, L\}$, where G and G_1 are cyclic groups of prime order p, e is a bilinear mapping of $G \times G \rightarrow G_1$, and g is a generator of G. The function U maps the user identification u_{id} to an instance in G. The function R assigns an attribute to an instance in G, and signs it with the secure signature system $Sign$. The elements of the construction are the set of attributes A and the set of permissions P, and for each attribute $attr_i \in A$, there is a permission $per_i \in P$ that maps an input $attr_i$ to the corresponding per_i.

2 The authorisation setup consists of two algorithms: $CTP\_Setup(CGP) \rightarrow (TPK, TMK)$, a key generation algorithm with $Sign$ is run by the CTP to generate the master key TMK and the public key TPK. $AAs\_Setup(CGP, per_i, attr_i, A) \rightarrow$

$(APK_{per_i}, AMK_{per_i})$, each *AAs* manages its collection of attributes *A*. For each $attr_i$, each *AA* takes as input the *CGP*, which randomly chooses $\eta_{per}, \zeta_{per_i} \in Z_P^*$ to compute $h_{per_i} = g^{\eta_{per_i}}$ and $C_{per_i} = g^{\zeta per_i}$. It also randomly selects $\varphi_{per_i}, \rho_{per_i} \in Z_P^*$. The final extracted APK_{per_i} and AMK_{per_i} are as follows.

$$APK_{per_i} = \left\{ e(g,g)^{\varphi_{per_i}}, g^{\rho_{per_i}}, h_{per_i}, C_{per_i} \right\} \tag{6}$$

$$AMK_{per_i} = \left\{ \eta_{per_i}, \zeta_{per_i}, \varphi_{per_i}, \rho_{per_i} \right\} \tag{7}$$

3 Multi-authority attribute key, generated by algorithm *CTP_KeyGen* and algorithm *AAs_KeyGen*. By running *CTP_KeyGen*, the CTP sends the identity-related key to the DC. AAs then run *AAs_KeyGen* and send the DC the attribute-related key. CTP main AAs and DCs, and issue corresponding ids for AAs and DCs, and connect attribute keys with the same per_i into an attribute key string through u_{id} to realise CP-ABE with multiple privileges.

The inputs to *CTP_KeyGen* are *CGP*, u_{id}, and $attr^{u_{id}}$, and the output is the private key $CSK_{u_{id}}$ corresponding to the domain attribute set, the private key $TPPK_{u_{id}}$ and the public key $TPSK_{u_{id}}$ related to u_{id}. For each *DC*, *CTP* randomly selects $\gamma, \varepsilon \in Z_p$, $\mu_i, v_i \in Z_P^*$, and calculates the private key as in equation (8), and signed with *TMK*, for each attribute compute $aek_i = h_{per_i}^{v_i}$, then $TPPK_{u_{id}} = \{TMK, u_{id}, sign_i, aek_i, attr_i^{u_{id}}\}$, compute $CSK_{u_{id}} = g^{\mu_i}$ and send the generated message to DC.

$$TPSK_{u_{id}} = \left\{ g^{(\varepsilon+\mu_i)/v}, g^{\mu_i} \times R\left(attr_i^{u_{id}}\right)^{v_i}, g^{v_i} \right\} \tag{8}$$

$$sign_i = \sum_{sign}^{TMK} \left(TMK, TPSK_{u_{id}}, u_{id}\right) \tag{9}$$

The input to *AAs_KeyGen* is $attr^{u_{id}}$, *CGP*, AMK_{per_i}, $TPPK_{u_{id}}$, *TPK*, and the output is the attribute key $ASK_{attr}^{u_{id}}$ of *AAs*. It associates user attributes and users to the attribute tree Att_T. Att_T is used to distribute attribute group keys to DCs. For each *DC*, *AAs* generate a set of *AEK*, which are computed from the path from the root node of Att_T to the leaf nodes.

4 Multi-authority encryption and decryption contains *CD_Encrypt* and *CD_Decrypt*. The inputs to *CD_Encrypt* are message *M*, access policy *N*, public parameter APK_{per_s}, and *CGP*. *N* is denoted by $(\overset{o}{A}, \delta)$ and $\overset{o}{A}$ is the LSSS matrix containing 1 and *k* columns. Choose a random vector $\vec{v} = (s, r_2, ..., r_n) \in Z_p^n$, and for each row of $\overset{o}{A}$, randomly choose $l_x \in Z_p$, calculated as follows.

$$\begin{aligned} &C_0 = M \cdot e(g,g)^{\varepsilon S}; \quad C_1 = g^{\gamma S}; \quad C_x = g^{\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}{}^{l_x}}; \\ &C_x' = g^{l_x}; \quad C_x' = R\left(attr_{(x)}^{u_{id}}\right)^{\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}{}^{l_x}} \end{aligned} \tag{10}$$

where $n \in R_T$, R_T is the set of indexes of attribute authority centres (AA) in N. The ciphertext generated by DO is $CT' = \{N, C_0, C_1, (C_x, C_x)_{x \in I}\}$. DO sends the CT to AA and performs a re-encryption operation. For each attribute appearing in the AC tree embedded in the CT, the AA selects a random value and computes the following equation.

$$C_x'' = g^{\ell_x} \cdot g^{\xi_m} \tag{11}$$

$$C_x'' = R\left(attr_{(x)}^{u_{id}}\right)^{\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}{}^{\ell_x}} \cdot g^{\xi m} \tag{12}$$

where *CD_Decrypt* performs a decryption operation on the CT via $ASK_{attr}^{u_{id}}$ and $TPSK_{u_{id}}$ and successfully obtains M if the collection of attributes meets the access strategy requirements in the CT of the embedding text, otherwise, output $\perp$. DC execute the recursive decryption function $Decrypt(CT, TPSK_{u_{id}}, ASK_{attr}^{u_{id}}, x)$. If x is a leaf node, then let $attr_i^{uid} = attr_x^{uid}$, otherwise $attr_i^{uid} \in attr_x^{uid}$. The decryption function is defined as follows.

$$\begin{aligned} &Decrypt\left(CT, TPSK_{u_{id}}, ASK_{attr}^{u_{id}}, x\right) \\ &= \frac{e(g,g)^{\mu_i \cdot \left(\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}^{\ell_x}\right)} \cdot e\left(R\left(attr_i^{u_{id}}\right), g\right)^{v_i \cdot \left(\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}^{\ell_x}\right)} .e(g,g)^{\xi_m \cdot v_i}}{e\left(R\left(attr_i^{u_{id}}\right), g\right)^{v_i \cdot \left(\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}^{\ell_x}\right)} .e(g,g)^{\xi_m \cdot v_i}} \\ &= e(g,g)^{\mu_i \cdot \left(\gamma \cdot \overset{o}{A}_x \vec{v} + h_{\psi_x}^{\ell_x}\right)} = F_x \end{aligned} \tag{13}$$

If $attr_i^{uid} \notin attr_x^{uid}$, then decrypt function $Decrypt(CT, TPSK_{u_{id}}, ASK_{attr}^{u_{id}}, x) = \perp$. If the attributes of the DC match N, then $F_R = Decrypt(CT, TPSK_{u_{id}}, ASK_{attr}^{u_{id}}, x)$ is returned. where R is the root node of the tree. The DC obtains the plaintext by the following calculation.

$$\frac{M \cdot e(g,g)^{\varepsilon \cdot S} \cdot e(g,g)^{\mu_i \cdot S}}{e\left(g^{(\varepsilon + \mu_i)/\gamma}, g^{\gamma S}\right)} = \frac{M \cdot e(g,g)^{(\varepsilon + \mu_i)s}}{e(g,g)^{(\varepsilon + \mu_i)s}} = M \tag{14}$$

5 Data cross-domain accountability traceability is supported by assigning and identifying u_{id}, tracking their access behaviour to the data, and using u_{id} together with attributes to generate keys. When u_{id} detects a malicious user that may have exposed its decryption key, the malicious user can be located.

The inputs to *CD_Trace* are $ASK_{attr}^{u_{id}}$, $TPPK_{u_{id}}$, $TPSK_{u_{id}}$ and CGP. First verify that $ASK_{attr}^{u_{id}}$ is well formatted, if $ASK_{attr}^{u_{id}}$ is error free, the algorithm outputs u_{id}, which associates $ASK_{attr}^{u_{id}}$ and u_{id}, otherwise it outputs $\perp$. If $ASK_{attr}^{u_{id}}$ fails the key integrity check, output $\perp$, indicating that no trace is required. If aek_i is in Att_T, output u_{id}. The decryption key that passes the completeness check can be used to decrypt the ciphertext with a satisfying policy as shown in Equation (16).

$$e\left(aek_i, g\right) = e\left(k_i, g^{v_i}\right) \tag{15}$$

$$\begin{aligned} e\left(TPSK_{u_{id}}, g\right) &= e\left(g^{(\varepsilon+\mu_i)/v} \cdot g^{\mu_i} \cdot R\left(attr_i^{\mu_{id}}\right)^{v_i} \cdot g^{v_i}, g\right) \\ &= e(g, g)^{(\varepsilon+\mu_i)/v} \cdot e\left(R\left(attr_i^{u_{id}}\right), g^{v_i}\right) \cdot e\left(g, h_{per_i}^{v_i}\right) \\ &\quad \cdot e\left(g^{\eta_{per_i} v_i / k_i attr_i^{uid}} \cdot k_i, g^{\mu_i}\right) \end{aligned} \tag{16}$$

3.3 Security analysis

The MCACP-ABE algorithm is analysed for security and the system is proved to satisfy IND-CCA security through a game between attacker *A* and challenger *C*. First the system setup is performed, *C* performs the initialisation setup algorithm and *A* gets the CGP from *C*. Assuming that *A* knows the master key of each attacked *AA*, *A* selects the attribute set $C_{attr_i} \in A$ and the corresponding public key among the attacked *AA*. The set $S_{attr_i} \in A$ of attributes of the *AA* of security, where $S_{attr_i} \cap C_{attr_i} \neq \varnothing$. *A* requests the key associated with $attr^{u_{id}}$ and the conversion key. *A* then sends two messages of the same length M_0, M_1 and a set of access policies, for each of which the attribute information in the AC policy is managed by *AA*. *A* queries the ciphertext of any M_0 and M_1. *C* randomly selects $b \in \{0, 1\}$, generates *AA* public key corresponding to $S_{attr_i} \in A$, and obtains the key set by key generation algorithm. Then, the ciphertext of M_b verifying *A* is encrypted using a cryptographic algorithm. Finally *A* guesses $b' \in \{0, 1\}$. If *A* has a minimal and insignificant chance of accurately selecting *b* in the aforementioned security game, then the MCACP-ABE model is IND-CCA secure and the advantage defined as $|P_r[b' = b] - 0.5|$ is that of correctly guessing.

4 Design of fine-grained data cross-domain AC policy based on improved CP-ABE

Based on the above MCACP-ABE model, this paper designs a fine-grained data inter-domain AC policy, as shown in Figure 5. The policy consists of three modules: cross-domain negotiation, rule mapping and cross-domain encryption/decryption. The cross-domain negotiation module is responsible for authenticating the identity of cross-domain DO and DC, and ensuring the legitimacy of parties sharing data across domains. The rule mapping module is responsible for mapping the attribute name and attribute value space within each domain, and the cross-domain encryption module is responsible for encrypting data with multi-privilege accountability, checking for and locating the data accessor that has leaked the key.

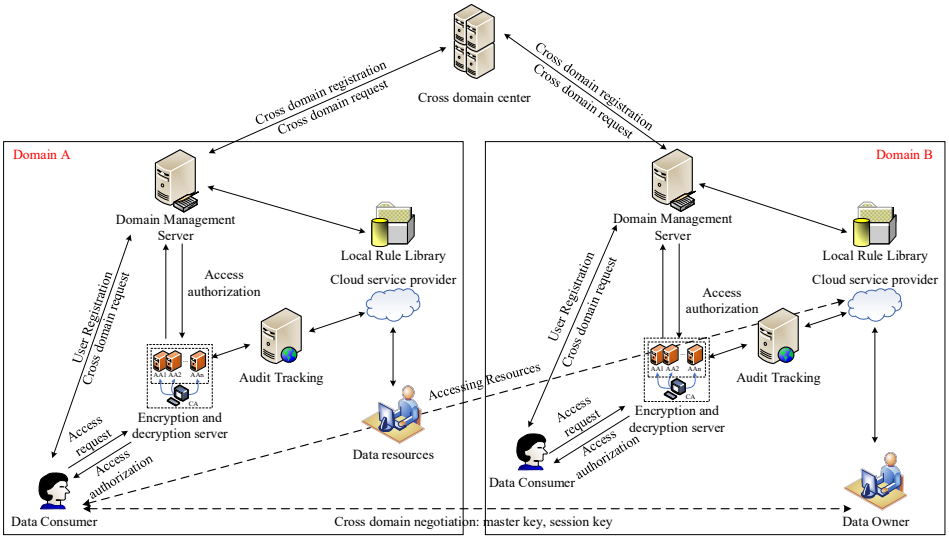
The designed AC policy consists of DO, DC, cross domain centre (CDC), AA and CSP, where CDC and AA are fully trusted and DC and CSP are semi trusted as follows.

1 DC and DO participating in cross-domain data sharing perform information registration and certification at CDC and get $Req_{cd_{reg}}$. DC and DO send domain information D_{inf}, attributes of cross-domain entities $attr_{inf}$, public keys of cross-domain entities pk_{cd}, cross-domain entity signatures $sign_d$, cross-domain attribute certificates cd_attr_{cert}, CDC verifies the legitimacy of the participating

domains and then updates the cross-domain trusted list cdt_{list}. Domain administrators perform cross-domain setups based on the legitimacy of cd_attr_{cert}. When the CDC receives information about a domain that has been registered, it gives priority to finding it in cdt_{list}. If it exists, it directly provides cross-domain services for it, and if it does not exist, it performs legitimacy authentication.

$$DC, DO \rightarrow CDC : Req_{cd_{reg}} = \{D_{inf}, attr_{inf}, pk_{cd}, sign_d, cd\_attr_{cert}\} \tag{17}$$

Figure 5 The designed fine-grained data cross-domain AC policy based on MCACP-ABE (see online version for colours)



2 The CDC performs cross-domain attribute certificate mapping to the DC, where $Res_{cd_{reg}}$ is the authentication response from the DC.

$$CDC \rightarrow DC : Res_{cd_{reg}} = \{acc\_attr_{cert}, sign_{cdc}\} \tag{18}$$

CDC extracts all the attribute information of the participating domains based on $Res_{cd_{reg}}$, and divides the attributes, represents the different attributes of different domains with vectors, binds the correlation of the attribute value space, and constructs the attribute mapping. Based on the mapping results, the CDC will return the mapped attributes and issue a cross-domain access attribute certificate acc_attr_{cert} for the DC, including information about the source and target domains. During cross-domain access during the validity period, the DC can directly use the new request information issued by the CDC for cross-domain access and cross-domain negotiation with the DO.

3 DO and DC perform cross-domain encryption and decryption, and DO applies MCACP-ABE algorithm to design cross-domain access policy and accountability algorithm for data resources. The DO performs a cryptographic operation on M using the cross-domain AC tree $cdac_{tree}$ to obtain the CT and sends the CT to the CSP store. If the DC completes the negotiation and authentication with the domain where

the DO is located, the AA of the domain where the data is located issues the attribute key set sk_{attrs} for the DC, where sk_{attrs} is calculated based on acc_attr_{cert}, CDC signature $sign_{cdc}$, D_{inf}. Finally DC decrypts the data stored on CSP, if DC carries attributes matching $cdac_{tree}$, it can decrypt CT, and access the data successfully, if it does not match with the access policy, the decryption fails.

$$DO \to CSP : Mess = \{M, cdac_{tree} \to CT\} \tag{19}$$

$$AAs \to DC : Mess = \{acc\_attr_{cert}, sign_{cdc}, D_{inf} \to sk_{attrs}\} \tag{20}$$

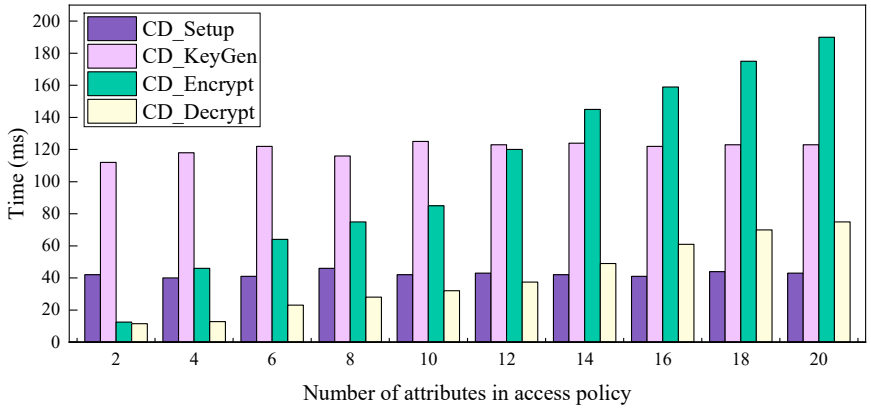
5 Experimental results and analyses

Each domain in this paper's experiments contains the necessary entities, and all machines are interconnected in a local network. The CDC operates on a 1.7 GHz quad-core Intel Core i7 processor, 16 GB of 2133 MHz LPDDR3 RAM, macOS 13.2.1 (22D68), and a database using SQL Server. AAC operations in each domain are performed on a virtual machine set up with 4GB of RAM, ubuntu version number Ubuntu 5.4.0-6ubuntu1, and hosted on a desktop using VMware Fusion 12.1.0. The virtual machine's network connection is configured in bridge mode to connect to the physical network, keeping the host LAN intact.

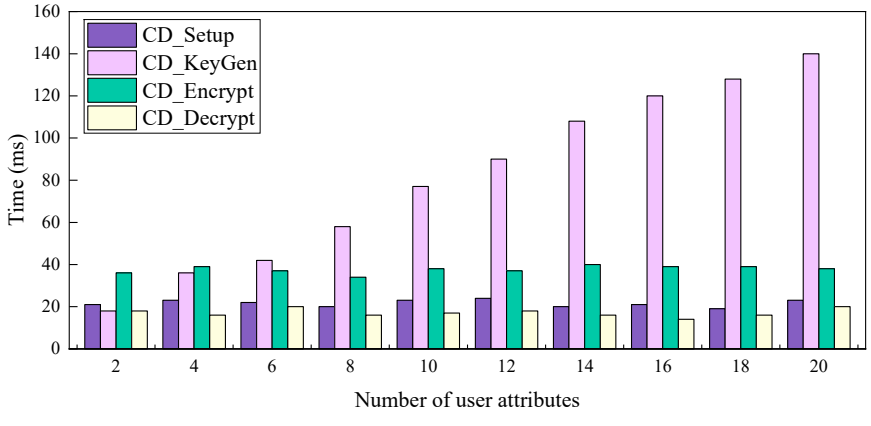
The number of access strategy attributes and the amount of user attributes varied among 2 and 20 in the experiment. By looking at the average time consumption of the four components: cross-domain system setup (*CD_Setup*), multi-privilege attribute key (*CD_KeyGen*), multi-privilege encryption (*CD_Encrypt*), and decrypt (*CD_Decrypt*), as implied in Figure 6. Figure 6(a) shows the average execution time of the access policy attributes, where the amount of access strategy attributes has a linear effect on encryption and decryption, while the average time needed for the other constituents remains largely unchanged. This demonstrates that our method of encryption and decryption solely depends on the number of attributes obtained from the access strategy. Figure 6(b) implies the average execution time for user attributes. As the DC attributes increase, the average time for key generation also increases, while the average time requirement for the other constituents is almost unchanged. This shows that the key generation covers all the attributes of the DC.

To verify the encryption effect of MCACP-ABE and measure the performance of Secure AC, a data file is arbitrarily selected from the database and encrypted by applying MCACP-ABE, and the changes of characters in the data file before and after encryption are shown in Figure 7. From Figure 7(a), it can be seen that before encryption, the characters in the data file are arranged neatly in a certain order, and the ASCII values are also distributed in the same order. In Figure 7(b), the encrypted data file has no regularity of arrangement, and the ASCII values are evenly distributed, which completely hides the characters in the original data file, which indicates that after encryption, MCACP-ABE is able to prevent unauthorised users from directly viewing the data file, thus improving the security of accessing the AC.

Figure 6 The average time consumption of the four components, (a) execution time of each component in relation to access policy (b) execution time of each component in relation to user attributes (see online version for colours)



(a)



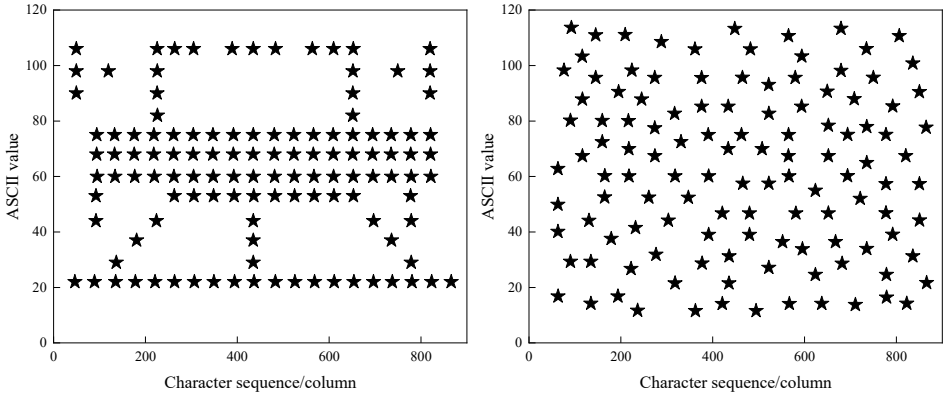
(b)

Furthermore, in this paper, MCACP-ABE is compared with MLCP-ABE (Banerjee et al., 2020) and MACP-ABE (Das and Namasudra, 2023), which are cross-domain AC strategies for data using CP-ABE. Since multiplication operation is much less costly than pairwise and exponential operations, the computational The multiplicative part will be ignored in the computation. The computational consumption of cross domain encryption and decryption for different strategies is shown in Table 1. p_{air} denotes the pairing operation, E_G is the exponentiation time, x is the amount of attributes in the access policy, and I is the complexity of the access policy. The computational consumption of DO and DC for MCACP-ABE is $(2I + 2)E_G$ and $2p_{air} + 2IE_G$, respectively, which are lower than that of MLCP-ABE and MACP-ABE, and thus this paper is computationally efficient.

Figure 8 shows the cross-domain communication efficiency comparison, when the amount of attributes is 20, the cross-domain communication time of MCACP-ABE, MLCP-ABE, and MACP-ABE are 502 ms, 560 ms, and 520 ms, respectively, and as the amount of attributes increases, the overall communication time of MCACP-ABE is shorter than the comparative schemes, and the overall communication time is calculated

as the sum of key generation and attribute encryption and decryption time sum. The decryption times for the three strategies are similar as the attributes increase, and from the perspective of a cross-domain user, the decryption times are almost identical. However, MCACP-ABE is faster in encrypting the system during the time of encryption. Ultimately it can be concluded that MCACP-ABE is usable, secure and effective.

Figure 7 The changes of characters in the data file before and after encryption, (a) distribution of ASCII values of data before encryption (a) distribution of ASCII values of data after encryption

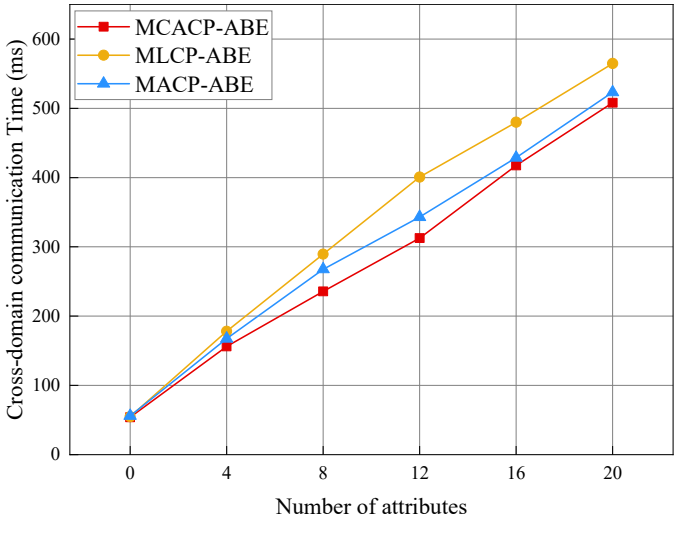


(a) (b)

Table 1 Cross-domain encryption and decryption calculation consumption

Strategy	*DO*	*DC*	*Attribute set*
MLCP-ABE	$(6x + 3)E_G$	$5p_{air} + 6IE_G$	Large
MACP-ABE	$(5I + 2)E_G$	$(3I + 1)p_{air} + IE_G$	Small
MCACP-ABE	$(2I + 2)E_G$	$2p_{air} + 2IE_G$	Large

Figure 8 The cross-domain communication efficiency comparison (see online version for colours)



6 Conclusions

With the increase of collaborative cooperation scenarios in the Internet industry chain, the communication and interaction between organisations are getting closer and closer, facing more and more cross-domain data security threats. This paper designs a fine-grained data cross-domain AC policy based on CP-ABE to address the existing cross-domain AC that is vulnerable to unauthorised access and communication inefficiency. Firstly, CP-ABE is optimised based on multi-authority accountability (MCACP-ABE), which is suitable for fine-grained cross-domain encryption with multi-attribute authority centres, and traces back the malicious user who leaks the private key in the cross-domain access, which achieves fine-grained attribute-based protection of cross-domain data. On this basis, a cross-domain setup module and an attribute mapping module are designed to ensure the legitimacy of all parties involved in cross-domain sharing of data by authenticating the identities of cross-domain data owners and data users, and the mapping of attribute names and attribute value spaces within each domain solves the issue of heterogeneity in inter-domain access management. In addition, a cross-domain AC structure is designed relied on MCACP-ABE, which realises the principle of least privilege for cross-domain AC of data, and improves the usability and security of cross-domain access of data. The results of security proof and performance analysis show that the proposed policy not only satisfies IND-CCA security, but also minimises the computational burden, improves the efficiency of cross-domain communication, and can achieve secure and efficient cross-domain AC.

Acknowledgements

This work is supported by the Key Core Technology Research Project of the 2024 Shaanxi Provincial Key Research and Development Plan (in the field of social development) (No. 2024SF2-GJHX-11).

Declarations

All authors declare that they have no conflicts of interest.

References

Abdelfattah, D., Hassan, H.A. and Omara, F.A. (2022) ‘A novel role-mapping algorithm for enhancing highly collaborative access control system’, *Distributed and Parallel Databases*, Vol. 40, No. 2, pp.521–558.

Banerjee, S., Roy, S., Odelu, V. et al. (2020) ‘Multi-authority CP-ABE-based user access control scheme with constant-size key and ciphertext for IoT deployment’, *Journal of Information Security and Applications*, Vol. 53, p.102503.

Chen, J., Zhan, Z., He, K. et al. (2021) ‘XAuth: efficient privacy-preserving cross-domain authentication’, *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 5, pp.3301–3311.

Cruz, J.P., Kaji, Y. and Yanai, N. (2018) ‘RBAC-SC: role-based access control using smart contract’, *IEEE Access*, Vol. 6, pp.12240–12251.

Das, S. and Namasudra, S. (2023) 'MACPABE: Multi-authority-based CP-ABE with efficient attribute revocation for IoT-enabled healthcare infrastructure', *International Journal of Network Management*, Vol. 33, No. 3, p.e2200.

Hu, V.C., Kuhn, D.R., Ferraiolo, D.F. et al. (2015) 'Attribute-based access control', *Computer*, Vol. 48, No. 2, pp.85–88.

Jia, X., Guo, Y., Luo, X. et al. (2022) 'A perfect secret sharing scheme for general access structures', *Information Sciences*, Vol. 595, pp.54–69.

Jiang, S. (2017) 'State-of-the-art medium access control (MAC) protocols for underwater acoustic networks: a survey based on a MAC reference model', *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 1, pp.96–131.

Karimi, L., Aldairi, M., Joshi, J. et al. (2021) 'An automatic attribute-based access control policy extraction from access logs', *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, No. 4, pp.2304–2317.

Liu, M., Yang, C., Li, H. et al. (2020) 'An efficient attribute-based access control (ABAC) policy retrieval method based on attribute and value levels in multimedia networks', *Sensors*, Vol. 20, No. 6, p.1741.

Nguyen, K.T., Oualha, N. and Laurent, M. (2018) 'Securely outsourcing the ciphertext-policy attribute-based encryption', *World Wide Web*, Vol. 21, No. 1, pp.169–183.

Qi, S. and Zheng, Y. (2019) 'Crypt-DAC: cryptographically enforced dynamic access control in the cloud', *IEEE Transactions on Dependable and Secure Computing*, Vol. 18, No. 2, pp.765–779.

Salehi, A., Han, R., Rudolph, C. et al. (2023) 'DACP: Enforcing a dynamic access control policy in cross-domain environments', *Computer Networks*, Vol. 237, p.110049.

Singh, P., Masud, M., Hossain, M.S. et al. (2021) 'Cross-domain secure data sharing using blockchain for industrial IoT', *Journal of Parallel and Distributed Computing*, Vol. 156, pp.176–184.

Sun, J. and Fang, Y. (2009) 'Cross-domain data sharing in distributed electronic health record systems', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 21, No. 6, pp.754–764.

Xue, L., Huang, H., Xiao, F. et al. (2022a) 'A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums', *IEEE Transactions on Network and Service Management*, Vol. 19, No. 3, pp.2409–2420.

Xue, T., Wen, Y., Luo, B. et al. (2022b) 'SparkAC: fine-grained access control in Spark for secure data sharing and analytics', *IEEE Transactions on Dependable and Secure Computing*, Vol. 20, No. 2, pp.1104–1123.

Yang, X. and Wang, H. (2016) 'A cross-domain access control model based on trust measurement', *Wuhan University Journal of Natural Sciences*, Vol. 21, No. 1, pp.21–28.

Zhang, Y. and Liu, X. (2020) 'An attribute-based cross-domain access control model for a distributed multiple autonomous network', *International Journal of Software Engineering and Knowledge Engineering*, Vol. 30, No. 11, pp.1851–1865.

Zhao, F., Yu, J. and Yan, B. (2022) 'Towards cross-chain access control model for medical data sharing', *Procedia Computer Science*, Vol. 202, pp.330–335.