Coordination Proxy for Secure Interaction of Mobile Devices

T. Nakagawa, M. Ohata, K. Ohta , and H. Inamura

Research Laboratories, NTT DOCOMO
3-5, Hikari-no-oka, Yokosuka, Kanagawa, 239-8536, Japan

ABSTRACT

Establishing ad hoc coordination between extensible smartphone and external devices will realize various applications.To solve the new problems that will arise when smartphones must coordinate with devices with different trust levels, we extracted the requirements for a comprehensive coordination framework through use case analysis and threat analysis. Our proposed proxy mechanism combines connectivity management, recovery, dynamic filtering, and message handling to satisfy the requirements.

We focus on dynamic filtering and show how to implement it on the Android platform. Because a smartphone is utilized in various situations and the terminal's internal or external status changes continuously, a uniform static policy is not acceptable[2]. Our coordination proxy enables smart cooperation between smartphones and external devices since the dynamic filtering policy changes with both the internal and external status of the smartphone. We implement the proposed proxy in Android G1, and an evaluation shows that the overhead of the processing time in the proxy is less than or equal to 2% for the file transfer scenario.

$\boldsymbol{Keywords}$: Security analysis, Smartphone, Dynamic filtering, Mobile devices, Ad-hoc network, Android platform

1 INTRODUCTION

Smartphones are gathering attention due to their high extensibility which enables flexible application development. For instance, in Google's Android, mechanisms called Intent and Content Provider are provided to support loose cooperation between applications by different developers, by enabling transfer of events or data among them. In the near future, we assume that applications on smartphone will need to cooperate with not only applications on the same smartphone, but also those on external devices such as PCs or home appliances.

In this paper, we enumerate exhaustive threats which are derived from security analysis, and clarified requirements for coordination between a smartphone and an external device. The threat analysis is characterized by the consideration on various confidential files such as business documents and contact information, and internal status such as remaining battery, storage depletion, and installed software. We also study requirements from the viewpoint of usability, and propose a coordination proxy which provides connectivity management function, recovery function, dynamic filtering function, and message handling function.

The proposed coordination proxy alleviates the risk of personal or confidential information leakage, and realizes smart coordination for resource preservation. We adopted Android for prototyping due to its high extensibility, and used Android platform's coordination mechanisms such as Intent or Content Provider to implement the coordination proxy. Although Intent mechanism realizes flexible message passing among internal applications, we must enhance the mechanism to realize the proposed coordination proxy. From the evaluation result, we confirmed that the overhead of the proposed method is less than or equal to 2% in the file transfer scenario, which seems small enough for practical use.

2 PREVIOUS RESEARCH

In [1], a filtering mechanism is proposed based on existing web proxy tunneling that prevents information disclosure. Though incoming and outgoing information flows have different filtering policies, the policies are statically defined.

In [2], the combination of an adaptive access control module and an adaptive function invocation module is proposed for secure access to remote services. The access control module takes the user's context such as user's role, location, and time into consideration to realize more fine-grained control. Although this is similar to our research in that the access control policy is dynamically changed by changes in the situation, we tackle the challenge to adapting to changes in the internal status of the terminal, an issue not considered in [2].

To coordinate different softwares on a smartphone, Android platform[3] provides Intent function. When an application issues an implicit Intent, the Intent resolution mechanism provide candidate softwares that can handle the Intent. This loosely-coupled coordination system provides revolutionary flexibility in that the software originating the Intent doesn't know which application will be activated. Thus, softwares developed across the globe can complement each other's functionality.

Among recent standards for the coordination of home appliances and PCs, Digital Living Network Alliance(DLNA)[4] offers services such as the secure transfer of photos or video among authenticated devices. DLNA uses Digital Transmission Content Protection over Internet Protocol(DTCP-IP) [5]to realize compatibility of different copyright protection technologies. Another facet of software coordination for smartphones is the recent activity to provide contents development environment based on HTML5[6]. There is a chance that various devices are remote controlled by using HTML5-enabled phone in the future. There is also a study based on Home Gateway Initiative[7], in which an architecture for discovery and management of heterogeneous devices is proposed to realize remote management of devices in home network[8]. In

papers [9], methods are proposed to protect contents or files by connecting the home network and mobile phone through VPN. In these studies regarding coordination of the home network and mobile devices, the standardization of communication protocol and media formats was the main target, and so the security problem still remains. In particular, the dynamic attributes of smartphones were not considered by previous research.

The goal of this research is to realize secure and convenient transfer of file and command between smartphone and other devices such as other smartphones, PCs, or home appliances. In this paper, we assume a use scenario that wide variety of information is exchanged with external devices under ever changing circumstances. We regard trust level of installed software, location or time of usage, connected network, etc. as the variable condition determining the security level the smartphone is situated. We tackle the challenge to realize flexible and secure coordination method considering these condition.

3 PROPOSED METHOD

3.1 Use Case Analysis

Sample use cases assumed in this paper are categorized into file transfer scenario and command transfer scenario, which features different requirements of communication quality. In the former scenario, files from several KB to sveral hundred MB are transfered. In order to shorten the total communication time, throughput improvement is important. In the latter scenario, commands consisting of small data segments are transfered, and the goal is to shorten message transfer delay to realize acceptable response time.

Examples of those scenarios are shown in Figure 1. As shown in the left side of the figure, e-mails or business documents stored in a PC can be viewed on the display of an Android-based smartphone without opening the PC when a user is travelling. When the user accesses the PC from the smartphone, ORIGINAL_PC_VIEWER Intent is issued from the smartphone to the PC. The noteworthy feature here is that a PC viewer, which is a software to browse files in the PC, is operated on the phone. The PC viewer gives the measure to send control information such as ”move directory” or ”open file” by using original protocol.

Also, as shown in the right of the figure, the same user's smartphone is coordinated with another PC in different occasion. When an incoming call is answered by the user, ANSWER Intent is generated by Android platform. This triggers the transmission of a message that can be interpreted by the external PC. The viewer application is activated on the PC on receipt of the message, and related information such as business card, schedule, or recent e-mails of the caller is automatically displayed.

Following two requirements are derived from the use cases mentioned above. Capital letters from A to D used in this section and the next section corresponds to the functions proposed in Section 3.3.

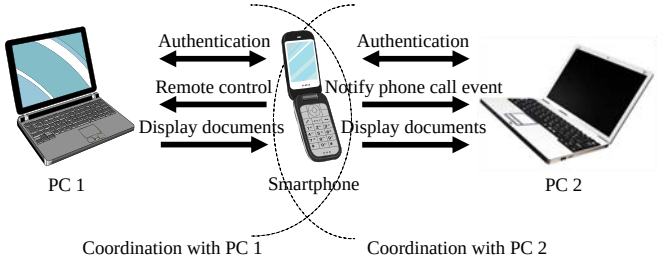

Figure 1: Use case of coordination with PCs

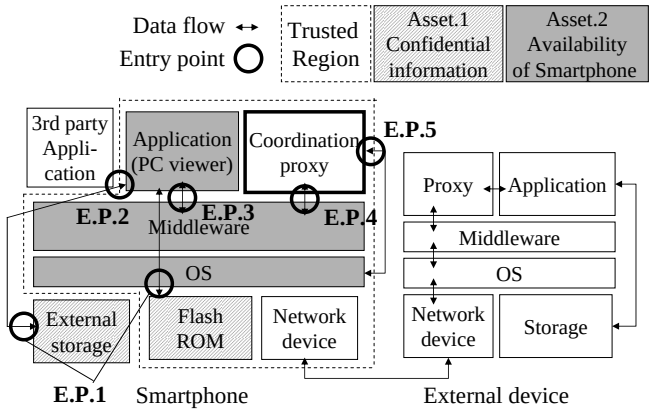

Figure 2: Data flow for coordination scenarios

- A.Efficient management of wireless link
- B.Handling of error raised in external device

For requirement A, it is assumed that a smartphone and a PC are connected by wireless link such as wireless LAN, Bluetooth, or possibly cellular network. As the management method of wireless link has large impact on battery resource and response time, it is necessary to provide flexible method to manage connection depending on the characteristics of coordinated application.

For requirement B, even if the connection status is healthy between the smartphone and the external device, some error can occur in the external device. Given that the smartphone user may not have direct physical access to the PC, the user should know what is going on at the external device. The proposed method, shown in Section 3.3, satisfies these usability requirements.

3.2 Threat Analysis

In this section, we study security requirements based on the use cases shown above, by applying Microsoft's threat analysis method[10]. Figure 2 shows the data flow diagram, which specifies assets and entry points. The assets to be protected are defined as 1)confidential information stored in the smartphone's internal flash memory or external storage, and 2)the availability of smartphone software including OS, middleware, and application. The availability is defined as the

Table 1: Threats for each entry point

	Asset 1			Asset 2			
	E.P.1	E.P.4	E.P.5	E.P.2	E.P.3	E.P.4	E.P.5
Spoofing			Th.5		Th.10		
Tampering	Th.1	Th.1	Th.1	Th.6	Th.6	Th.6	Th.6
Repudiation							
Information disclosure	Th.2 Th.3	Th.2 Th.3	Th.2 Th.3				
DoS attack	Th.4	Th.4	Th.4	Th.7	Th.11 Th.12	Th.11 Th.12	Th.11 Th.12
Elevation of privilege				Th.8 Th.9	Th.8 Th.9	Th8 Th.9	Th.8 Th.9

status that a user can utilize the software correctly[10].

Also, the smartphone is regarded as trusted region, where no malware is assumed to exist at shipment. The external storage of smartphone is not included in the trusted region because external memories such as micro SD card can be a potential input method of malicious files.

The enumeration of conceivable threat is shown in Table 1. The contents of each threat corresponding the threat number is shown in Table 2, which shows extracted threats and countermeasures for them. Threats already found in existing smartphone are out of scope because they are regarded to be remedied by security functions of existing smartphone.

Out of the extracted threat list, six items are intended to be solved in this paper as the new threats featured in our use cases. These are categorized as following requirements by aggregating related items, among which C2 and C3 are especially characteristic for smartphones in that they consider user's privacy and resource scarcity.

- A2.Authentication of external devices (Th.1,6)
- C1.Filtering of messages or files(Th.2,6,10)
- C2.Filtering considering internal status(Th.2)
- C3.Filtering considering external status(Th.4,9)
- D1.Flow control of coordination message(Th.10)

For C2, whether message and file transfer is allowed or not depends on not only the authentication information between the devices, but also internal status such as remaining battery power, storage space, or trust level of installed software.

Regarding the remaining battery and storage space, it is prohibited, or a warning message is displayed, to transfer large size files when the smartphone is lack of those resources. Thus, in the use case of remote control of a business PC, resource-aware smart operation is realized without user's cumbersome periodical check of remaining resources.

Also in the use case of phone call event notification, battery power consumption is alleviated by introducing multi-level power saving mode. With normal battery power level, full of the related information is displayed on the PC's display, including caller's business card, meeting schedules, and

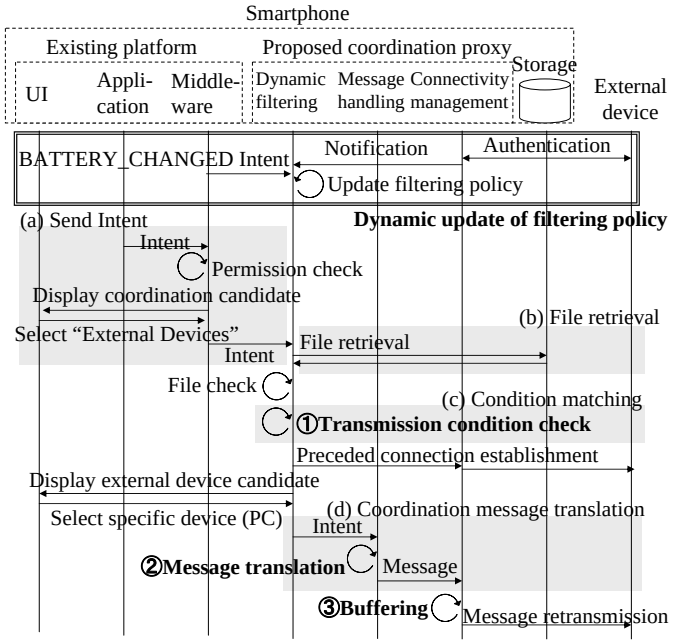

Figure 3: Coordination sequence for file transfer

recent e-mails. At the lowest level of battery, no information is transfered to avoid unnecessary power consumption.

In terms of installed software, sensitive files and phone call events are forbidden to be transfered when prohibited software is installed in the smart phone to prevent information disclosure. The background of this idea is that arbitrary software is freely downloaded and installed from any server to the smartphone. Function to restrict software installation is widely provided so far. For example, in Windows, an administrator can utilize group policy editor to limit software installation following a corporate policy. It would be a natural extension of this kind of service to provide function to change the policy depending on the installed software.

For C3, the permission of message and file transfer is judged by external status such as time and location. The goal is to limit the purpose to use confidential information. The risk of information disclosure is mitigated by restricting coordination to appropriate situation. For example, business documents are allowed to access only on weekdays and business hours, and in case the user is located at the office.

3.3 Architecture of Coordination Proxy

In order to fulfill the requirements derived from use case analysis and security analysis, we propose following coordination proxy that provide four major functions. These functions enable smooth coordination with external device from usability viewpoint. Additionally, from the viewpoint of security, information disclosure is prevented by fine-grained transmission control against files stored on smartphone. Moreover, availability of smartphone system is assured by guarding against malicious coordination from external devices.

Table 2: Countermeasures for enumerated threats

No.	Threat	Countermeasure
1	The file in the smartphone is tampered by illegal overwriting from external device.	Judge if the file is allowed to receive depending on the authentication status.
2	Information disclosure occurs due to prohibited software installed in the smartphone.	Select files and events which are permitted to send or receive considering installed software.
3	Information disclosure occurs due to unintended software usage.	Select files and events which are permitted to send or receive considering user's context.
4	The file managed by a smartphone application is illegally deleted by other applications.	Out of scope because it's existing threat and should be protected by file management function of the smartphone.
5	External device spoofs to be legitimate device and attacks smartphone OS, middleware, or application.	Conduct appropriate authentication.
6	The function of OS, library, middleware of the smartphone is tampered by malicious coordination message received from external device.	Discard illegal messages.
7	External device sends plenty of files and run out of smartphone's storage.	In addition to Countermeasure 1, judge if the file is allowed to receive depending on the available storage capacity.
8	Privilege is acquired by malicious file sent from external device.	Out of scope because it's existing threat and should be protected by smartphone's memory protection function.
9	Privilege is acquired by malicious file stored in smartphone's storage.	Same as Countermeasure 8.
10	External device spoofs to be legitimate device and illegally access confidential files.	Same as Countermeasure 5.
11	External device triggers transmission of a large number of files, intentionally causing slow system behaviour and battery depletion.	Monitor and control the flow of received messages. Judge transmission considering remained power resource.
12	Smartphone application sends plenty of coordination messages, causing slow system behaviour and battery depletion.	Out of scope because it's existing threat and should be handled by smartphone's application management system

- A.Connectivity management function
- B.Recovery function
- C.Dynamic filtering function
- D.Message handling function

Figure 3 shows the sequence of sending message out to external devices. Smartphone application transmits a coordination message to dynamic filter function. The message is at first handled by smartphone middleware, and then coordination candidate devices are displayed on the smartphone's screen. After user selects the category of "external device", the message is handled by the coordination proxy.

3.3.1 Connectivity Management Function

Connectivity of the device is susceptible to external factors such as user and device movement and wireless link instability. This function manages connectivity and changes the coordination candidates dynamically. Authentication of external devices is conducted by using existing protocols such as password authentication or PKI authentication.

In addition to simple management of connectivity, it supports always-on mode, which is advantageous in terms of response time, and on-demand mode, which is favorable in terms of power saving. Always-on mode is applied when the smartphone's battery is fully charged, or being recharged.

For on-demand mode, those applications which need quick connection establishment are provided with connection priority. The connection establishment is initiated when the device candidates are shown to the user. In case active connections to devices are not selected by the user, these connections are cancelled. Moreover, for messages that don't require realtime transmission, this function temporarily buffers them if device connection fails, and retransmits the buffered messages when the communication link is recovered.

3.3.2 Recovery Function

If the external device experiences some sort of error, this function displays the error messages or instructions sent from the device, with the goal of enabling problem solving from the smartphone. Take the example of the scenario that the PC displays caller's information such as business card, schedule, or latest e-mails on receipt of an incoming call. If the caller's information cannot be displayed due to a mistake in initial setting, the user is notified of the cause of the error and is invited to fix it by setting the correct folder.

The intent of this function is to handle irregular status on condition that correspondent software is working correctly and capable of responding with precise feedback. It is beyond the scope of this function to recover a system that has fatal errors.

3.3.3 Dynamic Filtering Function

This function judges whether the transmission of events created in the smartphone or transmission of confidential files are appropriate depending on internal and external conditions as detailed in Section 3.4.2. Whether specific application re-

ceives an Intent or not is defined by Android's Permission described in Android Manifest file. As the Android's Permission is statically defined, it is a challenge to provide an access control mechanism that provides both compliance with Android's mechanism and dynamic policy change required in the proposed method.

This function also conduct virus detection for the received files according to the trust level of the external device to prevent illegal privilege elevation.

3.3.4 Message Handling Function

In order to bridge smartphone's coordination protocol and the protocol for communication between the smartphone and the external device, message handling function provides fundamental element in the coordination proxy. Namely, the Intent generated by Android platform is translated to a message that can be interpreted by the external device.

Additionally, as a countermeasure in case the smartphone is deluged with huge number of messages from malicious external device, this function is equipped with flow control function. For example, in case the frequency of the received message surpasses predefined threshold, the messages are discarded so that the processing burden is alleviated, thus preventing denial of service. This simple method is adopted in existing discovery protocols such as rate limiting policy in RFC4066[11].

More sophisticated methods such as client puzzle[12] or puzzle auction[13] are also available as typical measures to reduce the damage caused by DoS attack. The idea of these methods is to require the sender of the message to solve a mathematical puzzle before a connection can be established.

3.4 Implementation on Android Platform

In this section we describe a method to implement coordination proxy on Android platform. The assumption for the security in this implementation is that Android platform is not cracked, working correctly.

Figure 4 shows a screenshot of the coordination process. Though coordination between Android applications is supported by Intent mechanism, additional filtering function is required to realize the proposed dynamic filtering function. The Intent is at first handled by Android platform's Intent resolution mechanism as shown in the left of Figure 4. Secondly, the Intent is handled by the proposed coordination proxy, which is implemented as an Android application to leverage existing Intent mechanism, as shown in the right side of the figure.

3.4.1 Android Manifest

Figure 5 shows how dynamic filtering function is composed. The PC viewer application is implemented as an Android application. When an Intent is sent out from the PC viewer application, it is handled by the Intent Resolution Mechanism provided by Android platform to determine coordination target. The target is chosen based on Intent filter, which describes which kind of Intent is supposed to be received and processed by each Android application.

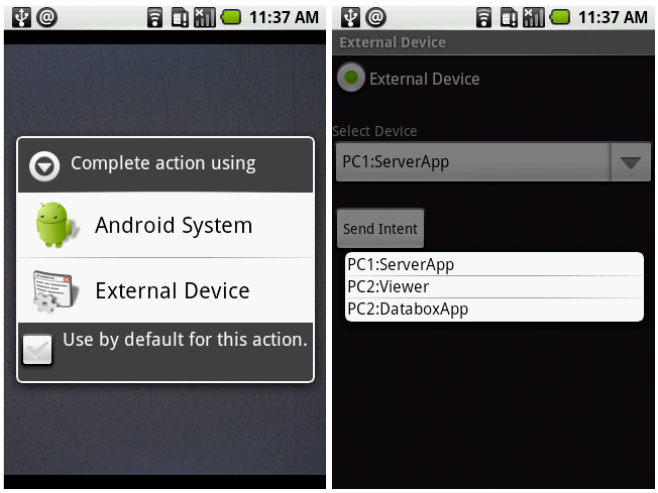

Figure 4: Screenshot of coordination process

Figure 6 shows an example of AndroidManifest.xml for the coordination proxy. Basically, the Intent Filter is defined to receive all the Intent to allow compatibility with future coordination applications.

In order to realize the use case shown in Figure 1, following Intents need to be described: SEND, which is utilized to send a file to a PC, ORIGINAL_PCVIEWER, utilized to receive original Intent from a PC, or ORIGINAL_ANSWER, which is utilized to notify an incoming call to a PC. In the use case depicted in Figure 1, the ANSWER Intent is not broadcasted, and just one application is allowed to handle it. Hence, the incoming call handler application have to be modified so that it generates original Intent to the coordination proxy on receiving ANSWER Intent. In addition, in order to detect internal status of the smartphone for filtering policy change, following Intents need to be included: BATTERY_LOW, which notifies low battery level, and DEVICE_STORAGE_LOW, which shows depletion of storage.

Regarding Permission of the AndroidManifest file, whether specific Permission is required or not depends on the kind of Intent. For coordination proxy, all the required Permission is supposed to be described in the AndroidManifest file. In order to provide flexible access control, additional filtering mechanism is provided in the proposed method as described in the next section.

3.4.2 Dynamic Update of Filtering Policy

Whether coordination with external device is allowed or not is determined depending on a filtering policy. An example of filtering policy description is shown in Figure 7.

The filtering policy is generated by integrating a preinstalled common policy and add-on policies downloaded after purchase. The common policy is defined by an operator or a mobile phone vendor, considering the balance of usability and

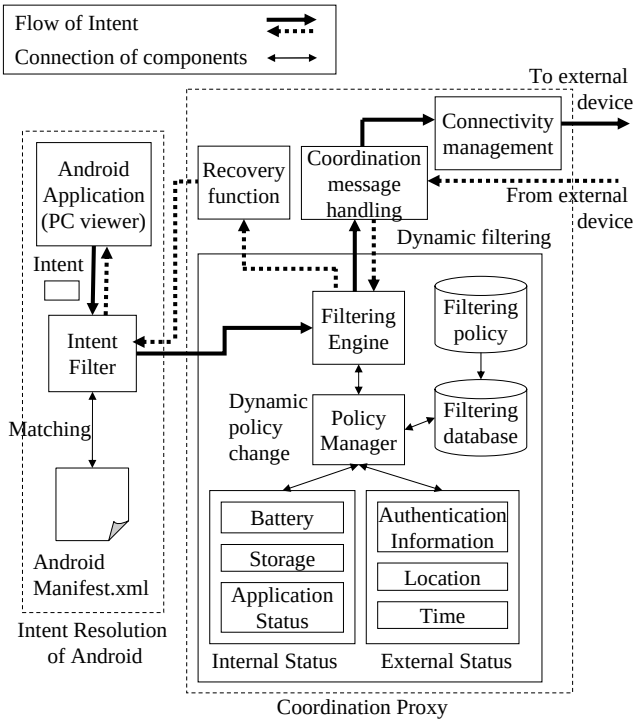

Figure 5: Mechanism of dynamic filtering function

```
<intent-filter>
   <action android:name="android.intent.action.SEND" />
   <action android:name="android.intent.action.ORIGINAL_PCVIEWER" />
   <action android.name="android.intent.action.ORIGINAL_ANSWER" />
   <action android.name="android.intent.action.BATTERY_LOW" />
   <action android.name="android.intent.action.DEVICE_STORAGE_LOW" />
   <category android:name="android.intent.category.DEFAULT" />
   <data android:mimeType="image/jpeg" />
</intent-filter>
```

Figure 6: Android Manifest for the coordination proxy

security. The add-on policy is defined by an administrator in an organization, depending on additional requirements especially focused on security, following governmental or corporate policy.

3.4.3 Filtering Database

The filtering policy is registered to filtering database, which is generated for each filtering conditions(Figure 8). The filtering database manages the permission list that defines a list of Intents which can be transfered to specific application on the external device. The update of filtering policy is immediately reflected to the filtering database.

The basic process of filtering is as follows. When a user selects an external device as a coordination candidate, an Intent is sent to dynamic filtering function of the coordination proxy. The Intent is utilized as a query to the filtering database, and several device options to start transaction are presented to the user.

An appropriate filtering database is dynamically selected

```
<?xml version="1.0" encoding="UTF-8"?>
<filtering_policy>
   <condition>
      <internal_status>          Pre-installed policy
         <battery>HIGH</battery>
         <storage>HIGH</storage>          Add-on policy
         <application>  TRUSTED </application>
      </internal_status>
      <external_status>
         <authentication_level>HIGH</authentication_level>
         <time>BUSINESS_HOUR</time>
         <location>BUSINESS_LOCATION</location>
      </external_status>
   </condition>
   <application_name>ServerApp</application_name>
   <allowed_intent_list>
      <intent>SEND</intent>
      <intent>ORIGINAL_PC_VIEWER</intent>
      <intent>ORIGINAL_ANSWER</intent>
   </allowed_intent_list>
</filtering_policy>
```

Figure 7: Description example of filtering policy

Status		Database No.
Internal	External	
(Condition A)		1
(Condition B)		2

Database No.	Intent	Application	Host
1	SEND	A, B, C	A

Database No.	Intent	Application	Host
2	SEND	A	A

Figure 8: Structure of filtering database

by the policy manager depending on the internal and external status of the terminal. The filtering database is selected considering following conditions.

- Remaining battery or storage space
- Installed software
- Time and location

In order to detect resource-related events, Android platform provides predefined Intents such as BATTERY_LOW, BATTERY_CHANGED, and DEVICE_STORAGE_LOW. Also, for location detection, an Android application with location-detection permission is allowed to get location information based on cellular base station or GPS, which is an appropriate function to realize this filtering policy.

4 EVALUATION

In order to examine the overhead caused by coordination with an external device, we prototyped the coordination proxy based on Android G1[14].

4.1 Evaluation Scenario

Based on the use case shown in Figure 1, we assumed a smartphone is connected to a PC, and a file is sent from the

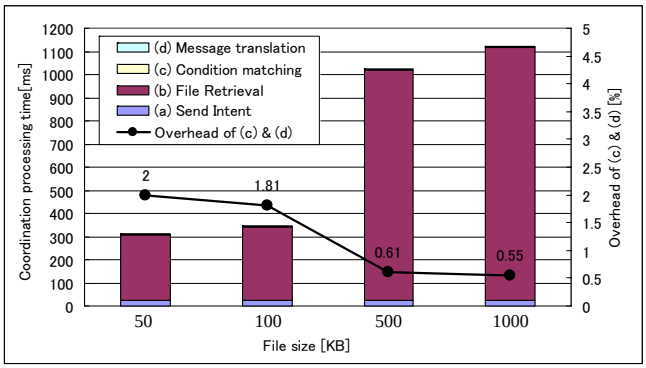

Figure 9: Coordination time in file transfer scenario

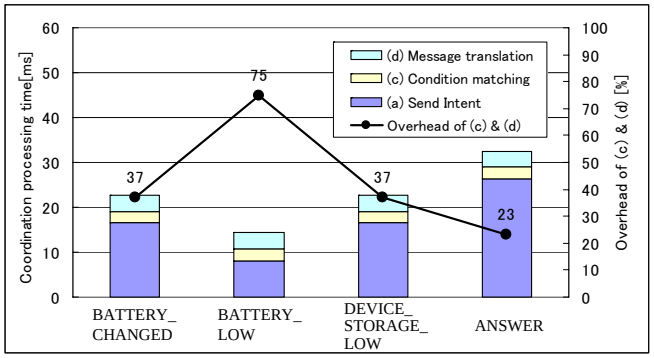

Figure 10: Coordination time in event notification scenario

smartphone to the PC so that the file is presented on a PC's display. The overhead of processing time caused by the coordination proxy is measured against the coordination time of Android applications in single OS. The evaluated part is shown from (a) to (d) in Figure 3.

The process confined in Android terminal is composed of (a) and (b), which is comparison target against the proposed method. Additional overhead created by the proposed method are (c), which is the matching process of filtering database, and (d), which is translation from an Intent to a message.

4.2 Coordination Time

Figure 9 shows the processing time for coordination in the smartphone, and the overhead caused by the proposed method in file transfer scenario. The coordination time is shown as the bar graph, indicating the breakdown of each steps from (a) to (d). The overhead of (c) and (d) against the coordination time in single OS, which is composed of (a) and (b), is shown as the line graph. The file size sent from the smartphone to the external device is changed from 50KB to 1MB.

The processing time is around 2 to 3 ms for each of (c) and (d), which is negligible on the bar graph, and their overhead is equal to or less than 2% as shown in the line graph.

According to the breakdown of the coordination time, the processing for file retrieval is dominant in the file transfer scenario. The processing time of filtering engine and the coordination message translation for the proposed method is vanishingly small.

Figure 10 shows the processing time in event notification scenario for various notification events measured by using predefined Intents. The coordination time for internal applications is shown as (a), whereas the additional processing time for the proposed method is shown as (c) and (d). The overhead of the proposed method is shown by the line graph.

Though the overhead becomes larger compared to the file transfer scenario, the total processing time for the event notification is smaller than 50ms, which is practically negligible and doesn't have any negative impact for usability.

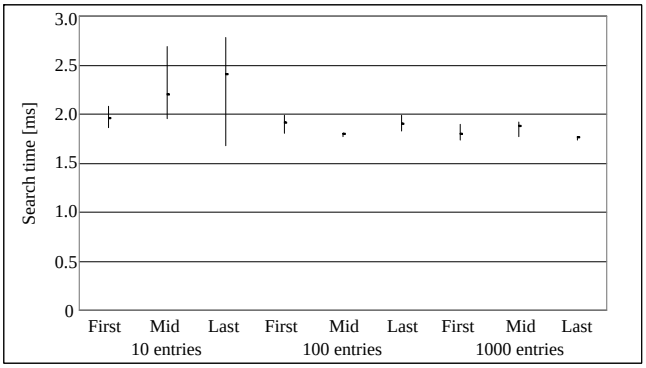

Figure 11: Condition matching time with more entries

4.3 Consideration

4.3.1 Scalability

As the dynamic change of filtering policy is conducted before the coordination sequence starts, the matching process is implemented as simple database search. This architecture is expected to be scalable in case the matching condition becomes complicated or number of coordination application increases.

In the evaluation of Figure 9 and 10, ten entries are registered on the database. Figure 11 shows the database search time when the number of registered entries is increased to 100 and 1,000. For each evaluation, the location of the targeted information in the database is changed at first, last, and in the middle of the entries. The result is less than 3ms for every pattern of the evaluation, which shows that even if the number of external devices is increased to several dozen or more, the coordination time is not affected at all.

The coordination message translation is also implemented as the process to refer to a database and change the received Intent to a message, which is similar to the process of filtering engine. Hence, the impact of increasing the number of command is supposed to have the same tendency. The total number of Intent defined in Android SDK 1.5 is 76, and according to Figure 11, future increase of the number of Intent

has substantially no impact on the coordination time.

4.3.2 Round-trip Time

The evaluation is conducted by measuring internal overhead, which doesn't include message transmission over the air between the smartphone and the PC. We conducted additional experiment by connecting Android G1 and a PC with wireless LAN(802.11n infrastructure mode) to evaluate overall processing time. The total time is increased, whereas the overhead of the proposed method becomes relatively smaller

For file transfer scenario, in case the file size is 100KB, the transmission time over the air is 686ms. As a result, the total processing time becomes 1033ms, and the overhead of the proposed method is 0.6%. For event notification scenario, in case the message size is 128B, the transmission time is 713ms. The total time becomes 769ms, and the overhead is 4.3%. The transmission time includes the socket connection time of 312ms and 335ms for each scenarios. From this result, it is reasonable to provide preceded connection of connectivity management function as described in Section 3.3.1.

On the other hand, among the components of dynamic filtering function, file check module is not implemented in this prototype. In case file check include just simple file format check, the processing time would be small enough. In case it includes virus detection, the overhead becomes larger. According to [15], the throughput of virus detection is 580Mbs on a PC(Pentium4 2.8GHz, RAM 1GB). Provided that the specifications of the smartphone catches up in the near future, time required for virus detection of 1MB file is 13.8ms. It seems to be feasible to include such function in the proposed method.

In addition, the evaluation is conducted on the assumption that wireless LAN interface is enabled both on the smartphone and the PC beforehand. It is a future work to study a protocol to further save the battery by disabling the wireless interface for normal time.

5 CONCLUSION

In order to extend smartphone's flexible application coordination function to external devices, we derived the requirements for the coordination proxy on the smartphone by analyzing use case and security threats. Then we proposed a novel architecture to meet these requirements, whose strong feature includes dynamic filtering function that changes filtering policy considering internal and external status of the smartphone.

An evaluation of a prototype based on Android G1 showed that the overhead of the proposed filtering engine and coordination message translation is equal to or less than 2%. This result shows that the proposed method assures the security requirements derived from the threat analysis with very little overhead.

It is a future work to evaluate the proposed method on more complicated scenario. For example, the smartphone could be connected with several devices at the same time, or conduct more complicated transaction with a device. Also it is a future work to extend the coordination message translation to support standardized protocols such as HTML5, so that various devices other than PCs can be coordination targets.

REFERENCES

[1] H. Karen Lu and Asad M. Ali, A Proxy Agent for Small Network-Enabled Devices, International Performance Computing and Communications Conference 2008(2008).

[2] Hanping Lufei, Weisong Shi, and Vipin Chaudhary, Adaptive Secure Access to Remote Services in Mobile Environments, IEEE Transactions on Services Computing, Volume 1, Issue 1, pp.49-61(2008).

[3] Android Official Website. *http://www.android.com/*

[4] DLNA - connect and enjoy. *http://www.dlna.org/home*

[5] Michael Ripley, C. Brendan S. Traw, Steve Balogh, and Michael Reed, Content Protection in the Digital Home, Intel Technology Journal Volume 06 Issue 04(2002). *http://www.intel.com/technology/itj/2002/volume06issue04/art05_protection/vol6iss4_art05.pdf*

[6] HTML5 Draft Standard. *http://www.whatwg.org/specs/web-apps/current-work/*

[7] Home Gateway Initiative. *http://www.homegatewayinitiative.org/*

[8] A. Delphinanto, B.A.G.Hillen, I.Passchier, B.H.A. van Schoonhoven, and F.T.H. den Hartog, Remote discovery and management of end-user devices in heterogeneous private networks, 6th Annual IEEE Consumer Communications & Networking Conference(2009).

[9] Jung-Tae Kim, Yeon-Joo Oh, Hoon-Ki Lee, Eui-Hyun Paik, and Kwang-Roh Park, Implementation of the DLNA Proxy System for Sharing Home Media Contents, IEEE Transactions on Consumer Electronics(2007).

[10] Frank Swiderski and Window Snyder, Threat Modeling, Microsoft Press(2004).

[11] Marco Liebsch and Ajoy Singh, RFC4066 - Candidate Access Router Discovery, http://www.ietf.org/internet-drafts/draft-ietf-seamoby-card-protocol-07.txt(2004).

[12] Ari Juels and John Brainard, Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks, In Network and Distributed System Security Symposium(1999).

[13] XiaoFeng Wang and Michael, K. Reiter, Defending Against Denial-of-Service Attacks with Puzzle Auctions, Proceedings of the 2003 IEEE Symposium on Security and Privacy(2003).

[14] T-Mobile G1 Specification. *http://www.htc.com/www/product/g1/specification.html*

[15] Guoning Hu and Deepak Venugopal, A Malware Signature Extraction and Detection Method Applied to Mobile Networks, Proceedings of the 26th IEEE International Performance Computing and Communications Conference(2007).