Enhancing Cross-Device Security with Fine-Grained Permission Control

Han Hu[1], Daibin Wang[2], Tailiang Hong[2], and Sheng Zhang[1(✉)]

[1] Key Laboratory of Advanced Sensor and Integrated System, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China
{hu-h21,zhang_sh}@mails.tsinghua.edu.cn
[2] Huawei Technologies Co., Ltd., Shenzhen, China

Abstract. With the proliferation of smart devices in personal and home environments, there is a growing need for cross-device interaction. However, distributed scenarios that cross device boundaries pose unique security and privacy challenges. While existing cross-device security mechanisms focus primarily on authentication, there is little research on fine-grained permission control. Permission models, which are critical security mechanisms for single devices, do not adequately support cross-device access control. To address this gap, we proposed and implemented a distributed role and attribute hybrid-based access control (DHBAC) model to enhance the security of cross-device access. DHBAC extends the single-device permission system to cross-device access control, providing fine-grained control based on users, devices, and applications. This approach effectively eliminates the over-authorization problem and supports the principle of least privilege. In addition, DHBAC can dynamically adjust and assign permissions based on specific scenarios and user requirements, improving the flexibility and adaptability of the system. To evaluate DHBAC, we deployed it on Harmony Operating System and tested it in several real-world, cross-device scenarios. Our evaluation shows that DHBAC effectively blocked malicious cross-device access and mitigated the associated security risks with acceptable system overhead.

Keywords: Mobile device · Access control · Permission · Cross device · Operating system

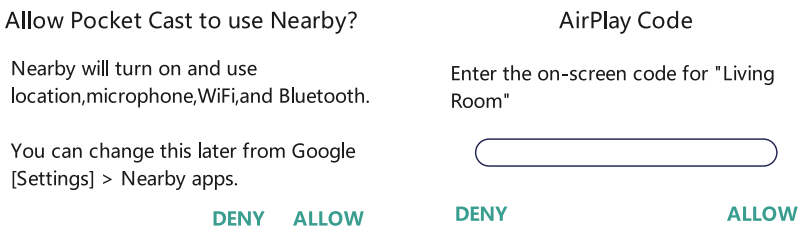
1 Introduction

In recent years, there has been a significant increase in the number of smart devices owned by individuals and households worldwide. According to Ipsos [1], the average number of smart devices per household worldwide will be around 9 in 2021 and is expected to reach 14 by 2025. An ecosystem of multiple devices adds value to each other. For example, viewers can rent a movie on TV by using their phone to enter payment information; cyclists can share their bike route with others they are riding with. The popularity of cross-device access has provided greater convenience for users [18], but it has also posed unique security

H. Duan et al. (Eds.): SecureComm 2023, LNICST 568, pp. 101–121, 2025.
https://doi.org/10.1007/978-3-031-64954-7_6

challenges [7,32]. One of the most common and vulnerable cross-device security issues is unauthorized resource access. Unauthorized request across devices without proper permissions can raise privacy concerns for users. In addition, malware-infected devices can perform unauthorized operations and extract sensitive personal information across devices, creating even greater security and privacy risks.

Despite the security risks, existing cross-device access control solutions are far less comprehensive than single-device solutions. Permission model[1], which are critical security mechanisms for single devices, do not adequately support cross-device access control. Google provides coarse-grained and inflexible cross-device permission control. The Nearby[2] cross-device communication API developed by Google presents a pop-up window requesting location, microphone, WiFi, and Bluetooth permissions from the object device all at once, as shown in Fig. 1(a). Apple Handoff[3], which enables cross-device access between Apple devices, provides cross-device control limited to data, not actions. As shown in Fig. 1 (b), Handoff ensures data sharing security by requiring a verification code when transferring data from a personal device to a shared device.



(a) Google Nearby cross-device permission control (b) Apple Handoff cross-device privacy

Fig. 1. Existing cross-device access control solutions.

In addition, existing cross-device research primarily focuses on cross-device functional interaction [4,19], with relatively less emphasis on cross-device security. Among the existing cross-device security studies, researchers have focused on multi-user conflict issues in smart homes [3,24,25,29], cross-device authentication [10,27], and cross-device privacy design [30]. However, these solutions

[1] Permission is a security feature that an app must request from the user in order to access sensitive resources or data on the device. https://developer.android.com/guide/topics/permissions/overview.

[2] Nearby is a proximity-based communication technology that enables devices to discover and interact with nearby devices and services. https://developers.google.com/nearby.

[3] Handoff is a feature that enables seamless transfer of tasks and data between Apple devices, allowing users to continue their work uninterrupted across different platforms. https://support.apple.com/en-us/HT209455.

provide only coarse-grained control based solely on the device or user, and they have yet to address cross-device permission control.

In this paper, we propose and implement a distributed role and attribute hybrid-based access control (DHBAC) model to prevent unauthorized cross-device access. DHBAC supports flexible policy configuration to meet user access control needs. For example, children are not allowed to use payment apps on their parents' phones with their own phones across devices. DHBAC is the first cross-device access control model that provides permission-level control based on the genuine needs of users. We conducted a user study to gain insight into users' practical requirements for access control when initiating cross-device access. Based on the study, we designed the DHBAC model to provide permission granularity control. DHBAC extends the single-device permission system to cross-device access control, providing granular control based on users, devices, and applications.

Contributions. Our contributions can be summarized as follows:

- We conducted a user study to investigate security issues related to cross-device access and to gain insight into user expectations for cross-device access control models. Based on the research findings, we provided design principles for effective cross-device access control solutions.
- We extended the single-device permission model and proposed a novel cross-device access control model called DHBAC. DHBAC is based on multiple dimensions, including users, applications, devices, and contexts, which allows for fine-grained control of permissions.
- We implemented DHBAC on the Harmony Operating System (HarmonyOS) and evaluated its effectiveness in common cross-device scenarios. The results showed that DHBAC effectively and flexibly controls cross-device permissions while enhancing the security of cross-device access.

2 Background and Motivation

2.1 Access Control Needs of Cross-Devices Access

Users with multiple devices require access control mechanisms that are more flexible [31], fine-grained [23], and dynamic. The mechanism should be configurable and adaptable to various user roles, device types, and contexts. For example, access to medical devices needs to be restricted to authorized medical personnel and unauthorized activation across devices needs to be prevented. Similarly, mobile phones are typically configured to restrict access from public devices, which helps maintain personal privacy. In addition, corporate displays must prohibit projection to off-site locations to maintain the confidentiality of corporate information.

The smart home is a common scenario for cross-device access, where the hub is granted all permissions to initiate cross-device access directly to other smart

devices. However, many other cross-device access scenarios require permission control, especially when the initiating device is not trusted. Our goal is to control the permissions of all connected devices without compromising cross-device functionality. We strive to ensure the privacy and security of the called device at all times.

2.2 User Study and Design Guidelines

We conducted a user study to understand users' real-world access control needs for cross-device function calls. While previous research has explored the security and privacy preferences of smart home users [8,10,29,30], most studies have focused on single devices [2,22] or Internet of Things (IoT) devices [15] controlled by hubs. Cross-device access control presents unique security and privacy challenges and requires special measures. Our study aimed to investigate users' real-world access control needs for cross-device function calls. To accomplish this, we conducted a user study. The following paragraph describes our research process and presents our findings.

We first designed a questionnaire with simple options to understand the number of smart devices users owned, their understanding of smart home technology, and their familiarity with cross-device function calls. Forty-two participants completed the questionnaire, and we conducted semi-structured interviews with 15 of them, selecting people who owned at least three smart devices and were familiar with cross-device function calls. The 15 users came from different age groups, professional backgrounds, and education levels, and each interview lasted approximately 40 min. During the interviews, we focused on several topics, including users' security and privacy concerns with cross-device access, the granularity of users' expectations for cross-device permission control [12], users' ability to identify data at risk of leakage and users' needs for scenario-specific access control. We summarized and analyzed the results thematically, leading to the following conclusions:

- Participants' mental models of cross-device access control varied. More technically savvy participants expressed more doubts and concerns about data security than less technically savvy participants.
- Participants wanted more granular access control mechanisms that supported the principle of least privilege. They wanted more control over which devices and users could access which functions and data.
- Participants believed context-based dynamic access control in cross-device scenarios is necessary. Access control requirements vary based on the user's location, current time, and whether the application is foreground or background.
- Participants were uncomfortable and confused by unknown device connections. They expressed a desire for device connections and disconnections to be identifiable and controllable.

Based on the results of the above user study and recommendations from previous research on access control for smart homes [8,10,30], we developed

design guidelines for DHBAC. DHBAC is designed to meet these guidelines, which are as follows:

- Transparency of source device actions. Prevent the source device from unknowingly accessing the target device's resources.
- Minimize the privileges granted to the target device. DHBAC follows the principle of least privilege and grants permissions only when the source device actively requests them from the target device.
- Source device authentication. DHBAC requires the target device to verify the source device's trustworthiness before allowing cross-device function calls.
- Dynamic granting of permissions. DHBAC dynamically grants permissions for cross-device access based on context.
- Multiple access control policies. DHBAC supports customized access control policies to meet the cross-device security needs of different users.

2.3 Terminology

We define several terms that we consistently use throughout this work.

- **Subject Device:** The device that initiates the permission request.
- **Object Device:** The device that provides the associated resources.
- **Role:** A group of users with a defined set of permissions or capabilities.
- **Permissions:** Application permissions that support user privacy by protecting access to data and actions.
- **Cross-Device Access:** The subject device initiates an access request to the object device, and the object device provides the appropriate function.
- **Context:** Contextual information during cross-device access, such as location and time.
- **Cross-Device Access Control:** Manage and control cross-device access based on specified rules.

3 Scope and Threat Model

3.1 Scope

- We focus solely on multi-device security concerns and do not investigate multi-user conflicts in this study. Specifically, our research focuses on permission management for cross-device function calls.
- The access control model proposed in our work emphasizes the granularity of permissions granted rather than the granularity of operations performed. For example, allowing access to the camera is considered permission, while activating the camera is an operation.
- We do not consider remote function calls from the subject device to the object device. In other words, we assume that both devices' environmental context is the same.

3.2 Threat Model

Multi-device access control systems face threats from two primary adversaries: legitimate device owners and external attackers. Careless legitimate users may pose a security risk due to their mistakes or a lack of security awareness, while external attackers may gain unauthorized access through malicious devices. Although DHBAC can effectively protect against external attackers [12,33], it cannot eliminate the security risks posed by careless legitimate users. The effectiveness of DHBAC relies heavily on the security awareness of its users, as DHBAC provides users with autonomy in access control policies. We assume that neither the subject nor the object device will introduce malicious users and that the user-configured access control policies will prioritize security and privacy.

4 DHBAC Model

In this section, we introduce the DHBAC model. According to the evaluation of current IoT access control models by [5,16], the role-based access control (RBAC) model and the attribute-based access control (ABAC) model are considered the most appropriate for multi-user and multi-device smart home environments. While the RBAC model is easier to manage and audit, the ABAC model is more flexible, sophisticated, and customizable. To enable fine-grained control in the cross-device access control model and to facilitate user management, we propose a distributed role and attribute hybrid-based access control (DHBAC) model.

DHBAC is a cross-device access control model designed for mobile systems. The model provides a granularity of permissions for cross-device access, considering static and dynamic attributes. With DHBAC, users can easily customize access control policies for specific scenarios, avoiding default policies that may be too restrictive or permissive for different users. Elements controlled by DHBAC include subject and object users, subject and object devices, subject and object applications, and environmental contexts. Figure 2 illustrates the essential components of the DHBAC model and how they relate to each other.

4.1 DHBAC Controlled Elements

DHBAC is context-aware and regulates the subject and object devices in cross-device access through dynamic [20,28] and static attributes [6,9]. As shown in Fig. 2, the static attributes consist of user, device, and application attributes, while the dynamic attributes, i.e., environmental contexts, consist of current time and device location. The users predefine the static attributes, while the sensors capture the dynamic attributes. The subject user, subject device, and subject application determine the subject role, and the same is true for the object. The DHBAC control strategy involves identifying roles based on static attributes and defining policies for each role based on dynamic attributes.

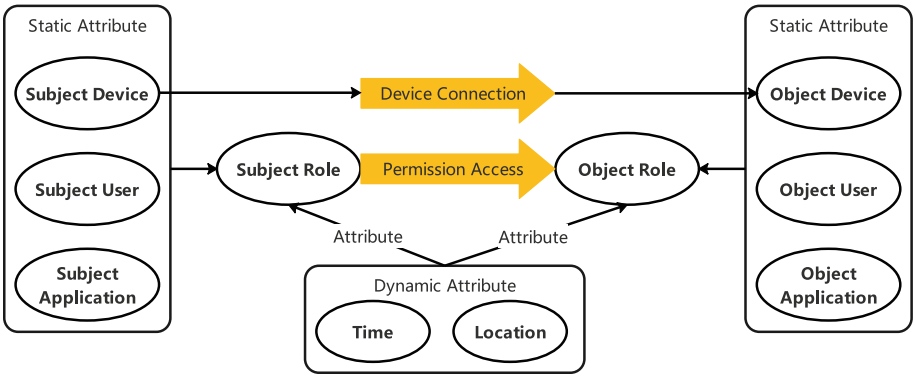


Fig. 2. The elements that DHBAC controls and the interaction between elements.

Determining Roles Through Static Attributes. Determining roles using static attributes involves the user, device, and application attributes. First, it is essential to clarify the classification and control methods of these three attributes.

User. Users log into the device's operating system using an online account. Only one user account can be logged on to a device at a time, and one user account can be logged on to multiple devices. Configuration of the trusted user list is based on the user identifier (user ID). Access is restricted only to those user IDs that are part of the trusted user list. The trusted user list plays a significant role in determining user roles, and all authorized roles require that the user ID is in the trusted user ID list.

Devices. Devices fall into two distinct categories based on their level of security: personal devices and shared devices. A single user owns and uses personal devices, while multiple users use shared devices. Personal devices log in with the owner's user ID; shared devices have no associated user ID. DHBAC specifies that permission requests cannot be initiated by shared devices, meaning that the subject device cannot be a shared device.

Application. Ensuring the security of third-party applications can be more challenging than system applications. To address this, DHBAC pre-configures system applications as high-security and third-party applications as low-security applications. However, the user has the flexibility to adjust and customize the security level of applications. For example, a user can manually designate a third-party application as a high-security application.

Role. Roles (R) are determined based on static attributes: User (U), Device (D), Application (A), i.e., $R = f(U, D, A)$ through the following rules:

- R = Administrator, *security level:3*
  Conditions: $U \in \{list\ of\ trusted\ user\ IDs\}\ \ AND\ D \in \{personal\ devices\}$ $AND\ A \in \{high\ security\ level\ applications\}$
- R = Host, *security level:2*
  Conditions: $U \in \{list\ of\ trusted\ user\ IDs\}\ \ AND\ D \in \{personal\ devices\}$
- R = Guest, *security level:1*
  Condition: $U \in \{list\ of\ trusted\ user\ IDs\}$

Determining Policies Through Dynamic Attributes. DHBAC provides fine-grained control based on dynamic attributes for high-risk permissions. DHBAC pre-configures basic policies for default smart home scenarios to simplify user operations. It also facilitates user-defined policies based on individual privacy preferences and varying cross-device access situations [21]. The dynamic attributes, i.e., environment attributes encompassing location and time [13]. Policies are formulated based on environment contexts (C), users (U), devices (D), and applications (A). Table 1 outlines the fundamental components of policy configuration.

Table 1. Fundamental components of policy configuration.

Component	Explanation
Subject Or Object Context	Contexts behind user privacy decisions
Subject Or Object User	Users logged into the subject or object device
Subject Or Object device	Device that initiates the call or the device that responds
Subject Or Object Application	Application that requests permissions or application that provides functionality

We use the default smart home scenario as an example to illustrate the strategy configuration method, as depicted in Table 2.

Table 2. Pre-configured smart home scenario strategy.

		Subject	Object
1	+	$C \in \{Home\}$	$D \in \{Gas\ Stove\}$
2	−	$C \in \{Night\}$	$A \in \{Steam\}$
3	−	$A \in \{App\ Store\}$	$A \in \{Paid\ apps\}$

Table 2 uses "+" to indicate an allowed policy and "−" to indicate a denied policy. Policy 1 specifies that cross-device access to the gas stove is only allowed when the subject device is at home. Policy 2 specifies that cross-device use of the Steam gaming application at night is prohibited on any device. Finally, policy 3 states that installing paid applications across devices is prohibited in any context.

4.2 DHBAC Work Flow

DHBAC uses two levels of protection to regulate cross-device access: low-risk permission control and high-risk permission control.

Low-Risk Permission Control. DHBAC establishes device connections by comparing user IDs. If the subject and object devices are logged in with the same user ID, DHBAC allows a direct connection between the subject and object devices. If the two devices are logged in with different user IDs, DHBAC requires that the user ID logged in by the subject device be in the list of users trusted by the object device. Once connected, the object device is granted access to the low-risk permissions of the subject device.

High-Risk Permission Control. Once a connection between the subject and object devices has been successfully established, access to high-risk permissions requires further control. High-risk permissions may pose a risk to user privacy, such as camera, microphone, and application data. These permissions need to be better protected, especially during cross-device access. To regulate access to high-risk permissions, DHBAC uses the concept of roles. First, DHBAC requires contexts to comply with user-defined policies; second, it compares the levels of subject and object roles. Specifically, the following rules are applied:

- High-risk permissions will be denied if the context does not comply with the user-defined policy.
- If the context complies with the user-defined policy and the subject role level is higher than or equal to the object role level, high-risk permissions will be granted directly.
- If the context complies with the user-defined policy, but the subject role level is lower than the object role, a pop-up window for the object device is displayed, prompting the user to decide whether to grant permission.

Figure 3 presents a flow chart illustrating the two levels of control for both low-risk permissions and high-risk permissions during cross-device function calls.

5 DHBAC System

This section outlines the architecture of DHBAC in mobile systems and its primary modules. As shown in Fig. 4, the dotted line represents the cross-device scheduling data flow, while the solid line represents the access control information flow. The subject application requests access to object device resources, and the business scheduling module is responsible for cross-device communication. DHBAC performs authentication on the object device before the object business scheduling module can access resources. DHBAC consists of five main modules: (1) device authentication module, (2) policy input module, (3) attribute collection module, (4) permission checking module, and (5) policy execution module.

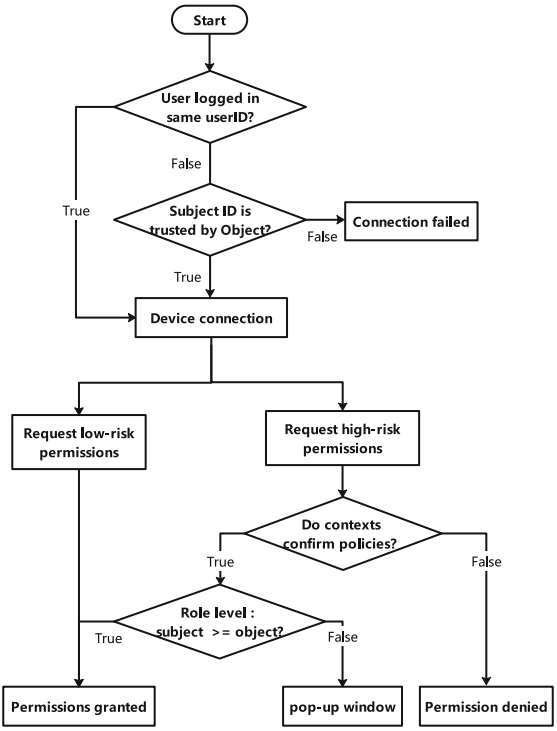


Fig. 3. DHBAC Work Flow

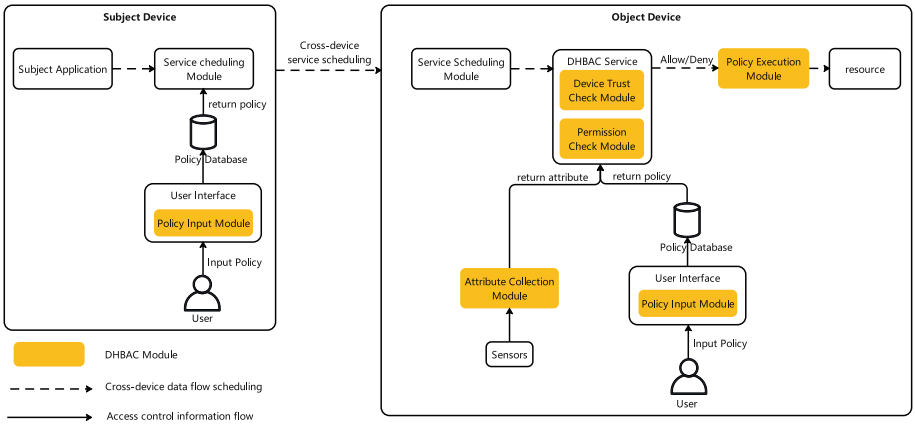


Fig. 4. The architecture of DHBAC in mobile systems and its primary modules.

5.1 Device Authentication Module

The device authentication module is used to evaluate the eligibility of the device connection. The evaluation is based on the user IDs registered by the subject and object devices. If two devices are registered to the same user IDs, DHBAC considers them to be personal devices of the same user and allows a direct connection. However, suppose the two devices are registered under different user IDs. In that case, DHBAC requires that the user ID registered by the subject device is included in the list of user IDs trusted by the object device.

5.2 Policy Input Module

The policy input module allows users to customize access control policies through a user interface. Policies can be created according to the user's privacy preferences. Policies are created based on environmental contexts, users, devices, and applications and then stored in the policy database.

5.3 Attribute Collection Module

The attribute collection module is responsible for collecting and storing both static and dynamic attributes. When an authorization request is initiated, the subject's static attributes are synchronized to the object device and then used by DHBAC for authentication. It is important to note that in our work, the dynamic attributes of the subject and object devices are assumed to be identical.

5.4 Permission Checking Module

The attribute collection module is responsible for collecting and storing both static and dynamic attribute information. Static attributes include user, device, and application, while dynamic attributes include the current time and location. When an authorization request is initiated, the subject's static attributes are synchronized with the object device and then used by DHBAC for authentication. It is important to note that, for this paper, the dynamic attributes of the subject and object devices are assumed to be identical.

5.5 Policy Execution Module

The policy execution module is responsible for executing the final access control decisions, including accessing the object device's resources if permission is granted and notifying the subject device of rejection if permission is denied.

The device authentication and permission checking module are collectively called the DHBAC service. During the cross-device access, the subject device synchronizes its policy information to the object DHBAC service over the distributed soft bus. The object DHBAC service then obtains the object policy information through cross-process communication and the current time and location through the mobile device's embedded sensors. Based on this information,

the DHBAC service performs permission control, consisting of two steps: low-risk and high-risk permission control. First, DHBAC verifies the subject and object user IDs. If the verification is successful, low-risk permissions are granted; otherwise, an empty permission set is returned. Second, DHBAC filters the available high-risk permissions based on user-defined policies. The DHBAC model then prompts the user to confirm the permission request via a pop-up box, as shown in Fig. 5.

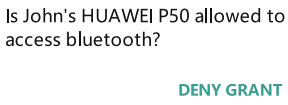


Fig. 5. Permission request pop-up window

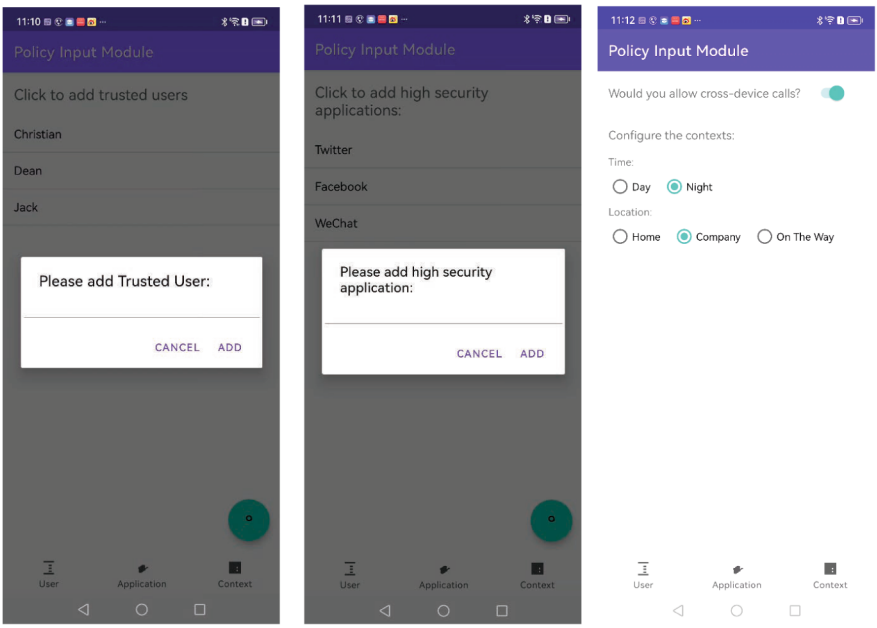
DHBAC follows the standard access control architecture. The Policy Entry Module serves as the Policy Administration Point (PAP), allowing users to create and store policy information in the policy database. The Attribute Collection Module acts as the Policy Information Point (PIP), collecting contextual information for policy evaluation through sensors on mobile devices. The Permission Checking Module and the Device Authentication Module act as Policy Decision Points (PDPs), retrieving the access policy from the policy database, retrieving the attribute values of the access request from the attribute database, and finally evaluating the access request against the policy. Finally, the Policy Execution Module acts as a Policy Decision Point (PEP), determining whether to accept or deny the permission and providing the system interface.

6 DHBAC Implementation

The DHBAC system is built on HarmonyOS version 3.0.0, a distributed operating system developed by Huawei to collaborate and interconnect with multiple smart devices in the IoT ecosystem. The HarmonyOS cross-device software development kit (SDK) enables developers to perform efficient distributed development. In our tests, we used a Huawei P50 as the subject device and a Huawei Mate40 as the object device. DHBAC extends and modifies the existing HarmonyOS framework. We integrated the DHBAC service, i.e., the device authentication module and the permission checking module, into the Harmony system security service as an application package. The DHBAC service accordingly provides distributed authentication interfaces to the system security service. The HarmonyOS distributed soft bus capability supports remote communication between devices. DHBAC comprises approximately 6000 lines of code, primarily written in Java.

To minimize changes to the existing HarmonyOS framework, we installed the Policy Input Module as a standalone application on both the subject and

object devices. It provides a policy configuration interface for users and stores user-defined policies. Figure 6 shows its policy configuration interface. Before initiating cross-device access, the subject and object devices prepare the following data through the policy input module:



(a) Configure trusted users (b) Configure high-security level applications (c) Configure context

Fig. 6. User interface of policy configuration application.

- User: A list of trusted user IDs.
- Devices: A flag bit used to differentiate between shared and personal devices.
- Applications: A list of high-security level applications.
- Permissions: A list of permissions that restrict cross-device access.

7 Performance Evaluation

7.1 Effectiveness

In this section, we test four common everyday cross-device access control scenarios to verify the effectiveness of DHBAC. We installed DHBAC on both the subject and object devices to enhance the security of cross-device access. Figure 7 shows the cross-device access process of the four scenarios using the elements of users, devices, applications, and contexts.

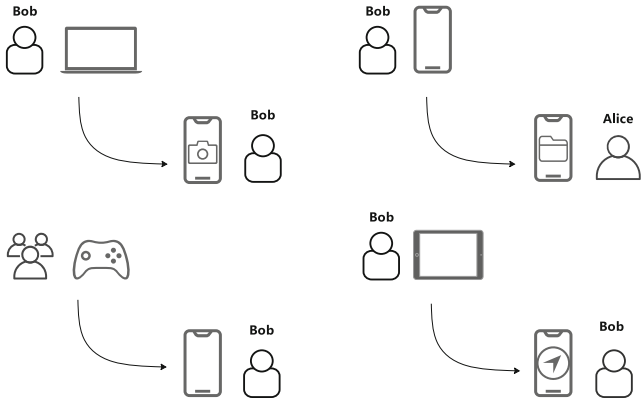


Fig. 7. Cross-device access process of the four test scenarios.

Scenario 1. The Tencent Meeting application on Bob's laptop wants to call the camera on Bob's phone for a multi-camera conference. In this scenario, as shown in Table 3, the subject and object devices are logged in under the same user ID and are personal devices. Since Tencent Meeting is a three-party application, it is a low-security application by default. Based on the user, device, and application, DHBAC defines the subject role as the host and the object role as the administrator. The requested camera permission is considered a high-risk permission.

Table 3. Scenario 1: The subject role has a lower security level than the object.

	Subject	Object
User	Subject and Object logged into the same user account	
Device	Personal device	Personal device
Application	Low-security application	/
Role	Host	Administrator
Permission	High Risk Permission	
Context	Contextual compliance with user-defined policies	

DHBAC Result: The subject role has a lower security level than the object role. DHBAC activates a permission request pop-up window on Bob's phone. After Bob clicks the grant button, Bob's laptop gains access to the camera permission on Bob's phone.

Scenario 2. The notes application on Bob's phone wants to access personal files on Alice's phone. In this scenario, as shown in Table 4, the subject and

object devices are logged on with different user IDs. The primary requirement for initiating a connection between Bob's and Alice's phones is that Alice has added Bob's phone to the list of trusted devices. Both the subject and object devices are personal devices, and the note application is a system application, making it a high-security level by default. DHBAC defines the subject and object roles as administrators based on the user, device, and application. The requested file permission is considered high-risk permission.

Table 4. Scenario 2: Different users of the subject and object.

	Subject	Object
User	Subject and Object logged into the different user account	
Device	Personal device	Personal device
Application	High-security application	/
Role	Administrator	Administrator
Permission	High Risk Permission	
Context	Contextual compliance with user-defined policies	

DHBAC Result: The two devices can successfully connect because Alice has added Bob's phone to the list of trusted devices. Since the security levels of the subject and object roles are equivalent, the note application is granted media and file permissions.

Scenario 3. The arcade machine in the mall wants to access the phones of the customers using it to get personal information. In this scenario, as shown in Table 5, DHBAC has a special treatment for shared devices. If the subject device is a shared device, the object device filters all device resources and user data permissions, which helps prevent privacy leaks and unwanted interference.

Table 5. Scenario 3: Shared Device Scenario.

	Subject	Object
User	/	Personal ID
Device	Shared device	Personal device
Application	Low-security application	/
Role	Guest	Administrator
Permission	High Risk Permission	
Context	Contextual compliance with user-defined policies	

DHBAC Result: Game machines in an arcade are shared devices, and customers' phones will deny any requests for high-risk permissions.

Scenario 4. The map application on Bob's tablet seeks to access the location permissions on Bob's phone. In this scenario, as shown in Table 6, we defined the contextual rules as $time : [Weekday, 9:00 - 17:00], location : [Company]$. It means high-risk permissions can only be granted across devices during weekday work hours and when the device is in the company. The requested location permission is a high-risk permission. In our test, the context is a Sunday, and the phone is in the company.

Table 6. Scenario 4: Context-based access control.

	Subject	Object
User	Subject and Object logged into the same user account	
Device	Personal device	Personal device
Application	/	/
Role	Administrator	Administrator
Permission	High Risk Permission	
Context	Context does not match user-defined policies	

DHBAC Result: Since the current context does not match the user-specified policy, all high-risk permissions are denied, regardless of the security level of the subject role and object role. DHBAC issues an alert on the subject device, notifying the user that the environmental context is not permitted.

7.2 Performance Overhead

We tested the permission checking overhead of DHBAC. In HarmonyOS, permission checking is performed at the framework and kernel layers. Since our modifications only affect the framework layer and the cost of permission checking at the framework layer is always higher than that at the kernel layer. We mainly evaluated the latency of permission checking at the framework layer.

First, we configured user policies for the subject and object devices using the policy configuration interface. Since DHBAC provides a distributed authentication interface to the system service, we developed a test application to call the interface provided by DHBAC. DevEco Studio's[4] *bytrace* performance analysis tool was used to capture the relevant threads and record the latency of the distributed authentication process.

The test's time delay encompasses when a subject device initiates a permission request and receives a verification result from the object device. The delay includes the following processes:

[4] HUAWEI DevEco Studio. An integrated development environment (IDE) designed to help developers create high-quality applications for Huawei devices and other platforms. https://developer.harmonyos.com/en/develop/deveco-studio/.

- The subject device triggers the DHBAC interface, and the business scheduling module initiates cross-device access to synchronize the policy information of the subject device to the object device.
- The object device receives the request and simultaneously reads the object device's policy information from the database. It then analyzes the subject and object policy information and context to determine whether permission should be granted.
- The object device returns the authentication result to the subject device.

The total time required for the process consists primarily of cross-device communication, database queries, and permission checks. The number of synchronized subject device policies affects the cross-device communication latency. Database query latency is affected by the size of the policy set and the frequency of database accesses. In real-world scenarios, policies for appropriate access control are generally easy to implement, and the number of stored policies will not exceed 10. Furthermore, the database only needs to be revisited when the user policy changes, making access manageable.

In this test, the number of policies stored on the subject and object devices is 10, and the four scenarios in Sect. 7.1 are repeated ten times for testing. The results are shown in Table 7.

Table 7. DHBAC Permission Check Overhead in Different Scenarios.

Test Scenario	No. of policies	Average time (s)	Success rate
Scenario 1	10	0.32	100%
Scenario 2	10	0.38	100%
Scenario 3	10	0.19	100%
Scenario 4	10	0.26	100%

The number of user-configured policies affects the latency of cross-device communication and database queries. To investigate the correlation between the number of policies and the latency of DHBAC permission checking, we performed additional tests by increasing the number of policies through the policy configuration interface. The results, shown in Fig. 8, show a positive correlation between the number of policies and distributed authentication latency.

8 Related Work

Research in cross-device security falls into four areas: cross-device authentication, cross-device privacy data preservation, cross-device communication security, and cross-device access control. Previous work on cross-device authentication includes a multi-device user authentication solution proposed by Laing *et al.* [14], which supports second-factor and passwordless authentication using

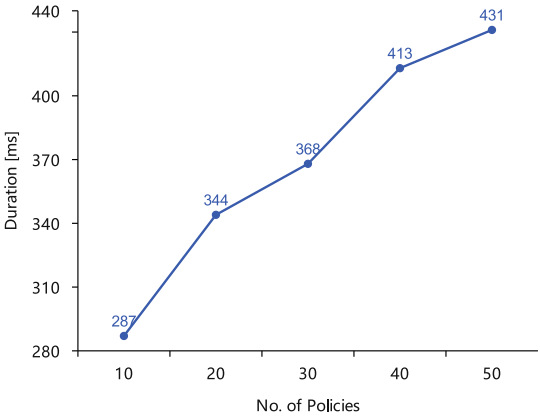


Fig. 8. Impact of different number of policies on DHBAC's performance.

hardware tokens. In addition, Hintze *et al.* [11] proposed a risk-aware multimodal biometric cross-device authentication scheme that uses multiple biometric features for authentication on mobile devices. Cross-device privacy data preservation has been addressed in recent work, such as the cross-device federation learning-based Android security solution proposed by Singh *et al.* [26]. Cross-device communication security research has been conducted by Obaidat *et al.* [17], who introduced a distributed cross-device communication approach to enable a new trust paradigm for access control of IoT applications. Cross-device access control is where we're focused, with most studies in this area focusing on smart home scenarios. Some of these studies have investigated security and privacy issues and access control expectations of smart home users and provided recommendations for the design of IoT access control [10,30]. These studies provide valuable insights for the design of our proposed approach, the DHBAC model. In addition, addressing the multi-user conflict problem has been a popular topic in cross-device access control research. Studies such as the novel multi-user and multi-device aware access control mechanism proposed by Sikderd *et al.* [25] have attempted to address this problem. DHBAC extends the permission system, addressing the multi-device access control problem ignored in these previous studies.

9 Conclusion

In this study, we propose and implement a Distributed Hybrid-Based Access Control (DHBAC) model to enhance the security of cross-device function calls on mobile devices. We developed the design guidelines for the model by conducting a user study, taking into account the unique security and privacy concerns associated with cross-device access control. DHBAC extends the single-device permission system to cross-device access control. We have conducted real-world

testing of DHBAC. Our results show that it can effectively block malicious cross-device access and mitigate the security risks of cross-device access with acceptable system overhead.

References

1. Global infrastructure index 2021. Technical report, Ipsos (2021). https://www.ipsos.com/sites/default/files/ct/news/documents/2021-10/Global-Infrastructure-index-2021-ipsos.pdf
2. Abdella, J., Özuysal, M., Tomur, E.: CA-ARBAC: privacy preserving using context-aware role-based access control on Android permission system. Secur. Commun. Netw. **9**(18), 5977–5995 (2016). https://doi.org/10.1002/sec.1750
3. Abdi, N., Ramokapane, K.M., Such, J.M.: More than smart speakers: security and privacy perceptions of smart home personal assistants. In: SOUPS@ USENIX Security Symposium (2019). https://doi.org/10.5555/3361476.3361510
4. AlDuaij, N., Nieh, J.: Tap: an app framework for dynamically composable mobile systems. In: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, pp. 336–349 (2021). https://doi.org/10.1145/3458864.3467678
5. Ameer, S., Benson, J., et al.: Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT. IEEE Trans. Dependable Secure Comput. (2022). https://doi.org/10.1109/ACCESS.2021.3149170
6. Bezawada, B., Haefner, K., Ray, I.: Securing home IoT environments with attribute-based access control. In: Proceedings of the Third ACM Workshop on Attribute-Based Access Control, pp. 43–53 (2018). https://doi.org/10.1145/3180457.3180464
7. Brudy, F., et al.: Cross-device taxonomy: survey, opportunities and challenges of interactions spanning across multiple devices. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–28 (2019). https://doi.org/10.1145/3290605.3300792
8. Geeng, C., Roesner, F.: Who's in control? Interactions in multi-user smart homes. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2019). https://doi.org/10.1145/3290605.3300498
9. Goyal, G., Liu, P., Sural, S.: Securing smart home IoT systems with attribute-based access control. In: Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems, pp. 37–46 (2022). https://doi.org/10.1145/3510547.3517920
10. He, W., et al.: Rethinking access control and authentication for the home Internet of Things (IoT). In: USENIX Security Symposium, pp. 255–272 (2018). https://doi.org/10.5555/3361476.3361510
11. Hintze, D., et al.: Cormorant: on implementing risk-aware multi-modal biometric cross-device authentication for Android. In: Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia, pp. 117–126 (2019). https://doi.org/10.1145/3365921.3365923
12. Jeon, J., et al.: Dr. Android and Mr. Hide: fine-grained permissions in Android applications. In: Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 3–14 (2012). https://doi.org/10.1145/2381934.2381938

13. Kumar, S., Shanker, R., Verma, S.: Context aware dynamic permission model: a retrospect of privacy and security in Android system. In: 2018 International Conference on Intelligent Circuits and Systems (ICICS), pp. 324–329. IEEE (2018). https://doi.org/10.1109/ICICS.2018.00073
14. Laing, T., Marin, E., Ryan, M.D., Schiffman, J., Wattiau, G.: Symbolon: enabling flexible multi-device-based user authentication. In: 2022 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1–12. IEEE (2022). https://doi.org/10.1109/SP40001.2021.00042
15. Mare, S., Girvin, L., Roesner, F., Kohno, T.: Consumer smart homes: where we are and where we need to go. In: Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, pp. 117–122 (2019). https://doi.org/10.1145/3301293.3302371
16. Mohammad, Z.N., Farha, F., Abuassba, A.O., Yang, S., Zhou, F.: Access control and authorization in smart homes: a survey. Tsinghua Sci. Technol. **26**(6), 906–917 (2021). https://doi.org/10.26599/TST.2021.9010001
17. Obaidat, M.A., Brown, J., Al Hayajneh, A.: A novel paradigm for access control trust in IoT applications: a distributed cross-communication approach. In: 2021 13th IFIP Wireless and Mobile Networking Conference (WMNC), pp. 25–31. IEEE (2021). https://doi.org/10.23919/WMNC51386.2021.9618899
18. Oh, S., et al.: Fluid: flexible user interface distribution for ubiquitous multi-device interaction. In: The 25th Annual International Conference on Mobile Computing and Networking, pp. 1–16 (2019). https://doi.org/10.1145/3300061.3345443
19. Oh, S., Yoo, H., Jeong, D.R., Bui, D.H., Shin, I.: Mobile plus: multi-device mobile platform for cross-device functionality sharing. In: Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pp. 332–344 (2017). https://doi.org/10.1145/3081333.3081348
20. Olejnik, K., Dacosta, I., Machado, J.S., Huguenin, K., Khan, M.E., Hubaux, J.P.: SmarPer: context-aware and automatic runtime-permissions for mobile devices. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 1058–1076. IEEE (2017). https://doi.org/10.1109/SP.2017.41
21. Ringer, T., Grossman, D., Roesner, F.: Audacious: user-driven access control with unmodified operating systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 204–216 (2016). https://doi.org/10.1145/2976749.2978344
22. Rohrer, F., Zhang, Y., Chitkushev, L., Zlateva, T.: DR BACA: dynamic role based access control for Android. In: Proceedings of the 29th Annual Computer Security Applications Conference, pp. 299–308 (2013). https://doi.org/10.1145/2523649.2523676
23. Sanders, M.W., Yue, C.: Mining least privilege attribute based access control policies. In: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 404–416 (2019). https://doi.org/10.1145/3359789.3359805
24. Sikder, A.K., et al.: Kratos: multi-user multi-device-aware access control system for the smart home. In: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 1–12 (2020). https://doi.org/10.1145/3395351.3399358
25. Sikder, A.K., et al.: Who's controlling my device? Multi-user multi-device-aware access control system for shared smart home environment. ACM Trans. Internet Things **3**(4), 1–39 (2022). https://doi.org/10.1145/3543513
26. Singh, A., Goyal, N.: Android web security solution using cross-device federated learning. In: 2022 14th International Conference on COMmunication Systems &

NETworkS (COMSNETS), pp. 473–481. IEEE (2022). https://doi.org/10.1109/COMSNETS53615.2022.9668449
27. Tian, Y., et al.: SmartAuth: user-centered authorization for the Internet of Things. In: USENIX Security Symposium, vol. 5, pp. 8–2 (2017). https://doi.org/10.5555/3241189.3241219
28. Wijesekera, P., et al.: The feasibility of dynamically granted permissions: aligning mobile privacy with user preferences. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 1077–1093. IEEE (2017). https://doi.org/10.1109/SP.2017.49
29. Zeng, E., Mare, S., Roesner, F.: End user security and privacy concerns with smart homes. In: Symposium on Usable Privacy and Security (SOUPS), vol. 220 (2017). https://doi.org/10.5555/3235924.3235931
30. Zeng, E., Roesner, F.: Understanding and improving security and privacy in multi-user smart homes: a design exploration and in-home user study. In: USENIX Security Symposium, pp. 159–176 (2019). https://doi.org/10.5555/3361338.3361350
31. Zhang, H., Agarwal, Y., Fredrikson, M.: TEO: ephemeral ownership for IoT devices to provide granular data control. In: Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services, pp. 302–315 (2022). https://doi.org/10.1145/3498361.3539774
32. Zhang, R., et al.: User experience for multi-device ecosystems: challenges and opportunities. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–5 (2021). https://doi.org/10.1145/3411763.3441325
33. Zhang, Y., Yang, M., Gu, G., Chen, H.: FineDroid: enforcing permissions with system-wide application execution context. In: Thuraisingham, B., Wang, X.F., Yegneswaran, V. (eds.) SecureComm 2015. LNICST, vol. 164, pp. 3–22. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-28865-9_1