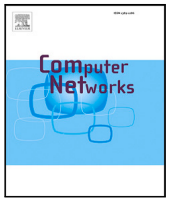
Computer Networks 237 (2023) 110049

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



DACP: Enforcing a dynamic access control policy in cross-domain environments

Ahmad Salehi S. [a,*], Runchao Han [b], Carsten Rudolph [b], Marthie Grobler [c]

[a] Department of Computer Science and Information Technology, La Trobe University, Melbourne, Australia
[b] Cybersecurity LAB (Dep of Software Systems & Cybersecurity), Monash University, Melbourne, Australia
[c] Cybersecurity and Quantum Systems (CQS), CSIRO's Data61, Melbourne, Australia

ARTICLE INFO

Index terms:
Access control policy
Authentication
Attribute-based access control (ABAC)
Cross-domain environments (CDEs)
Security
Privacy

ABSTRACT

Enabling hybrid authorisations to enforce dynamic access control policy from single-domain to cross-domain environments (CDEs) is important for distributed services. However, traditional Attribute-Based Access Control (ABAC) models are incompatible with CDEs. To fill this gap, approaches that apply cryptographic primitives, e.g., attribute-based encryption (ABE), have been proposed. The computation and storage overhead in most ABE constructions is non-negligible and increases with the complexity of the associated policies. In addition, most access control policy systems enforce authorisation policies in a centralised way, raising serious security and privacy issues. In this paper, we introduce DACP – a practical Dynamic Access Control Policy system supporting dynamic cross-domain authorisation. DACP combines traditional ABAC approach and a novel cryptographic primitive Attribute-based group signature (ABGS). ABAC is used for the access control decision and policy enforcement according to the user's attributes whereas ABGS is used for managing the user's attributes between users and authorities. Thus, the user's attributes are securely distributed along with the access structure in CDEs while preserving the user's privacy. We present the concrete design and implementation of DACP, and evaluate it in real-world settings. The evaluation shows that DACP is practical and efficient in CDEs.

1. Introduction

In recent years, the rapid development of technologies contributed to the creation of a large volume of data in different organisational settings and for various different services (e.g., healthcare) [1]. The data is maintained and stored by different domain entities and policies, and access control systems are used to prevent unauthorised access to shared data in cross-domain environments (CDEs) [2–4]. Access to these CDEs is restricted to mechanisms for the explicit sharing of particular pieces of information, and not the dataset as a whole. Approaches to providing more comprehensive access are either based on centralised policies or require synchronised cross-domain policies. Unfortunately, the centralised system creates unwarranted risks for privacy breaches that can affect the complete set of available data. Entities would also need to give up data sovereignty, and a highly trusted entity would need to be established to control access, and decide on and enforce policies. Moreover, most service providers are not interested in sharing their policy settings with other service providers [5]. Thus, an important issue is to determine how policies can be enforced in untrusted CDEs.

Attribute-Based Access Control (ABAC) has gained researchers' attention among existing access control policies as a flexible access control system with dynamic decision capabilities [6–8]. According to the National Institute of Standards and Technology (NIST)'s definition [9], ABAC is a promising access control model that provides a significant opportunity to enforce access control policies based on entities' attributes [10]. It is considered a flexible and scalable dynamic access control policy option. In ABAC, access permits are assigned to entities according to their proven attributes (e.g., subject) [11,12].

There are two approaches to establishing ABAC models [13,14]. *Traditional ABAC models* are designed for centralised or single-domain environments. To date, no traditional ABAC model adequately satisfies the fundamental requirement for CDEs in terms of security and privacy, as all entities define their own policies as incompatible with each other. In addition, most traditional ABAC models need to access users' information, including their identity and attributes. In practice, users may not be willing to reveal such information to others. *Cryptographic ABAC models* employ attribute-based cryptographic primitives in order to enforce access control policies and meet the standard ABAC

* Corresponding author.
E-mail addresses: a.salehishahraki@latrobe.edu.au (Ahmad Salehi S.), Runchao.han@monash.edu (R. Han), carsten.rudolph@monash.edu (C. Rudolph), marthie.grobler@data61.csiro.au (M. Grobler).

https://doi.org/10.1016/j.comnet.2023.110049
Received 14 November 2022; Received in revised form 8 July 2023; Accepted 28 September 2023
Available online 4 October 2023

requirements [15,16]. Such attribute-based cryptographic primitives include attribute-based encryption (ABE) and attribute-based signatures (ABS). In most ABE schemes, the ciphertext size increases with the complexity of the policies. In ABS schemes, revoking permission of entities requires revoking entities' global and local secret keys, leading to high communication overhead in the distributed domain.

1.1. Cross-domain environment (CDE)

We consider a healthcare scenario as an example where data sharing in a CDE would be required. In this example, service providers may need different types of user information (such as doctor identification, attributes, and contextual information) to ensure proper service delivery and users' information may have to be shared with other service providers to achieve this objective. In this scenario, users' information privacy should be a serious consideration in CDEs when data is gathered and made available to other users. In addition, most service providers would prefer not to share their local policy settings with other service providers due to security concerns. For instance, Bob may belong to Domain 2 and can easily access the healthcare data in his domain using a local access control system, but he would need cross-domain permission to access Alice's data as she belongs to Domain 1. In practice, the domains would have separate security policies, and would not be interested in releasing their local security settings to each other the users also would have no interest in disclosing their information to others in the CDEs.

Overcoming these security and privacy challenges is a serious concern for service providers and designers, and several models [17] have been proposed to address this. However, satisfying core security requirements (e.g., attribute management and selective attribute) in a CDE access control policy system has not yet been appropriately considered. To solve this issue, the ideal solution is to find a way to manage user attributes in CDEs with traditional ABAC, where they need to satisfy local security settings. For example, Bob can access Alice's information if and only if he can satisfy access control policies belonging to Alice's authority and not a third-party. In general, ABAC can deliver sufficient security for a single-domain system; however, ABAC is not acceptable for CDEs' access control policy systems with different levels of security settings. Thus, to improve the level of security in CDEs, we need to utilise suitable attribute-based security techniques.

1.2. Our proposal – DACP

This paper proposes a novel ABAC solution for developing and implementing an access control policy based on NIST's ABAC model [9]. We propose a dynamic access control policy (DACP) for CDEs that can support policies associated with the entities' attributes, and exchange and verify the attributes in CDEs. The authorisation engine uses the user attributes to generate permission when an access request is received, enabling the user to access data across multiple domains by satisfying the access structure based on local policy setting. With this policy, the final decision and relevant enforcement entirely depends on the user's attributes. This paper extends an earlier cross-domain access control policy model [18] to facilitate our DACP system. In extending this model, we provide a construction of the DACP model by using a novel cryptographic primitive called Attribute-based group signature (ABGS) [19]. In ABGS, one can sign a message to prove that (1) it is a member of a certain group of people; and (2) it preserves a certain set of attributes, without revealing its identity or its full set of attributes. In the instantiated DACP model, such signature allows verifying messages from other domains while preserving the message signer's privacy.

Contributions of this paper are summarised as follows.

1. We define the DACP system, extending the traditional ABAC model [9,14,20] to support CDEs where a domain can exchange and verify messages from other domains;
2. We provide a construction of the DACP system by using ABGS [19], with security analysis and comparison with existing models; and
3. We implement the DACP system and evaluate its feasibility and performance.

1.3. Paper organisation

Section 2 summarises related works and Section 3 presents attribute-based access control as a background. Section 4 presents the system model and preliminary definitions used in this paper. Section 5 presents the building block of DCAP and Section 6 presents the detailed construction of our proposed DACP system. Sections 7 and 8 analyse DACP's security and performance. Section 9 discusses the significance of our proposed system and discusses the significance of our proposed system. Section 10 concludes the study and pinpoints future works.

2. Related work

This section reviews related works on cross-domain access control, including traditional and cryptographic approaches.

2.1. Traditional ABAC model

The multi-tenant ABAC access control model [21] was proposed to support multiple clients. This model's overall objective is to provide a trust relationship between domains [22] to enforce a single cross-domain policy, removing the possibility of separate and conflicting policies. To further minimise the system's complexity, clustering can be used [23] to reduce the chance of attribute explosion in a large domain by isolating attributes based on user groups and policy decisions [24]. However, this becomes complex because the groups of attributes must be verified and inherited for each request. To address this and the issue of risk-adaptive access control (RAdAC) [25], the quantitative ABAC authorisation model [26] was proposed to rank the attributes based on their priority and condition. Although the presented model components are well-defined and formulated, it is impossible to delegate permission because of the predefined conditions of the RAdAC.

A privacy-preserving access control model and framework were proposed by [27] for secure service provision and composition. An efficient solution is to enforce the policy to protect data privacy during online services [28] in order to enforce access control in web services composition. Although the existing traditional ABAC model [26,29] permits users to adopt a policy in real-time [30], most of them are unable sufficiently satisfy the cross-domain policy system.

2.2. Cryptographic ABAC approach

Attribute-Based Encryption (ABE). ABE is a data access control mechanism in multi-domains [31–33]. In this model, the authority is unable to decrypt messages from another domain without a corresponding key. Another ABE-based data access control model, DAC-MACS [34], was proposed for outsourcing data. A similar framework is used with a global authority to manage and handle the attributes [35]. The majority of ABE models have several drawbacks, such as attribute and key management in CDEs. Similar to the current ABE model [36,37], the third-party system has to permit users to download the necessary file to ensure data availability. The third-party system introduces a variety of vulnerabilities, e.g., the resource-exhaustion attacks.

Attribute-Based Signature (ABS). ABS aims to anonymise authentication between users and service providers, allowing data to be shared

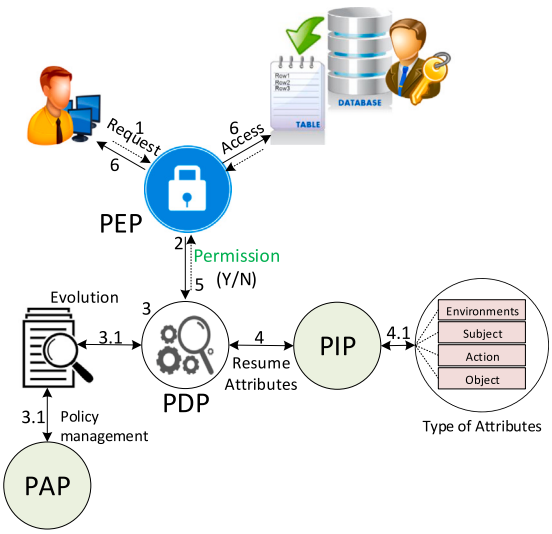


Fig. 1. NIST access control framework [9].

and accessed without revealing the actual identity of the user and patient [38]. It was introduced by Shanqing [39] and extended further using the advantages of ABE and ABS called attribute-based signcryption [40]. A multi-authority access control scheme was further proposed [41], in which the central authority is responsible for managing the required security parameters for users and authorities.

Attribute-Based Group Signature (ABGS). ABGS is a useful cryptographic primitive because users in a foreign domain cannot see who signed the message. This approach is very different from group signatures because the manager and user process their signature associated with a set of attributes, which enables the user to access data only if the user's attributes satisfy the system's access structure [42]. This work was extended to propose a general framework based on ABAC to achieve user anonymity, attribute anonymity, and attribute privacy [19].

3. Attribute-based access control (ABAC)

In ABAC, access to resources is granted by evaluating policies and attributes of the subject, object, environment, and action. The subject sends a request to access particular resources in a single domain. The access is granted based on the system's policies and entity attributes [10,43,44]. Although this provides better access control and flexibility, it cannot be applied in CDEs with multiple levels of security and privacy [9]. NIST has formalised the ABAC framework, depicted in Fig. 1.

The ABAC standard [9,45] has four main components, namely *policy decision point (PDP), policy administration point (PAP), policy information point (PIP),* and *policy enforcement point (PEP)* [9]. Generally, the flow in an authorisation model starts with the PEP, when the subject first sends a request to the system to access medical resources (Fig. 1). At this point, the request is sent onward to the PDP, which evaluates the request with the requirements as related to the subject request. The PDP then forwards a request to the PIP to collect the necessary attributes. A request is also forwarded to the PAP to check the policy related to the request. The output of the system model (e.g., permit, deny, or not applicable) then flows onward to the PEP for future service [8,9,43,45].

4. Dynamic access control policy

In this section, we present DACP system model, workflow and the relevant security requirements for the system model.

4.1. DACP system model

ABAC is designed for a single-domain environment where the data is stored in a single database [18]. However, a number of important real-world applications demand cross-domain settings where multiple domains maintain disjoint data and need to access data from the other domains with minimal trust. ABAC lacks support for accessing data outside the domain and does not support cross-domain settings, motivating the need for the DACP model that extends ABAC to further support cross-domain data access.

We use the healthcare scenario as an example of cross-domain data access. Assuming there are a number of hospitals, each of which constitutes one domain with its own security settings. Each hospital complies with the ABAC model defined in Section 3. In each hospital, some doctors provide healthcare services to some patients. Each patient or user is assumed to belong to a single hospital. Each patient owns their own healthcare data, which is stored in the hospital's database and can only be accessed by some authorised doctors. To fulfil certain duties, a doctor needs to access data from other domains [18]. There are two hospitals belonging to domains 1 and 2 with attribute authority AA_1 and AA_2, respectively. A doctor (as a data user) in Domain 2, may need to access the healthcare data of a patient (as a data owner) that resides in Domain 1.

DACP is an extension of ABAC that additionally supports cross-domain data access with minimal trust. In DACP, both the data owner and the user must register with their respective domains. Each domain is responsible for generating the necessary parameters such as the attributes and key pairs for their entities. This occurs independently in our system model without relying on a third-party. For example, the data user in AA_2 would send a request to access data in AA_1. We assume that both access policies are based on the NIST's ABAC standard [9] with a potential extension of attribute types. For this, the PEP request is forwarded to check the request and invoke a policy decision via the PDP. The request is then forwarded to the PDP to evaluate the access request and monitor if suitable policies can be applied or not. The PDP generates the attributes query based on the access structure and policy defined by the system. The access structure includes specific attributes such as an object, subject, action, and environment. The user can access the data only if their attributes satisfy the access structure as checked by the authorisation engine; otherwise, additional attributes based on the access request might be requested by the evaluation system.

A suitable protocol for the exchange of attributes in CDEs is required. ABGS, such as [19], is one option to confirm attributes without revealing too much information about the user's identity. After the PDP in AA_1 receives and checks the attributes, the authorisation system either permits or denies the access request if the attributes satisfy the policy requirements.

4.2. DACP workflow

We use a simple abstract of cryptographic primitives for DACP and a two-step authorisation for access control. For example, Bob belongs to AA_2 and can easily access the healthcare data in AA_2, but needs cross-domain permission to monitor Alice's data as she belongs to AA_1. We further specify a similar set and function of access control model for single-domain based on [9,14,18,20] but modified it based on our cross-domain model to adapt it to the construction of our ABGS. In our system, a single-domain user has its own internal authorisation system to verify and validate the requests from its local users. This has already been addressed in other research [14] and is therefore out of scope for this study.

As depicted in Fig. 1 and the DACP system model, our authorisation protocol is abstracted as follows. The first step is set up to initiate the attribute authorities (e.g., AA_1 and AA_2); each authority is responsible for generating the security parameters (e.g., attributes and key pairs) for their entities. A set of attributes is assigned to a user (Bob), based

on his duty and eligibility; Bob sends a request to access Alice's data (AA_1). The authorisation system (PEP) at the AA_1 evaluates the user's request and forwards his request to the policy engine (PDP) for further evaluation. The PDP evaluates the request and creates a response along with the Γ (here the predicate is acting as an access structure tree, similar to τ in traditional ABAC) based on the policy system and forwards it to the PEP. The PEP transmits the response back to the user, with the Γ specifying what attributes are required to approve the access. The user selects a subset of his attributes based on Γ, and obtains the key pairs to generate a signature of the new set of attributes. The user signs this request with the appropriate information (e.g., Γ), and returns it to the PEP at AA_1 for further processing. The user's attributes are then extracted from the signature if, and only if, the signature is verified and validated by the PEP at the AA_1. Permissions are generated for the user and forwarded to the user if, and only if, the user is valid and the user's attributes can satisfy the policy defined by the authorisation system (PDP); otherwise, the access request is denied.

4.3. Security requirements of DACP

We consider the following security requirements for DACP.

Attribute management: Attributes need to be maintained and controlled in the CDE during the access request to reduce the complexity of the cross-domain access control policy system.

Selective attribute: The minimum number of attributes should be used in access control models to satisfy the access structure defined by the system.

Secure decision-making of policy: Based on existing literature, policies and permissions split the cross-domain validation into central validation. This means that the confirmation of policies is happening in a domain (e.g., third-party) where the actual data may not belong. This introduces several risks, while third-parties are evaluating the policy or decision.

Flexible access control: The access control model should be flexible enough to enable a user to obtain access to specific data without any registration from the local authority to read and write.

5. Building block of DACP

Compared to ABAC, DACP additionally allows a user to access data in another domain. To be authorised, the user has to prove that it (1) preserves all attributes required by the data; and (2) belongs to an authorised domain, *without revealing the user's identity*.

As shown in Section 3, the first requirement has been studied and achieved in ABAC systems through various approaches, especially attribute-based signatures. However, the latter requirement is left as an open question in access control literature. A widely adopted approach in cryptography to prove group membership while preserving the group participant's privacy is by using the *group signature* primitive. In a group signature, a group participant can sign messages. Anyone can verify whether the signer belongs to that group, but the identity of the signer remains hidden. However, the group manager can learn the signer's identity via a special trapdoor.

To achieve the two requirements simultaneously, a combination of attribute-based signatures and group signatures is necessary. Fortunately, such a combination, which is known as *ABGS* [19], has been proposed. In ABGS, a group manager can issue (and revoke) key pairs to participants. Participants can sign messages, and anyone with the signature can verify that (1) the public key is issued by the group manager, (2) the signature is for the message, and (3) the predicate is a subset of attributes of the public key. With the group manager's secret key, one can further determine which participant in the group signed the message.

Formally, ABGS is defined as a tuple of algorithms (Setup, ABGKeyGen, ABGSign, ABGVerify, ABGOpen):

Table 1
Summary of notations.

Notation	Description
1^λ	Security parameter for ABGS
1^n	Maximum number of attributes in ABE
AA_k, AA_ℓ	The attribute authority in domain k and ℓ
ID_k^{AA}	The ID of AA_k
sk_k^{AA}, pk_k^{AA}	The secret key and public key of AA_k
$param, msk$	ABGS' parameter and master secret key
r, crs	NIZK's randomness and common reference string
U_i^k	The ith user in domain k
$sk_{k,i}^U, pk_{k,i}^U$	The secret key and public key of U_i^k
$\mathbb{A}_i$	The attribute set of user U_i^k
$sk_{k,i}^{\mathbb{A}}$	The attribute secret key of U_i^k
Λ_i	The subset of $\mathbb{A}_i$
Γ	Predicate of U_i^k (access structure tree)
π_i^k	The certificate of U_i^k
Σ	The final signature

Setup$(1^\lambda, 1^n) \rightarrow (param, msk, mpk)$**:** Security parameter 1^λ and the maximum number of attributes 1^n serve as inputs, a public parameter $param$, a master secret key msk and its corresponding public key mpk serve as outputs.

ABGKeyGen$(param, msk, \mathbb{A}) \rightarrow (sk_{\mathbb{A}}, pk_{\mathbb{A}})$**:** $param$, msk and the attribute set $\mathbb{A}_i$ serve as inputs, a secret key $sk_{\mathbb{A}}$ and its corresponding public key $pk_{\mathbb{A}}$ serve as outputs.

ABGSign$(param, sk_{\mathbb{A}}, msg, \Gamma) \rightarrow \sigma$**:** $param$, $sk_{\mathbb{A}}$, a message msg, and a predicate i.e., subset of attributes Γ serve as inputs, generating signature σ.

ABGVerify$(param, mpk, msg, \sigma, \Gamma) \rightarrow 0/1$**:** $param$, mpk, msg, σ and Γ serve as inputs, either 0 (false) or 1 (true) as output.

ABGOpen$(param, msk, \sigma) \rightarrow \mathbb{A}/\perp$**:** $param$, msk, and σ serve as inputs, generating either an attributes set $\mathbb{A}$ or an empty set $\perp$.

ABGS satisfies three security properties, namely *user anonymity*, *attribute anonymity*, and *full traceability* [19]. User anonymity ensures that given a signature, anyone apart from the group manager cannot recover the signer's identity. Attribute anonymity ensures that given a signature and a predicate, anyone can only learn whether the signer satisfies the predicate or not, but cannot learn all attributes preserved by the signer. Full traceability ensures that anyone cannot create a valid signature that cannot be opened by the group manager.

6. Construction of DACP

In this section, we present the construction of our proposed model, a policy-based cross-domain access control framework using the workflow presented in Section 4.2, based on ABAC [9,20] and ABGS [19]. We used the concept of [19] to cryptographically exchange the required attributes for our policy system in CDEs. This helps to exchange and confirm the user's attributes in CDEs without revealing any information (e.g., attribute and identity) while attributes need to be transferred and used in the policy system. The proposed scheme is detailed in the following subsections, with the assumptions relevant to each step. Table 1 summarises all notations used in our construction.

Certificate authority setup (Algorithm 1). This step – executed by a *certificate authority* – initiates the CDEs system. The certificate authority issues an ID ID_k^{AA} for each attribute authority AA_k who governs a domain.

Algorithm 1: System Setup, executed by the certificate authority.

for $k \in [1, n]$ **do**
  Generate ID ID_k^{AA} for AA_k;

Attribute authority setup (Algorithm 2). In this step, each attribute authority AA_k generates the necessary metadata and key pairs for itself and acts as a group manager and certificate issuer in the ABGS scheme for governing domain k. AA_k generates the public parameter *param* and master secret key msk by running $\mathsf{Setup}(\cdot)$, and generates key pair (sk_k^{AA}, pk_k^{AA}), randomness r and common reference string crs by running $\mathsf{AAKeyGen}(\cdot)$. Randomness r and common reference string crs are used in the Non-Interactive Zero-Knowledge (NIZK) scheme.

Algorithm 2: Attribute Authority Setup, executed by each attribute authority AA_k.

Input: Security parameter 1^λ.
Output: Parameter *param*, master secret key msk, key pair (sk_k^{AA}, pk_k^{AA}), randomness r and NIZK's CRS crs.
$(param, msk) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$;
$(sk_k^{AA}, pk_k^{AA}), (r, crs) \leftarrow \mathsf{AAKeyGen}(1^\lambda)$;

User setup inside domain (Algorithm 3). In the user setup, each of AA_k enrols user U_i^k to domain k. Specifically, user U_i^k requests AA_k with his set $\mathbb{A}_i$ of attributes. Then, AA_k generates an attribute secret key $sk_{k,i}^{\mathbb{A}}$ and key pair $(pk_{k,i}^U, sk_{k,i}^U)$ for user U_i^k by running $\mathsf{ABGKeyGen}(\cdot)$.

Algorithm 3: User Setup. Executed by each user U_i^k.

Input: Parameter *param*, master secret key msk, security parameter 1^λ, and user i's attribute set $\mathbb{A}_i$
Output: User's key pair in AA $(pk_{k,i}^U, sk_{k,i}^U)$ and attribute secret key $sk_{k,i}^{\mathbb{A}}$.
$(sk_{k,i}^{\mathbb{A}}, pk_{k,i}^U, sk_{k,i}^U) \leftarrow \mathsf{ABGKeyGen}(param, msk, \mathbb{A}_i)$;

Cross-domain environments request (Algorithm 4). In the CDEs request step, user U_i^k in domain k requests access to another domain ℓ. For example, a doctor U_i^k in hospital k requests to access healthcare resources at hospital ℓ. AA_ℓ then needs to authenticate U_i^k by checking whether (1) U_i^k has the predicate, i.e., the required set of attributes, and (2) U_i^k is a member in domain k.

After receiving U_i^k's request, the PEP in domain ℓ forwards the request to the PDP for validation. The PDP records this request and generates a certificate $\pi_{\ell,i}$ for U_i^k. The certificate $\pi_{\ell,i}$ – represented as the signature of $pk_{k,i}^U$ signed by $sk_{k,i}^U$ – is used to prove the user's ownership of $sk_{k,i}^U$. To authenticate U_i^k, the PDP challenges the user to provide a predicate Γ, i.e., a set of attributes $\Lambda_i \in \mathbb{A}_i$, specified by the policy system. In order to get the access, U_i^k should provide a valid ABGS signature on $\pi_{\ell,i}$ with predicate Γ.

Sign (Algorithm 5). U_i^k signs $\pi_{\ell,i}$ with $sk_{k,i}^{\mathbb{A}}$ and the predicate $\Gamma \in \Lambda_i$, using $\mathsf{ABGSign}(\cdot)$. $\mathsf{ABGSign}(\cdot)$ outputs a final signature Σ. U_i^k then replies Σ to AA_ℓ. We instantiate the CP-ABE in ABGS using [46,47], and the Groth–Sahai protocol [48] for NIZK.

Access decision (Algorithm 6). AA_ℓ decides whether to grant U_i^k access to domain ℓ by running $\mathsf{ABGVerify}(\cdot)$ (in Algorithm 6). If Σ is verified, then the protocol returns 1 and AA_ℓ grants U_i^k access. Otherwise, the protocol returns 0 and AA_ℓ rejects the access request.

Algorithm 4: Access Request Setup. Executed by U_i^k

Input: User's secret key $sk_{k,i}^U$, a subset of user's attributes $\Lambda_i \subset \mathbb{A}_i$.
Output: User's key pair $(sk_{\ell,i}^U, pk_{\ell,i}^U)$ and certificate $\pi_{\ell,i}$.
// Join phase
A user U_i^k requests AA_ℓ to enrol;
AA_ℓ runs $\mathsf{ABGKeyGen}(param, msk, \Lambda_i)$ to generate his key pair $(sk_{\ell,i}^U, pk_{\ell,i}^U)$;
AA_ℓ sends $(sk_{\ell,i}^U, pk_{\ell,i}^U, pk_\ell^{AA}, crs)$ to U_i^k;
U_i^k uses $sk_{k,i}^U$ to sign $pk_{k,i}^U$ and gets π_i^k;
U_i^k requests AA_ℓ with $\mathsf{Join}(\pi_i^k, pk_{k,i}^U)$;
// Challenge phase
AA_ℓ sends $\pi_{\ell,i}$ and the predicate $\Gamma \subset \Lambda_i$ to U_i^k;

Algorithm 5: Sign. Executed by user U_i.

Input: Parameter *param*, NIZK's CRS crs, U_i's attribute secret key $sk_{k,i}^{\mathbb{A}}$, the predicate Γ, and U_i^k's certificate $\pi_{\ell,i}$.
Output: the final user signature Σ.
$\Sigma \leftarrow \mathsf{ABGSign}(param, crs, pk_{k,i}^U, \pi_{\ell,i}, sk_{k,i}^{\mathbb{A}}, \Gamma)$;
return Σ

Algorithm 6: Access Decision. Executed by AA_ℓ

Input: Parameter *param*, $pk_{\ell,i}^U$, pk_k^{AA}, U_i^k's certificate $\pi_{\ell,i}$, the verifiable signature key Σ and Γ.
Output: return 1 if the user Σ verified; otherwise the outcome is 0.
$result \leftarrow \mathsf{ABGVerify}(param, pk_{\ell,i}^U, pk_k^{AA}, \pi_{\ell,i}, \Sigma, \Gamma)$;
return Λ_i

Identifying misbehaviours (Algorithm 7). It is possible that some users behave maliciously after being granted access to domain ℓ. If a user exhibits suspicious behaviour, AA_ℓ can open Σ to view the certificate $\pi_{\ell,i}$ by running $\mathsf{ABGOpen}(\cdot)$ (in Algorithm 7). The extracted attributes are forwarded to the PDP for the final evaluation. Hence, suitable permissions are created for the user if the user's attributes can satisfy the policy system defined by the authorisation system on behalf of the patient; otherwise, the access request is denied.

Algorithm 7: Opening the signature. Executed by AA_ℓ

Input: Parameter *param*, master secret key msk, NIZK's CRS crs, pk_k^{AA}, Γ, U_i^k's certificate $\pi_{\ell,i}$, and the verifiable signature key Σ.
Output: return 1 if the user Σ verified; otherwise the outcome is 0.
$\Lambda_i \leftarrow \mathsf{ABGOpen}(param, msk, crs, sk_\ell^{AA}, \Sigma)$;
return Λ_i

Remark. We say that the DACP scheme is correct if, for Algorithms 1–6, the verification algorithm output must be 1 and then the output of algorithm 6 is true and not $\perp$. In comparison to the original ABGS, our work is partially decentralised and can be applied to the CDEs. The original ABGS is centralised, and users' attributes and security parameters generate and are managed by a central system.

7. Security analysis

In this section, we analyse the proposed DACP system in terms of the security requirements, and show that it satisfies all requirements.

7.1. Attribute management

The minimum number of attributes should be used in the access control system to reduce the size of the message, signature and attribute secret key. Moreover, no user should be able to lie about the attributes they have in their system. To address this, we used ABAC which is mainly focused on policies. There are a few types of attributes required that can be used in the policy system based on ABAC and the business system model. With DACP, we show that our work is the only system that can achieve these critical requirements in CDEs.

7.2. Cross validation

As mentioned in the literature review, central cross-policy validation is important. In DACP, the policies and final decision-making occur in the domain to which the data owner belongs. According to this rule, a user's universal attribute needs to be securely transferred to the data owner's domain, to maintain the user's privacy and cut down the risk of cross-domain policy evaluation. Paying attention to cross-validation is essential as all the cross-domain access requests have the same weight, and the system administrator can add more critical ways to have a better policy system.

7.3. Flexibility

Our method uses the popular NIST access control policy standard, which is mainly focused on how to enforce static security setting policies. NIST also introduced ABAC, which supports many requests and produces permission under various local domain conditions and targets. Observe that these policies should not be distributed in CDEs due to the privacy-preserving static security settings. Therefore, obtaining the attributes of a user who would like to gain access is well-suited. To achieve this, we used the ABGS' cryptographic approach. Our system promotes higher flexibility for dynamic authorisation in CDEs.

7.4. Collusion-resistance

Our protocol does not allow users to combine their attributes, as long as only a single signature is used to validate the access structure. For example, collusion might be possible if there are several steps. If in the first exchange of attributes the PDP is not satisfied, it may request additional attributes. If only these additional attributes are confirmed in the second step, and the initial attributes from the first step are not confirmed again, collusion may occur. Therefore, it is always necessary to extend and re-confirm previous attributes if several exchanges are necessary for the same access request.

7.5. Revocation support

A user can obtain a signature in our system if, and only if, the user's attributes are valid. This means that the signature is dependent on the user's attributes for each access request. A user cannot validate in CDEs if an attribute is revoked or the user is no longer a valid user within the domain. According to our system, attribute lists are compactly stored in AA, and the AA can easily renew, rename or delete their users' attributes while any attributes need to revoke. As a result, the revoked attributes are no longer valid to allow users to use them for other access. Additionally, AA can revoke the eligibility of the user by only changing the access structure. Then, the user's signature and identity would no longer be valid.

8. Implementation and evaluation

In this section, we implement the DACP system and evaluate its performance. The evaluation results show that the DACP system is practical.

8.1. ABGS: implementation and evaluation

To implement the DACP system, we instantiate ABGS with existing cryptographic primitives, implement the instantiated construction and evaluate its performance. The evaluation shows that the instantiated ABGS construction, although embedding a group signature, is about 80 times faster in terms of setup and four times faster than the state-of-the-art ABS implementation in terms of verifying messages. The signing speed remains stable and performs better than the ABS implementation with more than five attributes.

(A) Implementation:

We implement the ABGS scheme proposed by Kuchta et al. [19]. Each component in the ABGS scheme is instantiated as follows. For the ABE algorithm, we choose Rouselakis-Waters large-universe CP-ABE (RW13) [46], which inherently supports key encapsulation. As RW13 requires bilinear pairing, we choose `a_160_512.properties`, a group of pairing parameters pre-generated in the JPBC library [47]. `a_160_512.properties` can generate type A prime-order bilinear groups with 160-bit Z_p and 512-bit G. We adopt the SHA1 PRNG to generate random security parameters, AES for symmetric encryption (data encapsulation), Ed25519 [49] over Curve25519 [50] for digital signatures, ECIES [51] over AES and Curve25519 for asymmetric encryption, and the Groth–Sahai proof system [48] for NIZK.

We implement ABGS using the Java programming language. We use BouncyCastle (v1.60) [52] for basic cryptographic primitives, the JPBC library (v2.0.0) for bilinear pairing, [53] for the RW13, [54] for the Groth–Sahai, and [55] as the ECIES. We use Gson [56] for serialising and deserialising Java objects. As a proof-of-concept, we do not apply any optimisation techniques in this ABGS implementation but can consider this for future work.

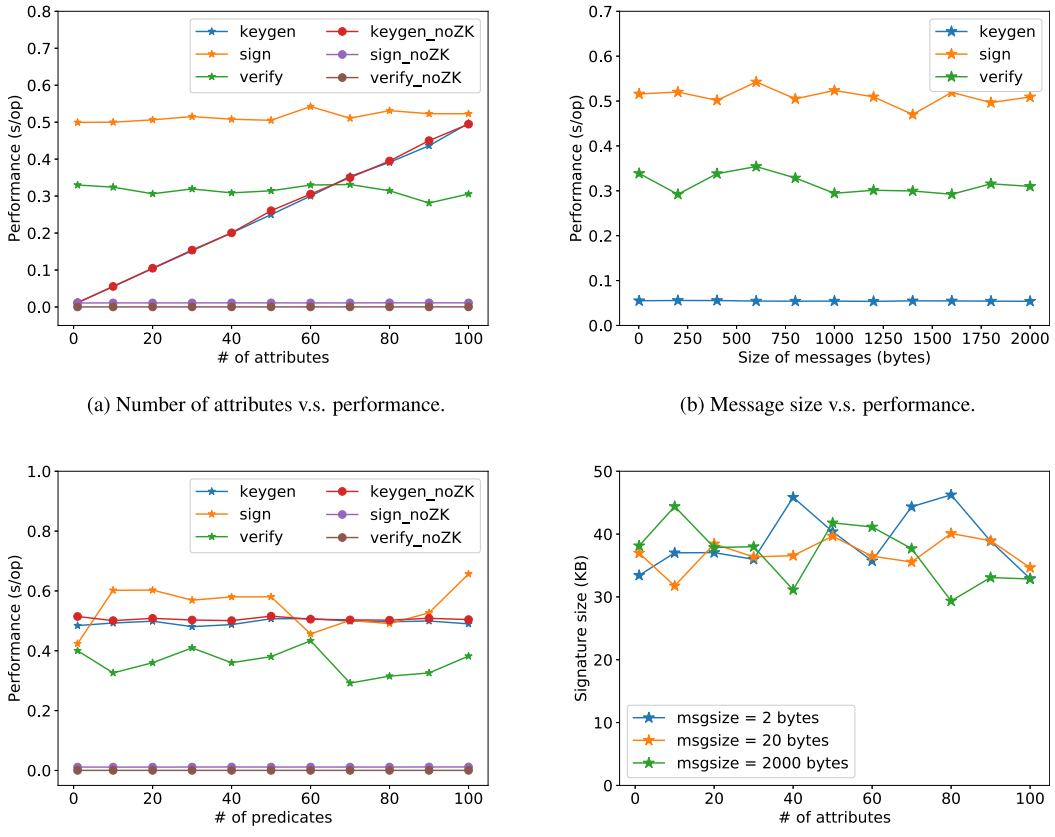
(B) Experimental setting:

All experiments were performed on a 2018 Macbook Pro with an Intel Core i7 CPU @ 2.2 GHz and 16 GB of RAM running OSX 10.14.6 and OpenJDK 1.8.0_222. We use the Java Microbenchmark Harness (JMH) framework v1.21 to benchmark the performance of ABGS. Each workload is first invoked twice for warmup, then invoked an additional five times. Each invocation repetitively runs the workload for 10 s, in order to get the average execution time of the workload. We calculate the average execution time and its standard deviation across five executions. We conduct four groups of experiments on ABGS, namely (a) number of attributes v.s. performance, (b) message size v.s. performance, (c) number of predicates v.s. performance, and (d) impact of message size and number of attributes on the signature size.

(C) Results and analysis:

Figs. 2(a)–2(c) show the performance of ABGS with different numbers of attributes, different message sizes, and different numbers of predicates, respectively. The performance of ABGS is shown to be acceptable. Signing and verifying messages take approximately 0.45 s and 0.3 s regardless of the number of attributes, the message size, or the number of predicates, because the signing and verifying of messages are bottlenecked by generating NIZK proofs and verifying NIZK proofs, respectively. As the statements used in NIZKs are irrelevant to these three parameters, the performance remains stable regardless of the values of these parameters. Meanwhile, the time of generating a key for a group member increases linearly with the number of attributes but has no relevance to the message size because the underlying CP-ABE's key generation has a linear time complexity with attributes.

Fig. 2(d) shows that the signature size is also irrelevant to the number of attributes or the message size since the two NIZK proofs are the most space-consuming part of the signature. The signature size does not monotonically increase or decrease but fluctuates with increasing message size and attributes. Such deviation is introduced by compressing signatures. In order to save space, Gson applies the GZIP [57] compressing algorithm when serialising objects. GZIP achieves different compression rates on different datasets, depending on the distribution of bits.



(a) Number of attributes v.s. performance.

(b) Message size v.s. performance.

(c) Number of predicates v.s. performance.

(d) Signature size.

Fig. 2. Performance evaluation of ABGS.

Table 2
Overhead comparison between ABGS and ABS.
Source: The data of ABS is from [38].

# of attributes	KeyGen					Sign					Verify				
	10	20	30	40	50	2	4	6	8	10	2	4	6	8	10
ABS [38]	1 s	2.1 s	3.2 s	4.2 s	5.3 s	0.2 s	0.3 s	0.5 s	0.8 s	1.0 s	0.075 s	0.075 s	0.075 s	0.075 s	0.075 s
ABGS without NIZK	0.05 s	0.10 s	0.15 s	0.20 s	0.25 s	0.02 s	0.02 s	0.02 s	0.02 s	0.02 s	0.01 s	0.01 s	0.01 s	0.01 s	0.01 s
ABGS with NIZK	0.05 s	0.10 s	0.15 s	0.20 s	0.25 s	0.50 s	0.50 s	0.50 s	0.51 s	0.51 s	0.32 s	0.32 s	0.30 s	0.31 s	0.30 s

Impact of NIZKs on the performance. NIZKs bottleneck the performance of signing and verifying messages. To improve the performance of signing and verifying messages, NIZKs should be the first to optimise. While Groth–Sahai is not state-of-the-art, instantiating the ABGS with more advanced general-purpose NIZK protocols can accelerate signing and verifying messages directly. Also, tweaking the ABGS construction to replace the general-purpose NIZKs or even mitigate NIZKs can be helpful.

To prove this, we test the performance of ABGS when omitting NIZK proofs, refer to Figs. 2(a) and 2(c). It shows that, while the performance of generating keys remains unchanged, signing and verifying can become significantly faster when NIZKs are omitted. This is out of the scope of this study.

Comparisons. Before this research, there exists no implementation or evaluation of ABGS. Thus, we compare our ABGS with the ABS scheme [38] (which does not employ NIZKs) as a reference. Table 2 provides the comparison results between ABGS and ABS.

The results show that the ABGS without NIZKs is much faster than the ABS scheme, while the ABGS with NIZKs is slightly slower than ABS, but the overhead is still acceptable. Specifically, for both constructions, the setup time increases with the number of attributes, and our ABGS scheme is 80x faster than the ABS construction. The time required to sign a message remains stable for our ABGS, increasing with the number of attributes for the ABS. With less than five attributes, the ABS scheme is faster than the ABGS scheme, and vice versa. For verifying messages, the ABS scheme is about four times faster than the ABGS scheme. As stated above, given that the number of attributes is reduced in the DACP system, the overhead of ABGS is acceptable. The overhead is mainly from the Groth–Sahai NIZKs and can be further minimised by using more efficient NIZK constructions.

8.2. DACP: implementation and evaluation

Based on the ABGS implementation, we implement the DACP system in the healthcare context and evaluate its performance. The evaluation shows that the DACP system is also practical.

(A) Implementation:

The DACP system consists of two components, namely an ABAC system and ABGS that enables cross-domain data access in ABAC. We implement the ABAC system as follows. We use the standard eXtensible Access Control Markup Language (XACML) to define access control

policies.[1] We develop ABAC components based on the Organisation for the Advancement of Structured Information Standards (OASIS) as bundles on Java [58,59]. OASIS is based on REST interfaces and XACML. We implemented our PDP by using and extending the open-source WSO2 Identity Access Management Server (IMAS) [60], which allows us to add, remove or edit policies. We implement the web service by using the Spring MVC framework [61], implement the database interface by using the SQLAlchemy library [62], and employ the MySQL database to store policies. We also implement a random policy generator using the WSO2 server to accommodate many policies [60], as well as a custom PIP to store the attribute values.

We then integrate ABGS into the ABAC system to establish the DACP system. Specifically, the ABGS is integrated into an attribute assertion system (AAS), which is a component to extract the user's attributes. According to our local authorisation system, the extracted attributes are passed into the PEP. To implement AAS, we use attribute extraction and mapping implementation by [63]. The extracted attributes are published through the system interpreter using MySQL to store in the PIP table. We adopted three implementation types related to single-domain (ABAC) and cross-domain (ABGS-based) access control.

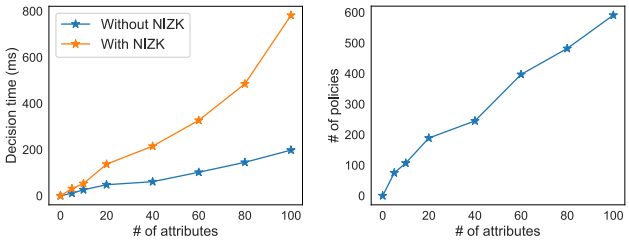
(B) Results and analysis:

Nowadays, domains and infrastructures require control over numerous access rules and policies distributed among multiple duties, security and privacy levels. This makes it very hard to manage and control authorisation in CDEs, especially those using any central system. Our system offers an independent authorisation system, which includes a standalone authorisation system. As mentioned in our system model, the local healthcare system is a trusted domain, with each domain responsible for managing its users and related services. We used the concept of ABGS to provide a secure connection between attribute authority's using their public certificates. This enables the authorities to manage and distribute users' attributes securely. Therefore, there is no need to transfer and distribute local policy security settings in CDEs. This significantly reduces the complexity of access control policy systems associated with identity access management, and policies in CDEs.

DACP reduces the number of policies required to be controlled by the decentralised system. Our system is designed and developed to reduce the decision time in CDEs and significantly enable the local domain to standardise their policies in CDEs. Access to the resources is clearly defined and controlled by the local domain security settings where the data are stored. Hence, our system enables attribute authorities to keep their security setting and gives the authorities to efficiently supervise and control restrictions and permissions across the board from a centralised platform.

Discussion. Fig. 3 provides the decision time (with and without NIZK) and the number of policies for generating new permissions for a user with up to 100 attributes. It shows that, after integrating ABGS into the ABAC system, the number of policies and access decision time is positively affected by using a high number of attributes for each access request. This means the processing time of this experiment grows linearly with the number of attributes, which is logical. Hence, the access decision time directly relates to the number of attributes used in the policy system. This makes our system model robust and practical as our system aims to select the minimum attributes in CDEs.

We also observe that the traffic overhead towards the interpreter system only depends on the number of attributes rather than other aspects, such as the number of users and access requests. A user belongs to a different domain in our system and can share the same attributes from single- to cross-domain based on the access request. Thus, the



(a) # of attributes v.s. Decision time, i.e., the time taken for DACP to decide the access of a user with 1-100 attributes.

(b) # of attributes v.s. # of policies, i.e., the number of policies used for deciding the access of a user with 1-100 attributes.

Fig. 3. Evaluation of DACP.

system's number of attributes will be smaller than existing ABAC systems, further reducing the system's computation and communication overhead. Also, the processing time of decisions with and without NIZK is different, giving flexibility over the security-performance trade-off when instantiating the system. However, our system is entirely acceptable based on decision time. The performance with NIZK can be further optimised by using more efficient NIZK protocols [64–67] or applying implementation-level optimisations [68].

Due to the facts that (1) DACP is the first of its kind that uses ABGS to reduce the number of attributes required in CDEs, and that (2) most of the access control systems are not open-source and thus, we cannot obtain any results under our experimental settings, we, unfortunately, cannot establish a fair benchmark for comparing DACP with other access control systems. In the future, we aim to establish such a fair benchmark specialised for the healthcare domain, implement other access control systems, and compare DACP with them.

9. Comparison with existing ABAC approach

In this section, we compare the proposed system model with existing traditional and cryptographic ABAC approaches. We compare the complexity of the policies mined by different ABAC mining algorithms.

9.1. The DACP scheme

Our system employs two-step authorisation [9,14,19,20] for access control: the *local authorisation* is based on a single ABAC model (see Section 3) and the *authorisation attributes in CDEs are securely and cryptographically exchanged* (see Sections 4.1 and 4.2), based on the ABGS model. In the first step, a user authenticated with his user attribute accesses the signature structure, i.e. the user is able to access sensitive records while his signature is verified with ABGS. In our ABGS, a person is able to sign a message if only his attributes satisfy the predicate associated with the signature; otherwise, the signature failed in a generation as well as the verification process. In the second step, the attributes of the user signature are extracted while ABGS verifies the signature. The extracted attributes will pass through the PDP via the local authorisation system and be stored in the PIP. Finally, suitable permissions are produced by the local authorisation system after both access control factors are achieved under different roles and conditions of the local authorisation system.

9.2. Comparison

We found that the ABAC approach is active within the access control policy systems for single-domain through the current approaches and literature review. We found that today's research has employed a considerable number of ABAC approaches in their recent developments in single-domain and the obtained outcomes showed that this approach

[1] Please refer to Section 3 for a summary, and to [9,14,20] for more details on XACML.

Table 3
Comparison with existing ABAC system.

	Cryptographic [19,37,38,40]	Traditional [22,29,33]	Our work
Attribute type	Subj, Obj	Sub, Obj, Act, Env	Sub, Obj, Act, Env
CDEs attribute-based	Partially	No	Yes
Policy distribution	Yes	Yes	No
Third-party trust level	High/Medium	High	Low
Attribute management	Partially	Partially	Yes
Selective attributes	Yes	No	Yes
Complexity of access policy	Medium	High	Low
ABGS model	Static	N/A	Dynamic
ABAC model	Static	Static	Dynamic

is promising and user-friendly technology as the system is attribute-based [60]. However, none of these works satisfies all the requirements identified in our study. As outlined in the literature review, there is no fundamental approach and valid research to employ attribute-based methods in CDEs to meet the security setting and requirements identified by our research study. Furthermore, existing works focus on exchanging the domain policy in CDEs, potentially introducing policy conflicts and increased security risks associated with compromised authority. This is outside the scope of this study.

Moving from a single-domain access control model to CDEs, we selected ABGS and modified it to provide a CDEs solution based on our ABAC model. ABGS can provide a platform for our system to accurately exchange the user's attributes cryptographically, as required for our access control policy system. Using ABGS enables our system to achieve a group manager property for every single domain, providing possibilities to evaluate and control the users in CDEs and giving attribute anonymity and privacy properties for our proposed model. We not only used the existing ABGS but also implemented this to represent the possibility and feasibility of our system in practice. We showed that our system has an acceptable output in signing, verifying, key generation, and size of message and signature, as discussed in Section 8.1. We showed that the user could gain access to particular data in CDEs using their own attributes based on his access request. We further showed that our system is working well by omitting NIZK. This can be part of future work to optimise the process.

In general, the complexity of access control policy is directly related to the number of user attributes in the system. This is because the cross-domain system requires the transfer of more attributes for verification between users and service providers. However, the complexity of DCAP is low and acceptable because our system uses the least number of attributes in cross-domain. A comparison of the proposed work with the traditional and cryptographic ABAC works is depicted in Table 3. We see that our system model is dynamic, and our system does not require distributing the domain security settings in CDEs.

10. Conclusion and future work

DACP is a new access control policy system that permits a user to access sensitive data in CDEs, based on traditional ABAC and a novel cryptographic primitive ABGS. The hybrid design enjoys the advantages of traditional and cryptographic approaches, enabling domains to control their security settings and policies independently. DACP is designed to prevent attribute collusion and is partially decentralised with dynamic authorisation, which allows users to access data without relying on a third-party. DACP is flexible enough to allow users to access data in CDEs without user privacy disclosure. Our implementation and evaluation have shown the feasibility of our proposed system. In practice, the application of this system is useful to prevent unauthorised user access in CDEs. We believe that our system can overcome the problems of existing traditional and cryptographic access control policy systems and meet the security and privacy requirements in collaborative environments such as healthcare.

Moving forward from this research, we plan to extend and formulate an ABAC component to integrate with existing open-source web services. We further plan to develop the construction of original $\mathsf{ABGOpen}(\cdot)$ only to recover attribute sets that are considered.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] K. Hasan, M.J.M. Chowdhury, K. Biswas, K. Ahmed, M.S. Islam, M. Usman, A blockchain-based secure data-sharing framework for software defined Wireless Body Area networks, Comput. Netw. 211 (2022) 109004.

[2] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, B. Fang, A survey on access control in the age of Internet of Things, IEEE Internet Things J. (2020).

[3] J.G. Panicker, A.S. Salehi, C. Rudolph, Authentication and access control in 5g device-to-device communication, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2021, pp. 1575–1582.

[4] Y. Zhuang, L.R. Sheets, Y.-W. Chen, Z.-Y. Shae, J.J. Tsai, C.-R. Shyu, A patient-centric health information exchange framework using blockchain technology, IEEE J. Biomed. Health Inform. 24 (8) (2020) 2169–2176.

[5] H.A. Maw, H. Xiao, B. Christianson, J.A. Malcolm, BTG-AC: Break-the-glass access control model for medical data in wireless sensor networks, IEEE J. Biomed. Health Inform. 20 (3) (2015) 763–774.

[6] A.S. Shahraki, C. Rudolph, M. Grobler, Attribute-based data access control for multi-authority system, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 1834–1841.

[7] B. Cremonezi, A.R. Gomes Filho, E.F. Silva, J.A.M. Nacif, A.B. Vieira, M. Nogueira, Improving the attribute retrieval on ABAC using opportunistic caches for fog-based IoT networks, Comput. Netw. (2022) 109000.

[8] P. García-Teodoro, J. Camacho, G. Maciá-Fernández, J. Gómez-Hernández, V. López-Marín, A novel zero-trust network access control scheme based on the security profile of devices and users, Comput. Netw. (2022) 109068.

[9] V.C. Hu, D. Ferraiolo, R. Kuhn, A.R. Friedman, A.J. Lang, M.M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., Guide to attribute based access control (ABAC) definition and considerations (draft), NIST Spec. Publ. 800 (162) (2013).

[10] C. Cotrini, T. Weghorn, D. Basin, Mining ABAC rules from sparse logs, in: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2018, pp. 31–46.

[11] L. Karimi, M. Aldairi, J. Joshi, M. Abdelhakim, An automatic attribute based access control policy extraction from access logs, IEEE Trans. Dependable Secure Comput. (2021).

[12] M.R. Rahman, S.S. Ahmad, C. Rudolph, Decentralized policy information points for multi-domain environments, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2021, pp. 1286–1293.

[13] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM, 2007, pp. 195–203.

[14] D. Servos, S.L. Osborn, HGABAC: Towards a formal model of hierarchical attribute-based access control, in: International Symposium on Foundations and Practice of Security, Springer, 2014, pp. 187–204.

[15] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, ACM, 2006, pp. 89–98.

[16] M.A. Islam, S. Madria, Attribute-based encryption scheme for secure multi-group data sharing in cloud, IEEE Trans. Serv. Comput. (2020).

[17] G. Wu, S. Wang, Z. Ning, B. Zhu, Privacy-preserved EMR information publishing and sharing: A blockchain-enabled smart healthcare system, IEEE J. Biomed. Health Inf. (2021).

[18] A.S. Shahraki, C. Rudolph, M. Grobler, A dynamic access control policy model for sharing of healthcare data in multiple domains, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), IEEE, 2019, pp. 618–625.

[19] V. Kuchta, G. Sharma, R.A. Sahu, O. Markowitch, Generic framework for attribute-based group signature, in: International Conference on Information Security Practice and Experience, Springer, 2017, pp. 814–834.
[20] X. Jin, R. Krishnan, R. Sandhu, A unified attribute-based access control model covering DAC, MAC and RBAC, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2012, pp. 41–55.
[21] C. Ngo, Y. Demchenko, C. de Laat, Multi-tenant attribute-based access control for cloud infrastructure services, J. Inf. Secur. Appl. 27 (2016) 65–84.
[22] M. Ghafoorian, D. Abbasinezhad-Mood, H. Shakeri, A thorough trust and reputation based RBAC model for secure data storage in the cloud, IEEE Trans. Parallel Distrib. Syst. 30 (4) (2018) 778–788.
[23] Y. Benkaouz, M. Erradi, B. Freisleben, Work in progress: K-nearest neighbors techniques for ABAC policies clustering, in: Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, ACM, 2016, pp. 72–75.
[24] P. Biswas, R. Sandhu, R. Krishnan, Attribute transformation for attribute-based access control, in: Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control, ACM, 2017, pp. 1–8.
[25] B. Farroha, D. Farroha, Challenges of 'operationalizing' dynamic system access control: Transitioning from ABAC to RAdAC, in: 2012 IEEE International Systems Conference SysCon 2012, IEEE, 2012, pp. 1–7.
[26] A.J. Rashidi, A. Rezakhani, A new approach to ranking attributes in attribute based access control using decision fusion, Neural Comput. Appl. 28 (1) (2017) 803–812.
[27] M. Amini, F. Osanloo, Purpose-based privacy preserving access control for secure service provision and composition, IEEE Trans. Serv. Comput. (2016).
[28] R. Ranchal, B. Bhargava, P. Angin, L. ben Othmane, Epics: A framework for enforcing security policies in composite web services, IEEE Trans. Serv. Comput. 12 (3) (2018) 415–428.
[29] A. Thakare, E. Lee, A. Kumar, V.B. Nikam, Y.-G. Kim, PARBAC: Priority-attribute-based RBAC model for Azure IoT cloud, IEEE Internet Things J. 7 (4) (2020) 2890–2900.
[30] H. Lv, J. Hillston, P. Piho, H. Wang, An attribute-based availability model for large scale IaaS clouds with CARMA, IEEE Trans. Parallel Distrib. Syst. 31 (3) (2019) 733–748.
[31] H. Wang, D. He, J. Han, VOD-ADAC: anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud, IEEE Trans. Serv. Comput. (2017).
[32] W. Ding, R. Hu, Z. Yan, X. Qian, R.H. Deng, L.T. Yang, M. Dong, An extended framework of privacy-preserving computation with flexible access control, IEEE Trans. Netw. Serv. Manag. (2019).
[33] H. Nasiraee, M. Ashouri, Privacy-preserving distributed data access control for CloudIoT, IEEE Trans. Dependable Secure Comput. (01) (2021) 1.
[34] K. Yang, X. Jia, K. Ren, B. Zhang, R. Xie, DAC-MACS: Effective data access control for multiauthority cloud storage systems, IEEE Trans. Inf. Forensics Secur. 8 (11) (2013) 1790–1801.
[35] B. Zhu, J. Sun, J. Qin, J. Ma, Fuzzy matching: multi-authority attribute searchable encryption without central authority, Soft Comput. 23 (2) (2019) 527–536.
[36] J. Hong, K. Xue, Y. Xue, W. Chen, D.S. Wei, N. Yu, P. Hong, TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud, IEEE Trans. Serv. Comput. (2017).
[37] J. Hong, K. Xue, N. Gai, D. Wei, P. Hong, Service outsourcing in F2C architecture with attribute-based anonymous access control and bounded service number, IEEE Trans. Dependable Secure Comput. (2018).
[38] H. Cui, R.H. Deng, J.K. Liu, X. Yi, Y. Li, Server-aided attribute-based signature with revocation for resource-constrained Industrial-Internet-of-Things devices, IEEE Trans. Ind. Inform. 14 (8) (2018) 3724–3732.
[39] G. Shanqing, Z. Yingpei, Attribute-based signature scheme, in: Information Security and Assurance, 2008. ISA 2008. International Conference on, IEEE, 2008, pp. 509–511.
[40] J. Yu, S. Liu, S. Wang, Y. Xiao, B. Yan, LH-ABSC: A lightweight hybrid attribute-based signcryption scheme for cloud-fog assisted IoT, IEEE Internet Things J. (2020).
[41] K. Fan, J. Wang, X. Wang, H. Li, Y. Yang, A secure and verifiable outsourced access control scheme in fog-cloud computing, Sensors 17 (7) (2017) 1695.
[42] S.T. Ali, B. Amberker, Short attribute-based group signature without random oracles with attribute anonymity, in: International Symposium on Security in Computing and Communication, Springer, 2013, pp. 223–235.
[43] E.F. Silva, D.C. Muchaluat-Saade, N.C. Fernandes, ACROSS: A generic framework for attribute-based access control with distributed policies for virtual organizations, Future Gener. Comput. Syst. 78 (2018) 1–17.
[44] M. Thimma, F. Liu, J. Lin, B. Luo, YHyXAC: Hybrid XML access control integrating view-based and query-rewriting approaches, IEEE Trans. Knowl. Data Eng. 27 (8) (2015) 2190–2202.
[45] A. Salehi, C. Rudolph, M. Grobler, A dynamic cross-domain access control model for collaborative healthcare application, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, 2019, pp. 643–648.
[46] Y. Rouselakis, B. Waters, Practical constructions and new proof methods for large universe attribute-based encryption, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, 2013, pp. 463–474.
[47] A. De Caro, V. Iovino, jPBC: Java pairing based cryptography, in: Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011, IEEE, Greece, 2011, pp. 850–855.
[48] J. Groth, A. Sahai, Efficient non-interactive proof systems for bilinear groups, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2008, pp. 415–432.
[49] D.J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang, High-speed high-security signatures, J. Cryptogr. Eng. 2 (2) (2012) 77–89.
[50] D.J. Bernstein, P. Birkner, M. Joye, T. Lange, C. Peters, Twisted edwards curves, in: International Conference on Cryptology in Africa, Springer, 2008, pp. 389–405.
[51] V. Shoup, A proposal for an ISO standard for public key encryption (version 2.1), IACR e-Print Arch. 112 (2001).
[52] B. Castle, Bouncy Castle crypto APIs, 2007, URL http://www.bouncycastle.org/, cited on page 82.
[53] W. Liu, CloudCrypto, 2019, URL https://github.com/liuweiran900217/CloudCrypto/tree/master/src/main/java/cn/edu/buaa/crypto/encryption/abe/cpabe/rw13.
[54] G.V. Laer, Groth-sahai, 2019, URL https://github.com/gijsvl/groth-sahai.
[55] L. Zanconato, Secrete, 2019, URL https://github.com/gherynos/secrete.
[56] Google, gson, 2019, URL https://github.com/google/gson.
[57] L.P. Deutsch, GZIP file format specification version 4.3, 1996.
[58] eXtensible Access Control Markup Language (XACML) Version 3.0, 2013, URL http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.
[59] OASIS eXtensible Access Control Markup Language (XACML) TC, 2020, URL https://www.oasis-open.org.
[60] WSO2 Identity server, 2020, URL https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html.
[61] Web on servlet stack (Spring web MVC), 2020, URL https://wso2.com/identity-and-access-management/.
[62] Mapping class inheritance hierarchies (SQLAlchemy), 2020, URL https://www.sqlalchemy.org/.
[63] AuthzForce server, 2019, URL https://github.com/authzforce/server0.
[64] A. Gabizon, Z.J. Williamson, O. Ciobotaru, PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge, IACR Cryptol. ePrint Arch. 2019 (2019) 953.
[65] M. Maller, S. Bowe, M. Kohlweiss, S. Meiklejohn, Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2111–2128.
[66] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, N. Ward, Marlin: Preprocessing zksnarks with universal and updatable SRS, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2020, pp. 738–768.
[67] S. Setty, Spartan: Efficient and general-purpose zksnarks without trusted setup, in: Annual International Cryptology Conference, Springer, 2020, pp. 704–737.
[68] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, M. Schofnegger, Poseidon: A new hash function for zero-knowledge proof systems, in: Proceedings of the 30th USENIX Security Symposium, USENIX Association, 2020.

Dr Ahmad Salehi Shahraki is a Lecturer of Cybersecurity in the Department of Computer Science and Information Technology (CSIT) at La Trobe University. Before joining La Trobe University, he worked as a Research Fellow at RMIT Blockchain Innovation Hub (BIH) and Centre for Cyber Security Research and Innovation (CCSRI). He received his M.Sc. degree in Information Security from the Faculty of Computing, UTM in 2013, M.Phil. Degree in Information Security from the Science and Engineering Faculty, QUT in 2017, and the Ph.D. degree in Cybersecurity from Faculty of IT (Cybersecurity Lab) at Monash University and the DSS Group at CSIRO's Data61 in 2020. He held a RA position from Cybersecurity LAB at Monash University in 2021. His research interests include Access Control, Blockchain, Cryptography, Cybersecurity, and Digital Health.

Dr Runchao Han is a senior research engineer at BabylonChain Inc. His research focuses on distributed system security, especially security and scalability issues in blockchains. He obtained this PhD degree at Monash University and CSIRO's Data61, Australia, the MSc degree from The University of Manchester, United Kingdom, and the bachelor's degree from Beijing University of Posts and Telecommunications, China.

Assoc Professor Carsten Rudolph is Professor for Cybersecurity and Deputy Dean of the Faculty of IT, Monash University, and Director for Research of the Oceania Cyber Security Centre OCSC in Melbourne, Australia. His research concentrates on information security, formal methods, cryptographic protocols, security of machine learning and human aspects of security with a strong focus interdisciplinary topics. He contributes to the development of secure solutions for different areas, such as digital health or future energy networks. Further, he drives scientific exchange between cybersecurity, law and organisational informatics. Another focus of his research is on nation-level cybersecurity maturity and policy development. Dr. Rudolph has established the OCSC and contributes to cybersecurity maturity reviews for nations in the Pacific region, a collaboration with Oxford University.

Dr Marthie Grobler is a Principal Research Scientist in human-centred cybersecurity at CSIRO's Data61 and is the Deputy Mission Lead of the Critical Infrastructure Protection and Resilience Mission. Marthie's research focus is on enhancing the usability of security solutions by considering human factors, with a strong interest in executive education, cybersecurity governance and critical infrastructure resilience and protection. Marthie spearheaded the establishment of the original human-centric security research team in CSIRO, which focuses on addressing the alignment and integration of human factors in the cyber domain to enhance security adoption and efficiency. Her research, management and consulting experience span multiple continents, national and state government departments, and a variety of domains linked with the digital domain.