Received: 30 January 2019 | Revised: 24 August 2019 | Accepted: 14 September 2019
DOI: 10.1002/cpe.5556

WILEY

RESEARCH ARTICLE

An efficient privacy-enhanced attribute-based access control mechanism

Yang Xu[1] | Quanrun Zeng[2] | Guojun Wang[3] | Cheng Zhang[2] | Ju Ren[2] | Yaoxue Zhang[2]

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[2]School of Computer Science and Engineering, Central South University, Changsha 410083, China
[3]School of Computer Science and Technology, Guangzhou University, Guangzhou 510006, China

Correspondence
Yang Xu, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China.
Email: xuyangcs@hnu.edu.cn

Present Address
School of Computer Science and Engineering, Central South University, Changsha 410083, China

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61702561 and 61632009; Guangdong Provincial Natural Science Foundation, Grant/Award Number: 2017A030308006

Summary

Owing to the rapid progress of network researching, attribute-based access control (ABAC) has attracted more and more attention due to its appreciable expressiveness, flexibility, and scalability. Unfortunately, collecting user attributes is necessary to complete the standard ABAC decision process, which increases the risk of privacy disclosure. This problem increases public doubts about ABAC and hinders its popularization. In this paper, a privacy-protected and efficient attribute-based access control (EPABAC) scheme is proposed to prevent the privacy leakage of access subject in the decision-making process of ABAC by introducing a novel hash-based binary search tree. The analyses and experimental evaluations show that the EPABAC achieves user privacy protection in the decision-making process with acceptable additional computing overhead.

KEYWORDS
attribute-based access control, binary search tree, digital signature, privacy, security

1 | INTRODUCTION

Recently, the rapid development of the latest network computing techniques has enabled billions of users to access online resources and services more conveniently than ever before. On the one hand, the new computing frameworks such as mobile cloud computing,[1] edge computing,[2] transparent computing,[3-5] and underlying 4G/5G networks bring unprecedented convenience and freedom to users. However, on the other hand, those innovations lead to more complicated scenes, which are significantly dynamic, distribute, fine-grained and changeable, and bring new challenges to the access control technology.[6,7] In these cases, like the earlier classical models such as discretionary access control (DAC)[8] and mandatory access control (MAC),[9] most typical access control models, including Role-Based Access Control (RBAC),[10] are gradually becoming unadaptable because of the extensive decision-making mechanism and the lack of flexible, fine-grained, and dynamic authorization.

Owing to the development of access control technology, the attribute-based access control (ABAC) model,[11] which diversely considers many security factors such as subject, object, and environment state, has emerged. By introducing the policy-based access control management mechanism, ABAC achieves stronger expressive ability, flexibility, and scalability than classical access control schemes, which lead to better adaptability for dynamic modern network service scenarios.[12-15] As a promising technique, the ABAC method has been applied in many frontier fields, including cloud computing,[16-18] big data,[19-23] and Internet of Things (IoT)[24,25] scenarios and has been developed into a mature business solution.[26,27]

However, even the ABAC model suffers the potential risk of privacy leakage. Due to the importance of disclosing attributes value involved in an initiated access request in the classical ABAC decision process, the unprotected attributes, especially static users attributes which may imply

https://doi.org/10.1002/cpe.5556

sensitive information, will bring the potential threat of privacy disclosure. Many researchers have already pointed out this defect and believe that it limits the application scenarios of ABAC and thus negatively affects the further development of ABAC technology.[28,29]

In order to eliminate the risk of privacy disclosure in standard ABAC model, some tentative exploratory schemes have been proposed.[30-36] Unfortunately, some schemes that attempt to modify the normal decision-making process incur excessive computational overhead, which may result in performance bottlenecks and limit their usability. Others achieve private-maintaining by introducing online third parties, which have severely limited the usability, thereby limiting its application scenarios.

In this paper, we present an access control mechanism supporting privacy attribute protection for above problems, which supports privacy access decision making without leaking user sensitive attributes. Firstly, a hash-based binary search tree structure and the corresponding generator are proposed with the one-way property of hash function, which guarantees that the holder of a specific generator can only get information about the corresponding leaf nodes in the tree, and we introduce the above structure into the ABAC model and transform the access control decision-making process into a challenge based on the generators of specific hash-based binary search tree. Secondly, according to the properties of attributes in ABAC model, we propose a method to transform access control rules and attributes into challenges and generators in a specific hash-based binary search tree, respectively. Finally, we examine the performance and security of our mechanism, and the results show that the mechanism can resist user deception attacks while achieving privacy access decision-making process and it has acceptable performance. The major contributions of our work are summarized as follows.

- With the one-way property of hash functions, we propose a data structure and corresponding scheme based on the binary tree structure to support the decision-making process in ABAC model, which ensures that the validated person can only answer challenges that happen to fall within his or her authority.
- We introduce the hash binary search tree into the standard ABAC authorization phase and decision-making phrase, which enables correct decision making without exposing the value of static user attribute or cheated by users.
- By introducing asymmetric encryption and digital signature, we emancipate the decision-making process in our privacy-enhanced scheme from the participation of online third party.
- We analyzed the EPABAC model theoretically for its effectiveness and efficiency and evaluate its effectiveness and efficiency by experiments to demonstrate our enhancements in privacy protection within an acceptable overhead.

The remainder of this paper is organized as follows. In Section 2, some articles related to our work are discussed, mainly about the solution of privacy protection in ABAC. The structure and workflow are detailed in Section 3, along with a case study. In Section 4, the effectiveness and efficiency of the model are discussed through analyses and experiments. Finally, we summarize our work and discuss possible improvements in Section 5.

2 | RELATED WORK

As the most immediate solution to protecting network resources from illegal accesses, which prevents potential security violations, the access control technology attracted considerable attention of researchers. Many famous access control schemes were proposed in the past decades.[8-10] As a novel and efficient solution, the attribute-based access control (ABAC)[11] model considers many security factors into its decision process, including the corresponding information of the user, resource, or access environment, which has become the next generation of the mainstream access control mechanism[12-15] and has attracted a lot of research and has been applied in many new network environments.[16,17,19-21,24,25] In addition, standardization organizations[11] and enterprises[26,27] have made more practical efforts to further promote the development and promotion of ABAC technology.

However, the lack of consideration for user privacy limits the development of ABAC. The standard ABAC scheme inevitably involves the disclosure of user attributes containing sensitive information, which causes obvious privacy risks, which reduces the public trust of ABAC, thus affecting the development and promotion of ABAC.[28]

In order to alleviate the privacy problem in ABAC model, some exploratory improving schemes have been proposed. Wu and Gao[31] proposed a scheme that extended the standard ABAC model to enable the attribute disclosure restriction, in which the subject is able to decide which attributes could be disclosed and which service providers can collect such attributes. Similarly, Park and Chung[32] proposed an attribute exposure scheme to expose a minimum set of user attributes. In their work, users submit only the necessary attributes in an access request, for the sake of the principle of least privilege. Zhang et al[33] introduced the trust mechanism into ABAC to prevent the leakage of privacy attributes. In their scheme, whether to expose users' attributes is decided by evaluating the sensitivity of attributes and the trustworthiness of service providers. Though the above schemes can alleviate the privacy crisis in access control to some extent, those solutions modified the attribute submission process and may interfere with some necessary attribute submissions, which unnecessarily decrease the usability of the whole system.

In addition, Esmaeeli and Shahriari[34] presented an ABAC solution for distributed environments. In the solution, the attributes of access subject are checked through the home organization of the subject to prevent the malicious access to users' privacy from the server side. Besides, Kolter et al[35] introduced a privacy protection mechanism into the ABAC framework, in which decision points are separated from service providers and users can choose a policy decision point which is authorized by a trusted third party for the decision-making process. Put and De Decker[36] proposed a privacy-oriented online service ABAC mechanism. Instead of the service provider's direct contact with user attributes, a privacy-friendly protocol was designed for collecting attributes of access requester from a trusted online third party to ensure that the service providers cannot learn accurate values of the user's attribute and eliminate the relevance of information. However, these methods have great limitations due to the dependence on the online trusted third parties, which are inappropriate in dynamic and distributed environments. In addition, the communication with online third parties has an impact on the performance of the whole system.

In summary, considering the deficiencies of existing ABAC solutions, there is an urgent need for an efficient and complex privacy enhancement ABAC scheme without the support of a trusted online third party.

3 | EPABAC

In this section, we first introduce the architecture of EPABAC, then we will describe the designation and properties of a hash-based binary search tree, followed by the detailed description of the authorization phase and decision-making phase in EPABAC. Finally, a case study is studied in detail for the ease of understanding.

3.1 | EPABAC model

3.1.1 | Notations

To simplify the description, we use the following notations in Table 1.

3.1.2 | Framework and workflow

The EPABAC model extents the authorization and decision mechanism of attributes on the standard ABAC model to enable the protection of privacy attributes. As represented in Figure 1, in the authorization phase, for each sensitive attribute, the attribute authority, which is an authority which assigns privileges by issuing attribute certificates, will send a set of evidence generators to the client according to the corresponding value of the client, while the nonsensitive attributes are processed like in the classical ABAC solution (steps 1 and 2). Thereafter, in the decision-making phase, when an access request is initiated by a client, the service provider will first collect all involved nonsensitive attribute, which will then be processed by the Standard ABAC matching component (step 3). Subsequently, for each sensitive attribute, the service provider will require the user to provide the evidence of meeting the policies, which will be verified in the privacy module. The requested evidence is independent of the subject's attribute value and can be computed with the evidence generator (step 4). Finally, the service provider will synthesize the decision results for privacy and nonprivacy attributes, and implement access control through enforcement facility (step 5).

TABLE 1 The notations and definitions

Notations	Definitions
$Gen_t(a, b)$	The generator of the *a*th to *b*th leaf nodes in hash-based binary tree *t*.
k_{pub}	The public key held by the service provider.
k_{pri}	The private key held by the service provider.
k_{sig}	The private key held by the attribute authority for digital signatures.
k_{ver}	The public key held by the attribute authority for signature verification.
$E_k(x)$	Encrypting the plaintext *x* with key *k*.
$D_k(x)$	Decrypting the ciphertext *x* with key *k*.
$Root(t)$	The root of a hash-based binary tree *t*.
$Left(p)$	The left child of a node *p*.
$Right(p)$	The right child of a node *p*.
$Sub_t(p)$	The set of all nodes in the subtree with node *p* as its root in hash-based binary tree *t*.
$Leaf_t(p)$	The set of all leaf nodes in the subtree with node *p* as its root in hash-based binary tree *t*.
$\lambda_t(a, b)$	The set of the *a*th to *b*th leaf nodes in hash-based binary tree *t*.
n	The height of a hash-based binary tree.

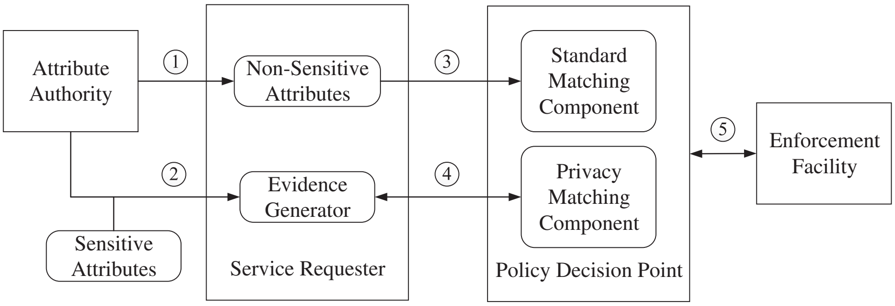

FIGURE 1 The workflow of EPABAC

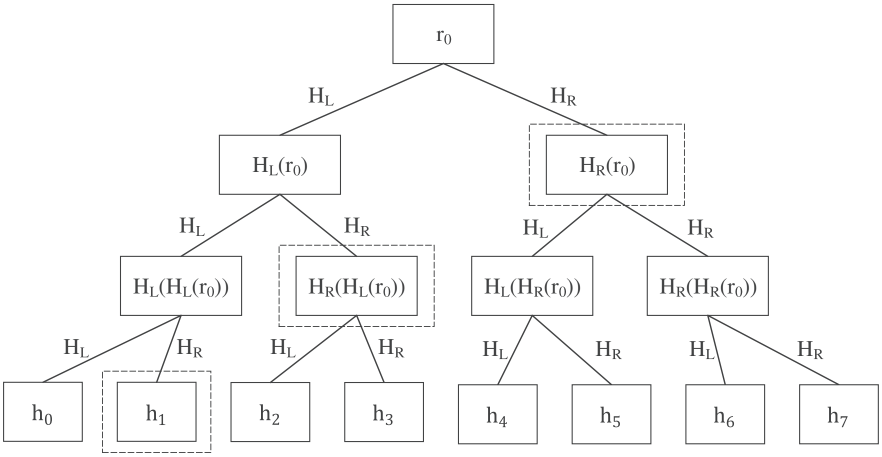

FIGURE 2 A hash-based binary search tree and corresponding largest generator

3.2 | Hash-based binary search tree

To achieve the purpose of privacy decision, a hash-based binary search tree scheme is introduced. As represented in Figure 2, the hash-based binary search tree is a type of the perfect binary tree structure with node values, which is uniquely determined by the root node value v_0 and two hash function H_L, H_R. In the hash-based binary search tree, the value of each left child node is $H_L(v_p)$, where v_p represents the value of the corresponding father node. Similarly, the value of each right child node is $H_R(v_p)$.

Obviously, given the values of some nodes in the hash-based binary tree and hash function H_L, H_R in such a tree, the values of corresponding leaf nodes can be deduced. In the following, a set of such nodes will be referred to as a generator of the corresponding sequence of leaf nodes.

According to the properties of hash function, given the specific v_r, H_R, H_L, and tree height n, the corresponding generated perfect binary tree, in which all interior nodes have two children and all leaves have the same level, satisfies the following properties.

- Within the knowledge of the value of a node p, the value of any node in its subtree can be calculated through H_L and H_R. However, it is difficult to deduce the value of any ancestor node by the value of a node p.
- For any $n \geq 2$, at most $n - 1$ nodes are needed to represent any continuous leaf sequence ending with the 2^n-th leaf node or starting with the 1st leaf node. (As an example, Figure 2 shows a case of the generator of continuous leaf sequence h_1 to h_8 with the most nodes, which needs three nodes for representing. No other continuous leaf sequence needs more than three nodes to represent.)
- Starting with a root node value, any generator corresponding to the sequence ending with the 2^n-th leaf node or starting with the 1st leaf node can be obtained by hashing no more than $2 \log 2^n$ times.

In addition, we give the definition of generator $Gen_t(a, b)$. Given a hash-based binary search tree t and $0 \leq a \leq b < 2^n$, the generator $Gen_t(a, b)$ is a set of node in t which has the following properties:

- $\forall p \in \lambda_t(a, b)$, there exists a node $p' \in Gen_t(a, b)$ satisfies that $p \in Sub_t(p')$.
- $\forall p \in Gen_t(a, b)$, we have $Leaf(p) \subset \lambda_t(a, b)$.
- $\forall p \in Gen_t(a, b)$, $(Sub_t(p) - \{p\}) \cap Gen_t(a, b) = \emptyset$, ie, the descendant nodes of a node $p \in Gen_t(a, b)$ will not be included in the set $Gen_t(a, b)$.

With the help of hash-based binary tree and the corresponding generators, we will propose our privacy-preserving access control decision scheme in the following sections.

3.3 | Authorization Phase

As a substitution to the usual attribute authorization process, for each privacy attribute, the attribute authority, which plays the role of granting user attributes, will create a set of generators within the hash search tree as follows.

Suppose the corresponding attribute A is in the integer field with a range of values (l_{min}, l_{max}), the attribute authority will first randomly select two random numbers r_1, r_2 and two hash functions H_L, H_R, in which r_1, r_2 are information revealed only to the attribute authority and the service provider while H_L, H_R are public.

After that, the attribute authority will create two hash-based binary search trees T_0, T_1, which are called less-than tree and great-than tree, respectively, with the root node value r_1, r_2 as shown in Figure 3, and figure out the evidence generator g as follows:

$$g = Gen_{T_0}(v - l_{min}, l_{max} - l_{min}) \cup Gen_{T_1}(0, v - l_{min}). \tag{1}$$

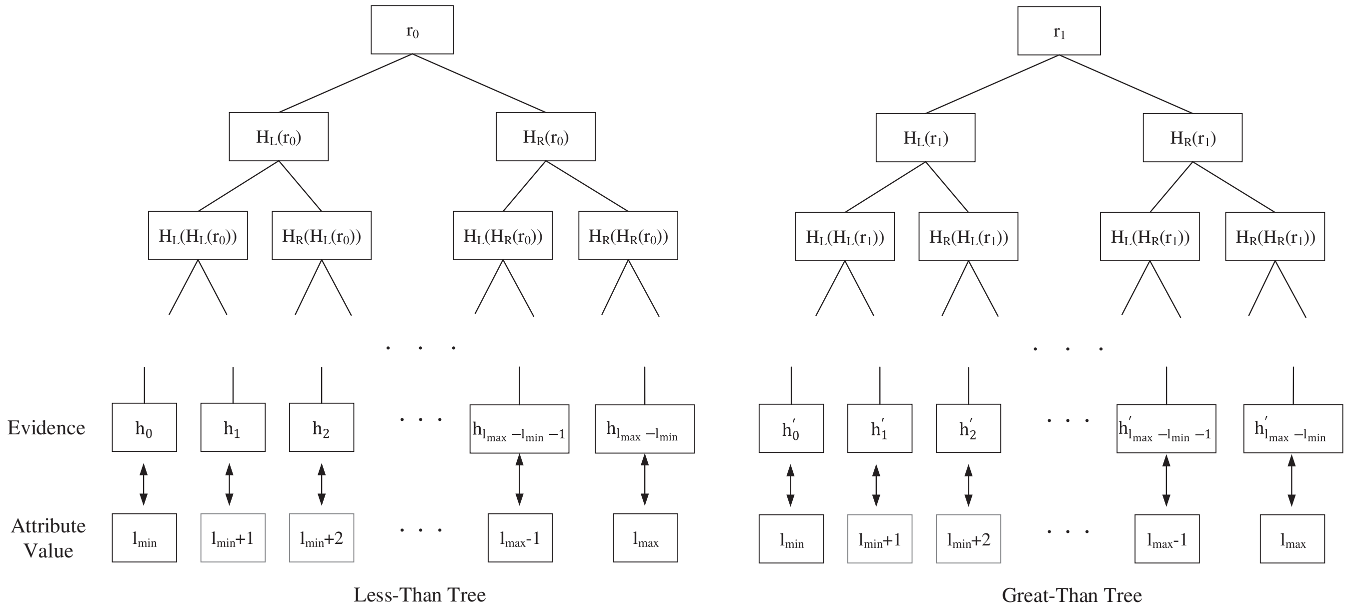

FIGURE 3 The less-than tree and great-than tree based on hash-based binary search tree

In the above formula, v is the corresponding value of user attributes and the $Gen_{T_0}(v - l_{min}, l_{max} - l_{min})$, and $Gen_{T_1}(0, v - l_{min})$ are two generators which can be calculated through Algorithm 1.

```
Algorithm 1 The process of calculating the generator $Gen_t(a, b)$
Require: A tree $t$ and the range of leaf node set $\lambda_t(a, b)$;
Ensure: The generator $Gen_t(a, b)$ consisting of a set of nodes in tree $t$;
$S_1 := root(t), S_2 := \emptyset$
while $S_1$ is not empty do
    $p := S_1.pop()$
    if $Leaf(p) \nsubseteq \lambda_t(a, b)$ then
        $S_1.push(Left(p), Right(p))$
    else
        $S_2.push(p)$
    end if
end while
return $S_2$;
```

Considering the properties of the binary search tree, the attribute authority actually only needs to send a small number of node values, and the user can calculate the corresponding result through H_L, H_R.

Subsequently, the attribute authority encrypts r_0, r_1 with the public key k_{pub} of service provider as follows:

$$\begin{cases} d_0 = E_{k_{pub}}(r_0) \\ d_1 = E_{k_{pub}}(r_1). \end{cases} \tag{2}$$

Then, the encrypted result $E_{k_{pub}}(r_0), E_{k_{pub}}(r_1)$ is digitally signed to get the corresponding digital signatures s_0, s_1 as follows:

$$\begin{cases} s_0 = E_{k_{sig}}(d_0) \\ s_1 = E_{k_{sig}}(d_1). \end{cases} \tag{3}$$

Finally, the attribute authority send the assemblage $\{g, d_0, d_1, s_0, s_1\}$ to the user.

3.4 | Decision-making phase

In the decision-making phase, for each policy clause involving privacy attributes, the service provider will first convert it into several expressions, which are in the form of $Attr \leq a$ or $Attr \geq a$, where $Attr$ and a represent the attribute value of requester and the constant in the clause, respectively.

Subsequently, for each expression in the form of $Attr < a$, the user first sends the encrypted root node value d_0, d_1 and corresponding signatures s_0, s_1 to the service provider. Then, the service provider decrypts s_0, s_1 using the public key k_{ver} of attribute authority to verify whether the following formula is true, thereby to check the correctness of d_0, d_1:

$$(D_{k_{ver}}(s_0) = d_0) \wedge (D_{k_{ver}}(s_0) = d_1) \tag{4}$$

After finishing the validation of correctness of d_0, d_1, the service provider calculates r_0, r_1 by the following formula:

$$\begin{cases} r_0 = D_{k_{pri}(d_0)} \\ r_1 = D_{k_{pri}(d_1)}. \end{cases} \tag{5}$$

Thereafter, the service requester will be required to offer the evidence $h_{a-l_{min}}$ in the corresponding less-than tree. At the same time, the service requester will calculate $h_{a-l_{min}}$ itself with r_0, r_1 and verify user's results. According to the distribution rules of generators, the generator held by users can only figure out the value of $(v - l_{min})$ to $lmax$ leaf nodes in the less-than tree. Thus the service requester can figure out the $h_{a-l_{min}}$ correctly if and only if the user's attribute value $Attr$ satisfies that $Attr < a$, which allows the service provider to learn whether the policies match the user.

Similarly, for $Attr > a$ judgment, the service requester will ask the service requester to provide the value $h'_{a-l_{max}}$ in great-than tree.

Finally, the service provider will synthesize the decision results for sensitive and nonsensitive attributes and send the final results to the access execution point to perform access control.

3.5 | Case study

For better understand our scheme, a case study will be described in detail in this article. Suppose we have a EPABAC system with a single involved attribute $Attr \in \{Guest, Engineer, Supervisor, Manager\}$. Firstly, the system maps attributes to integers in the range of $[0, 3]$ through the following map function F:

$$F(a) = \begin{cases} 0, a = Guest \\ 1, a = Engineer \\ 2, a = Supervisor \\ 3, a = Manager \end{cases} \quad (a \in Attr). \tag{6}$$

Thus, operations on $Attr$ will be converted to numeric operation. Considering a user whose job position is "*Engineer*," through the above mapping, his attribute value is mapped to 1. The attribute authority will figure out the evidence generator for the user, which will ensure that he can generate all less-than evidence for all $F(Attr) \geq 1$ and great-than evidence for $F(Attr) \leq 1$.

As represented in Figure 4, attribute authority will randomly select the number r_0, r_1, function H_L, H_R and then calculate the evidence generator $Gen_{T_0}(1, 3) = \{n_{1,1}, h_1\}$ and $Gen_{T_1}(0, 1) = \{n'_{1,0}\}$ accordingly, which will then be sent to the user along with the validation data d_0, d_1, s_0, s_1. Thus, the user can calculate evidence $\{h_1, h_2, h_3\}$ in the less-than tree and $\{h'_0, h'_1\}$ in the great-than tree. The selected node values should ensure that users can calculate the values of all leaf nodes in the interval $[1, 3]$ for in the less-than tree and in the interval $[0, 1]$ for the great-than tree.

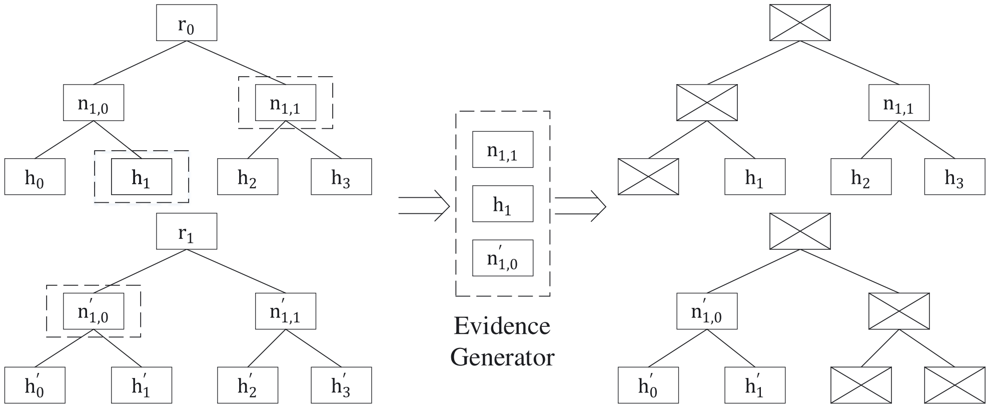

FIGURE 4 The principle of evidence generator

Then, consider the case when the user initiates an access request. Suppose there is a policy as follows:

IF *Attr* is *Engineer* or *Supervisor* **THEN** grant.

Based on the map function F, the service provider first converts that policy into the following logical function G:

$$\begin{aligned} G(Attr) &= G_0(Attr) \wedge G_1(Attr) \\ &= (\,F(Attr) \geq 1) \wedge ((\,F(Attr) \leq 2)), \end{aligned}$$

in which the $G_0(Attr) = F(Attr) \leq 2$ and $G_1(Attr) = F(Attr) \geq 1$ are two sub-function of G. For two sub-functions G_0 and G_1, the service provider will ask the user to calculate h_2 and h'_1, respectively. According to the generator owned by the user, the user calculates the following expression and sends the obtained h_2, h'_1 to the server:

$$\begin{cases} h_2 = H_L(n_{1,1}) \\ h'_1 = H_R(n'_{1,0}). \end{cases} \tag{1}$$

As validation, the service provider will calculate the h_2, h'_1 according to r_0, r_1 as follows:

$$\begin{cases} h_2 = H_L(H_R(r_0)) \\ h'_1 = H_R(H_L(r_1)) \end{cases}. \tag{8}$$

Finally, service providers will compare it with the return value h_2, h'_1 from user and complete the decision-making process. Since the user figured out the h_2, h'_1 correctly, the access request will be granted.

For continuous attributes, its processing depends on how a particular system stores continuous data. For example, in a computer, floating-point data are commonly presented as discrete values whose representation accuracy is limited by the accuracy of floating-point numbers. With the relative error as the minimum granularity, the comparison of floating-point numbers can be transformed into the comparison of fixed-point numbers in the discrete number field so that the attributes based on floating-point data can be transformed into similar processing as above.

4 | ANALYSIS AND EXPERIMENT COMPARISON

4.1 | Effectiveness

Supposing that the user has an attribute $Attr(Attr \in [l_{min}, l_{max}] \cap \mathbb{Z})$ and the corresponding value is v, as described above, the generators owned by user can deduce the evidence of matching, ie, the value of leaf nodes in hash-based binary search tree, for all policies in form of $Attr < a(a \in [v, l_{max}])$ and $Attr < a(a \in [v, l_{max}])$. The scheme guarantees that only users who satisfy the policies can provide correct evidence.

To prove this, we assume that there is an attribute value $v' \geq v$ and the corresponding access control rule $Attr > v'$, and the user has the ability to provide evidence to satisfy the access control rule, ie, the user can figure out the value of $h_{v'}$ in tree T_1. According to the properties of the hash-based binary search tree, there exists a node p in the generator $Gen_{T_0}(v - l_{min}, l_{max} - l_{min})$, which satisfies

$$h_{v'} \in Leaf(p). \tag{9}$$

Considering that $h_{v'} \notin \lambda_{T_0}(v - l_{min}, l_{max} - l_{min})$, we have

$$Leaf(p) \not\subseteq \lambda_{T_0}(v - l_{min}, l_{max} - l_{min}). \tag{10}$$

From Algorithm 1, it can be inferred that $p \notin Gen_{T_0}(v - l_{min}, l_{max} - l_{min})$, which contradicts the above assumptions. Therefore, for policies in the form like $Attr > a$, only the users who meet the attribute value requirements can provide corresponding evidence. Similarly, the above conclusions are also valid for rules in the form like $Attr < a$.

Besides, in the privacy decision process, the evidence that the service provider requires the user to provide is independent of the specific value of the user, ie, no matter what the attribute value is, the evidence that the user is required to send is a constant, which is only relevant to the policy itself. Therefore, service providers cannot learn the specific values of user attributes from the decision-making process. In conclusion, the access decision-making process for privacy attributes will not result in user privacy exposure.

TABLE 2 The experiment configuration and the corresponding results

	Group 1	Group 2	Group 3
Scheme	ABAC	EPABAC	EPABAC
Range of attribute	$[-2^{31}, 2^{31})$	$[-2^{31}, 2^{31})$	$[-2^{63}, 2^{63})$
CPU	Intel Core i7-7700HQ 2.80GHz		
RAM	15.6 GB		
Operating system	Microsoft Windows 10 Education (x64)		
Average time	0.037 ms	0.136 ms	0.251 ms

4.2 | Efficiency

Considering the computing overhead of authorization phase. Supposing that each attribute in the EPABAC system has no more than n possible values, at most $\log_2 n$ node will be selected as the evidence generator in each hash-based binary search tree (for attribute authority, at most $2\log_2 n$ hash calculations are needed to get the values of these nodes), ie, the calculation of whole evidence generator will run in $O(\log n)$, and the user needs $O(\log n)$ storage space to store any received evidence generator.

When it comes to the decision-making phase, for any attributes involved, service providers and users both need no more than $\log n$ calculating to figure out the corresponding evidence. Thus, the decision-making process will runs in $O(\log n)$.

Considering that the overhead of attribute comparison in the access decision-making phase only accounts for a very small proportion of the whole process of access control decision-making, the additional cost brought by this scheme is acceptable. In summary, the scheme in this paper can implement the access control process of privacy protection within a relatively reasonable cost.

4.3 | Experiment comparison

In order to analyze the performance of EPABAC, we implemented the attribute distribution and evidence generation algorithm of our scheme with MATLAB R2016b and conducted a comparative experiment between EPABAC and ABAC. The experiment is divided into three groups, each group contains 10 000 access requests with 10 subject attributes for each request, and the average time of decision-making process will be counted. To ensure the correctness of the EPABAC scheme, group 1 and group 2 share the same data set. The detailed system configuration and result are shown in Table 2.

As shown in the result, Comparing the experimental group 1 and 2, the decision-making overhead of EPABAC model is acceptable on the premise that the decision-making results of both sides are identical. Considering that the computation overhead of access control decision-making is negligible in the whole access control process, the EPABAC scheme obviously does not impose too much additional performance burden. Besides, compared with the time overhead caused by communication in access control, the time delay caused by this scheme is also negligible. In addition, we note that the experimental results do not fully accordance with the time complexity analysis, which may be due to the fact that the core execution process occupies only a small proportion of the whole decision-making process.

5 | CONCLUSION

The security of rapid developing computing models depends on appropriate access control mechanisms. Therefore, the ABAC model has attracted much attention due to its good dynamic, flexibility, and scalability. However, the ABAC model requires disclosure of personal attributes, which may conflict with customers' privacy preferences.

In this paper, we proposed an EPABAC model, which is able to prevent Privacy Leakage Threat in the basic ABAC model without the introduction of a trusted third party. In the EPABAC model, all sensitive attributes are protected by introducing a novel hash-based binary search tree, thus preventing the risk of privacy disclosure. Moreover, the server side can still process the consistent access requests correctly through the hash-based binary search tree, without learning privacy information. The workflow of EPABAC scheme is illustrated by a case study. The theoretical analysis further proves the effectiveness and efficiency of the scheme.

Future work will involve with automatically identifying the sensitive attributes needing protection within the introduction of deep learning technique, improving the efficiency of our scheme and providing adequate privacy protection. In addition, we also plan to explore support for attributes on continuous domains.

DISCLOSURE

This work was presented in part at the SpaCCS 2018, Melbourne, Australia, December 11-13, 2018.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under grants 61702561 and 61632009 and the Guangdong Provincial Natural Science Foundation under grant 2017A030308006.

ORCID

Yang Xu https://orcid.org/0000-0002-3194-8369
Guojun Wang https://orcid.org/0000-0001-9875-4182

REFERENCES

1. Fernando N, Loke S, Rahayu W. Mobile cloud computing: a survey. *Futur Gener Comput Syst*. 2013;29:84-106.
2. Mao Y, You C, Zhang J, Huang K, Letaief KB. A survey on mobile edge computing: the communication perspective. *IEEE Commun Surv Tutor*. 2017;19(4):2322-2358.
3. Zhang Y, Guo K, Ren J, Zhou Y, Wang J, Chen J. Transparent computing: a promising network computing paradigm. *Comput Sci Eng*. 2017;19(1):7-20.
4. Xu Y, Wang G, Yang J, Ren J, Zhang Y, Zhang C. Towards secure network computing services for lightweight clients using blockchain. *Wireless Communications and Mobile Computing*. 2018;2018:1-12.
5. Xu Y, Ren J, Wang G, Zhang C, Yang J, Zhang Y. A blockchain-based nonrepudiation network computing service scheme for industrial IoT. *IEEE Trans Ind Inf*. 2019;15(6):3632-3641.
6. Xu Y, Zeng Q, Wang G, Zhang C, Ren J, Zhang Y. A privacy-preserving attribute-based access control scheme. Paper presented at: International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage; 2018; Melbourne, Australia.
7. Xu Y, Wang G, Ren J, Zhang Y. An adaptive and configurable protection framework against android privilege escalation threats. *Futur Gener Comput Syst*. 2019;92:210-224.
8. Li N. Discretionary access control. In: van Tilborg HCA, Jajodia S, eds. *Encyclopedia of Cryptography and Security.*. Boston, MA: Springer; 2011.
9. Lindqvist H. Mandatory Access Control [master's thesis]. Umeå, Sweden: Umeå University; 2006.
10. Sandhu R, Ferraiolo D, Kuhn R. The NIST model for role-based access control: Towards a unified standard. Paper presented at: ACM Workshop on Role-Based Access Control; 2000; Berlin, Germany.
11. Hu VC, Ferraiolo D, Kuhn R, et al. Guide to Attribute Based Access Control (ABAC) Definition and Considerations (draft). NIST Special Publication 800-162. 2013.
12. Servos D, Osborn SL. Current research and open problems in attribute-based access control. *ACM Comput Surv*. 2017;49(4):1-65.
13. Luo E, Liu Q, Wang G. Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks. *IEEE Commun Lett*. 2016;20(9):1772-1775.
14. Xu Y, Gao W, Zeng Q, Wang G, Ren J, Zhang Y. FABAC: A flexible fuzzy attribute-based access control mechanism. Paper presented at: International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage; 2017; Guangzhou, China.
15. Xu Y, Gao W, Zeng Q, Wang G, Ren J, Zhang Y. A feasible fuzzy-extended attribute-based access control technique. *Secur Commun Netw*. 2018;2018:1-11.
16. Jin X. Attribute-Based Access Control Models and Implementation in Cloud Infrastructure as a Service [PhD dissertation]. San Antonio, TX: The University of Texas at San Antonio; 2014.
17. Qiu M, Gai K, Thuraisingham B, Tao L, Zhao H. Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry. *Futur Gener Comput Syst*. 2018;80:421-429.
18. Li Y, Yu H, Song B, Chen J. Image encryption based on a single-round dictionary and chaotic sequences in cloud computing. *Concurrency Computat Pract Exper*. 2019:1-15.
19. Gupta M, Patwa F, Sandhu R. An attribute-based access control model for secure big data processing in Hadoop ecosystem. Paper presented at: 3rd ACM Workshop on Attribute-Based Access Control; 2018; Tempe, AZ.
20. Cavoukian A, Chibba M, Williamson G, Ferguson A. The Importance of ABAC: Attribute-Based Access Control to Big Data: Privacy and Context. Toronto, Canada: Ryerson University; 2015.
21. Liu C, Beaugeard N, Yang C, Zhang X, Chen J. HKE-BC: hierarchical key exchange for secure scheduling and auditing of big data in cloud computing. *Concurrency Computat Pract Exper*. 2016;28(3):646-660.
22. Cui Z, Xue F, Cai X, Cao Y, Wang G, Chen J. Detection of malicious code variants based on deep learning. *IEEE Trans Ind Inf*. 2018;14(7):3187-3196.
23. Cui Z, Cao Y, Cai X, Cai J, Chen J. Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things. *J Parallel Distrib Comput*. 2019;132:217-229.
24. Sciancalepore S, Pilc M, Schröder S, et al. Attribute-based access control scheme in federated IoT platforms. Paper presented at: International Workshop on Interoperability and Open-Source Solutions; 2016; Stuttgart, Germany.
25. Monir S. A Lightweight Attribute-Based Access Control System for IoT [PhD dissertation]. Saskatoon, Canada: University of Saskatchewan; 2016.
26. Axiomatics. https://www.axiomatics.com. Accessed September 5, 2018.
27. Nextlabs. https://www.nextlabs.com. Accessed June 17, 2018.
28. Irwin K, Yu T. Preventing attribute information leakage in automated trust negotiation. Paper presented at: 12th ACM Conference on Computer and Communications Security; 2005; Alexandria, VA.
29. Zhang Q, Liu Q, Wang G. PRMS: A personalized mobile search over encrypted outsourced data. *IEEE Access*. 2018;6:31541-31552.
30. Peng T, Liu Q, Wang G. A multilevel access control scheme for data security in transparent computing. *Comput Sci Eng*. 2016;19(1):46-53.
31. Wu K, Gao H. Attribute-based access control for web service with requester's attribute privacy protected. Paper presented at: International Conference on Informational Technology and Environmental; 2008; Damascus, Syria.
32. Park SM, Chung SM. Privacy-preserving attribute-based access control for grid computing. *Int J Grid Util Comput*. 2014;5(4):286-296.
33. Zhang G, Liu J, Liu J. Protecting sensitive attributes in attribute based access control. Paper presented at: International Conference on Service-Oriented Computing; 2012:294-305.

34. Esmaeeli A, Shahriari HR. Privacy protection of grid service requesters through distributed attribute based access control model. Paper presented at: International Conference on Grid and Pervasive Computing; 2010; Hualien, Taiwan.

35. Kolter J, Schillinger R, Pernul G. A privacy-enhanced attribute-based access control system. Paper presented at: IFIP Annual Conference on Data and Applications Security and Privacy; 2007; Redondo Beach, CA.

36. Put A, De Decker B. Attribute-based privacy-friendly access control with context. Paper presented at: International Conference on E-Business and Telecommunications; 2016; Lisbon, Portugal.

How to cite this article: Xu Y, Zeng Q, Wang G, Zhang C, Ren J, Zhang Y. An efficient privacy-enhanced attribute-based access control mechanism. *Concurrency Computat Pract Exper*. 2019;e5556. https://doi.org/10.1002/cpe.5556