ANDROID SMARTPHONE REMOTE MONITORING APPLICATION USING SMS SERVICE

Hadi Pranoto[1], Eko Budi Setiawan[2]

[1,2] Universitas Komputer Indonesia
Jl. Dipatiukur 112 Bandung
E-mail : hadipranoto96@gmail.com[1], eko@email.unikom.ac.id[2]

ABSTRACT

Android has been updating the system of every version it releases. The addition of Application Programming Interface (API) is done every time Google releases a new Android operating system. The availability of APIs for third-party applications provides opportunities for developers to be able to monitor Android smartphones. Just like Google Device Manager which can instruct Android smartphones over the internet network, however, it still has a deficiency that is if the target smartphone in a state of inactive internet. In this research, the author utilizes SMS media to be able to process instructions and access the system API for monitoring purposes. The results of this research is by using SMS then user can instruct Android smartphone to take photos, get current location, ring, delete smartphone files, set screen protection, and backup contact with higher messaging reliability. This application can run well on Android Lollipop 5.1 (API Level 22) or above because it has enough API to support system functionality.

Index Terms : Android, Application Programming Interface(API), SMS, Remote monitoring, GPS, Device Admin.

1. PREFACE

Android is a collection of open source software that includes operating systems, middleware, and key applications along with a set of Application Programming Interface (API) for writing mobile applications that can shape the look, feel and function of a mobile device [1] . Android has been updating the system of every version it releases. For example, Android Marshmallow (API 23) has provided fingerprint recognition API support that allows users to use their fingerprint to open devices, Play Store authentication, and third-party application authentication. With these features will make users more "aware" of the activities undertaken by the application in the Android system.

Although on the Android operating system has available API to run the functionality of a mobile device, a problem that often arises is when the Android device is lost or left somewhere. Possible action to help locate the presence of device left behind somewhere is by monitoring using apps that can access Android APIs like Android Device Manager provided by Google. However, these methods still have weaknesses in monitoring that is when the device is in a state of inactive data packet internet then Android devices will be more difficult to reach by the user.

Based on the results of data processing questionnaires about behavior patterns in using smartphones distributed to 70 respondents, it is known 18.57% of respondents often use airplane mode and 32.85% of respondents actively use internet data packets when the device is used only. From this it can be concluded some people limit his smartphone in receiving communication messages. And to monitor Android devices using third-party applications required at least internet connection. Various studies have been done before to deal with such weaknesses as research conducted by Kumar and Qadeer [2], Zhang, Hui, Qizhen, and Kim Tai-hoon [3], Kuppusamy and Aghila [4], Rohitaksha, Madhu, Nalini, and Nirupama [5], Punjabi, Pooja, Mantur, and Sneha [6], Gupta and Reddy [7], Dave and Welekar [8], Chandran [9], using Short Message Service (SMS) service to control distance Much as did Rayarikar, Sanket, and Pimpale [10], and Jongseok and Kim Howon [11]. However, in the research done by Kumar and Qadeer [2] and Zhang, Hui, Qizhen, and Kim Tai-hoon [3] the message used to perform the monitoring has not been using encryption so it must include the password and the keyword plain text. Then in the research Kuppusamy and Aghila [4] have been using the process of encryption and decryption in the system, but not using the server so that the response that can be accepted by the user only in the form of text-based information.

Another problem can also arise when a smartphone is lost. Someone can turn off the power of the smartphone. This happens because the system on Android allows the action to power off directly by using the power off physical button. If the smartphone has been in a state of death then someone can just change the SIM card or even factory reset. If the SIM Card is replaced it will be a problem because the smartphone can not be monitored either using the internet or via SMS.

Thus, the applications to be built in this study are expected to be able to overcome the limitations of the Internet in conveying monitoring instructions so as to help improve the reliability of information delivery using SMS services.

2. CONTENTS OF RESEARCH

2.1 Android Application Components

The Android app is written in the Java programming language. Java code is compiled together with the data files required by the application where the process is packaged by tools called "apt tools" into the Android package so as to generate files with apk extension. Apk file is what we call the application and can later be installed on mobile devices [19]. There are four types of components in the Android app that is activity, service, broadcast receiver, and content provider.

Activity will present User Interface (UI) to user so user can do interaction. An Android app may have only one activity, but most applications have many activities that depend on the application's purpose and the design of the app itself. To move from one activity to another activity can be done using a trigger such as click the button on the application screen.

Service does not have a Graphic User Interface (GUI), but the service runs in the background. For example in playing music, the service may play music or retrieve data from the network, but each service must be in its parent class. When a music player is playing a song from an existing list, the app will have two or more activities that allow the user to play while selecting a new song. To keep music running an activity can run the service.

Broadcast receiver functions to receive and react to deliver notifications. Examples of broadcasts such as timezone notifications have changed, batteries are weak, images have been taken by the camera, and others. Applications can also initialize a broadcast such as providing information on other applications that there is data that has been downloaded to the device and ready for use.

Content provider creates a specific set of application data so that it can be used by other applications. Data is stored in system files such as SQLite database. Content provider provides a way to access data required by an activity.

In this study will use all these components. Activity used by author to interact with user. Service is used by the author to execute commands based on certain triggers. Broadcast receiver is used to listen to incoming SMS and detect sim card change. The content provider is used speci fi cally like reading the id of the sim card

2.2 System Overview

An overview of the application to be made can be seen in Figure 1.

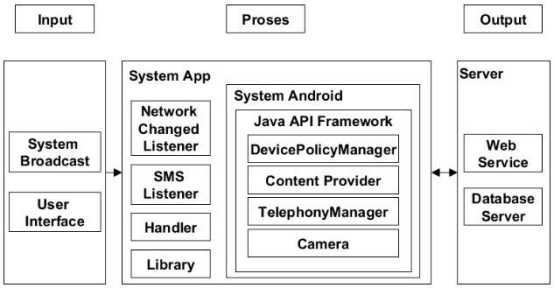


Figure 1. System Overview

In general the workings of Android smartphone monitoring system using SMS can be explained as follows: input comes from user interface (user interface) or from active Android system broadcast message. Upon receipt of input, the application system will communicate with the Application Programming Interface (API) on Android smartphones aimed at accessing camera resources, contacts, GPS, and others to process inputs. After successfully processing the input, when generating data output then the data will be sent to the server.

2.3 System Analysis and Planning

The system to be built is the application of remote monitoring Android smartphone using Short Message Service (SMS) service. The system to be built can be seen in Figure 2 below.

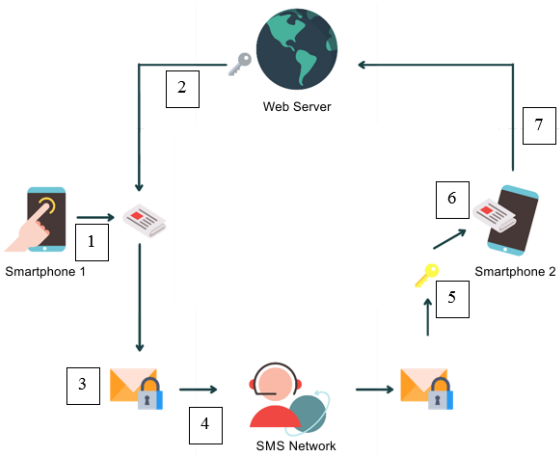


Figure 2. The Concept Of The System To Be Built

The system model built has the following stages to monitor Android smartphone using SMS service:

a. Stages of choosing instruction. Users will choose the features provided by the system and then will be formed plain text in the form of URL with parameters.

b. The public key reading process of the server that the user gets when logging in using his smartphone.

c. The plaintext encryption process becomes ciphertext using a public key that has been previously obtained. After encryption is done the addition of the '$$' validation symbol at the

beginning of the text to recognize whether the SMS is an instructional SMS or not.
d. The process of sending SMS encrypted instructions via SMS network. The built system will charge the cost of sending SMS to the users as much as one SMS.
e. The ciphertext decryption process uses the private key obtained simultaneously with the public key when the user logs into the system. If the message is decrypted with a different key pair then the system can not execute the monitoring instructions.
f. The process of reading the parameters that exist in the URL stored in the plaintext to then executed the command.
g. The process of uploading content when monitoring instructions is to capture photos, current location, or back up contacts.

2.4 Selection of monitoring instruction menu

In the system to be built there is an instruction menu that can be used for monitoring process Android smartphone. The list of instances that can be used to monitor Android smartphones can be seen in Table 1 below.

Table 1. List of Monitoring Instructions

Instructions	Explanation
action_ available_ camera	Access the available camera to take photos.
action_current_ location	Search for the latest Android smartphone location.
action_ringing	Draining Android smartphone.
action_erase_ external	Delete the files stored in the external directory.
action_wipe	Perform wipe on Android smartphone.
action_lock	Locks the Android smartphone according to the settings applied by the user.
action_lock_ new_password	Resetting user's Android smartphone protection.
action_get_ all_contacts	Back up smartphone contacts
action_get_ sim_contacts	Back up the contacts stored on the SIM Card smartphone
sim_card_ changed	Used to detect the change of SIM Card

2.5 List of system authorization requirements

Each feature provided has different resource requirements. At this stage is to define what access is required for system functionality to run. The need for system permissions can be seen in Table 2 below.

Table 2. List of Instruction Access Requirements

Instructions	Access Requirements
action_ available_ camera	Access camera hardware, internet data packets, storage
action_current_ location	Access GPS, internet data packets
action_ringing	Audio access
action_erase_ external	Storage access
action_wipe	Device admin access
action_lock	Device admin access
action_lock_ new_password	Device admin access
action_get_ all_contacts	Access contacts reading, internet data plan
action_get_ sim_contacts	Akses baca kontak, paket data internet
sim_card_ changed	Access read contacts, access send SMS messages

2.6 Formation of Instruction Objects

The instruction will then be formed into a resource accessible object. The object form used in the system is using the URL format so that it can accommodate the parameters. These parameters will be filled by the type of instructions and messages the user wants to convey to the target smartphone. In table 3 the following are instructions that have been formed into URL objects.

Table 3. Example of URL-Based Instructions

Instructions (URL Format)
https://catch.id/?action=action_available_camera
https://catch.id/?action=action_current_location
https://catch.id/?action=action_ringing
https://catch.id/?action=action_erase_external
https://catch.id/?action=action_wipe
https://catch.id/?action=action_lock
https://catch.id/action= action_lock_new_password&msg=pass
https://catch.id/?action=action_get_all_contacts
https://catch.id/?action=action_get_sim_contacts

After the instruction is converted into a form that is accessible to the resource then the next step is how to secure the instruction. One way that can be used is to use cryptography. In this research cryptography method used is RSA (Rivest, Shamir, Adleman). The RSA method works using asymmetric keys that can be used for systems involving multiple parties for the encryption process. In the system built using RSA 512 bit method which can accommodate plaintext along 64 characters. In table 4 is an example of ciphertext output from the

image capture instruction generated through RSA 512 bit encryption method that is as follows.

Table 4. Example of Instruction Encryption Process

Instructions (URL Format)	**Cipher Text**
https://catch.id/?action=action_available_camera	NjqsFhXSZN5JF/QdsavWYqJKNrAzLV4ymOuMpePcA/lwIqR+BlRmAYekshbKIcaPXO2+IgIum/oZ5RMzMEEWUQ==

Cipher text in table 4 is not directly sent to the target smartphone. Cipher text will be inserted symbol "$$" in advance so that SMS validation can be done. If this is not done then the system will try to decrypt on every incoming SMS. In table 5 is an example of the final instruction to access the front camera ready to be sent to the target smartphone.

Table 5. Sample Results End Message Instructions

The symbol "$$" + *Cipher Text*
$$NjqsFhXSZN5JF/QdsavWYqJKNrAzLV4ymOuMpePcA/lwIqR+BlRmAYekshbKIcaPXO2+IgIum/oZ5RMzMEEWUQ==

2.7 Event Listener Analysis On Monitoring System

When the system on Android receive incoming SMS messages then actually the message will be broadcasted to all applications that use SMS listener feature. Furthermore, the application will try to sort whether the SMS message has something to do with the system built or not. If the SMS contains the symbol "$$" it will be decrypted SMS messages starting from the third letter after the symbol "$$". Decryption is done using private key so as to generate plain text in the form of URL which can be accessed by its resources. The system will try to read what parameters are attached to the URL of the decryption and then process according to the algorithm of each instruction.

The same is true for the SIM card replacement notification feature. When there is a change in smartphone status such as from airplane mode to be connected to the internet then the Android system will broadcast messages to all applications that can listen to the event. Next the application will try to read the unique ID on the SIM card and compare with previous data to know whether there is a change of SIM card or not. As in Table 6 it discusses the listener on the system.

Table 6. *Event And Listener On Monitoring System*

Event	**Listener**
action_available_camera	1. Indexing the number of cameras available. 2. Open camera hardware access 3. Take a photo 4. Save the photos on external storage 5. Upload photos to the server 6. Repeat step 1 until the entire camera is accessed.
action_current_location	1. Check the GPS status. 2. Make a connection using GoogleApiClient and search for coordinates now 3. Upload information to the server in the form of latitude and longitude if available.
action_ringing	1. Play the ringtone from the system
action_erase_external	1. Take the external directory 2. Perform a recursive deletion
action_wipe	1. Check whether there is admin device permissions or not 2. Do wipe if it is known to have obtained permission
action_lock_now	1. Check whether there is admin device permissions or not 2. Lock the smartphone if it is known to have obtained permission
action_lock_now_new_password	1. Check whether there is admin device permissions or not 2. Update the password if it is known to have obtained permission
action_get_all_contacts	1. Read smartphone contacts 2. Create a file and fill it with read contact data 3. Upload file to server
action_get_sim_contacts	1. Read contact data on SIM Card 2. Create a file and fill it with read contact data 3. Upload the file to the server
sim_card_	1. Read the ID on the

Event	Listener
`changed`	SIM Card 2. Compare with previous ID 3. In case of change then send notification to the trusted number if available

2.8 Use Case Diagram Monitoring System

Use case diagram of monitoring system consists of 19 use case that is make account, help signing in, log in, log out, edit profile, create sms monitoring instruction, scan QRcode, show remote monitoring result, broadcast receiver filter, SIM card change notification, reset password, wipe smartphone, smartphone key, ringing, delete directory files, take photos, back up contacts, get current location. The use case diagram can be seen in Figure 3 below.

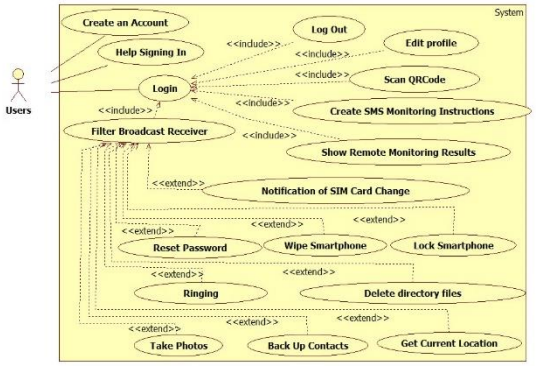


Figure 3. Use Case Diagram Monitoring System

2.9 Hardware Implementation

The hardware used to test the application is as follows:

Table 7. Hardware Implementation Testers

No.	Hardware	Specification
1.	*Processor*	Quad-core with 1.2 GHz speed
2.	RAM	3 GB
3.	Layar	5 *inch*
4.	Storage	40 MB
5.	Kamera	16 MP & 3.7 MP

2.10 Software Implementation

The software used to test the application is as follows:

Table 8. Software Builder Implementation

No.	Software	Specification
1.	Computer Operating System	Windows 8.1 Pro
2.	Android Studio	Versi 2.3.2
3.	Android SDK	Level 22 and above
4.	Programming language	Java, PHP, JSON
5.	Database	*MySQL*
6.	Operating system Smartphone	Android Lollipop 5.1 (API Level 22) and above

2.11 Interface Implementation

The interface is a means for users to interact with a system. Implementation of the interface of the Android smartphone remote monitoring application using Short Message Service (SMS) service is as follows.

a. Log In Display Interface

The login display interface is used to log into the system using a username and password.

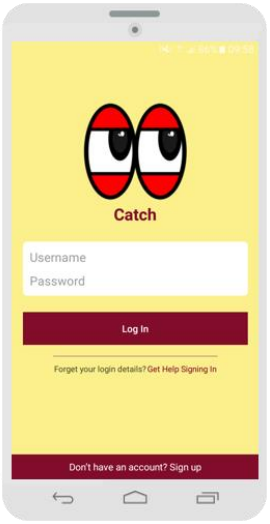


Figure 4. Log In Display Interface

b. Home Display Interface

The home display interface is used to view the monitoring results of images, locations, and contacts.

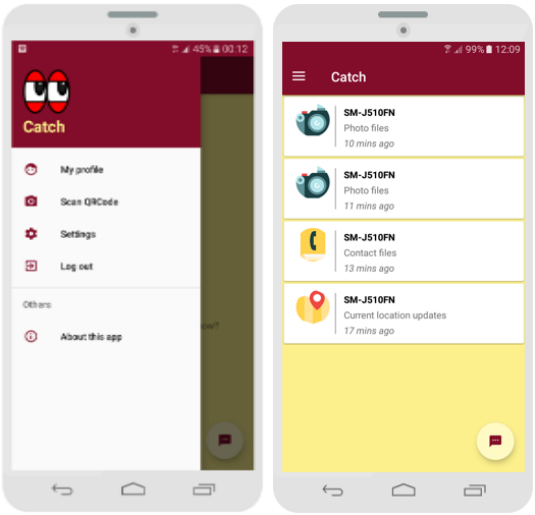


Figure 5. Home Display Interface

c. Create SMS Instruction Interface

This interface is used to select what instructions and choose which smartphone to monitor.

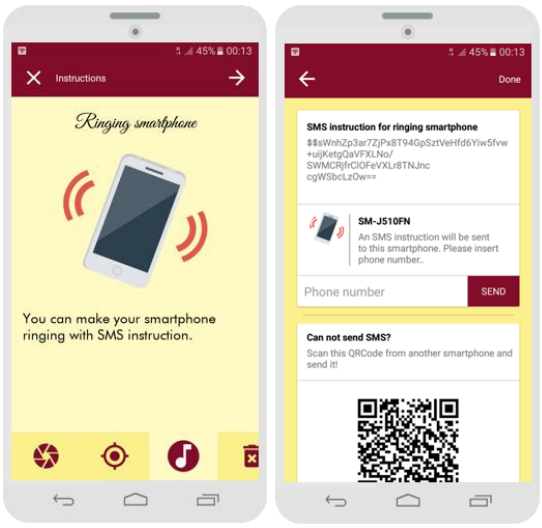


Figure 6. Create SMS Instruction Interface

2.12 System Testing

Testing is done in order to find errors and deficiencies in the software being tested. With the testing then it can know whether the software is made already meet the criteria in accordance with the purpose of software design. Testing is done by black box testing techniques that focus more on finding functional fault program.

Black box testing is performed for authentication functional components, broadcast message filters, SMS instruction making, profile editing, SMS sending, settings, and QRCode image scans. In this test is done by using seven different smartphone to measure the accuracy of application success. The percentage of success of applications on different devices can be seen in table 9 below.

Table 9. Results of Tests Against Smartphones

Smartphone	**API Level**	**Successness**
Samsung A3 2017	23	100%
Sony E5	22	98%
Xiaomi Redmi 4	23	100%
Samsung J5 2016	23	100%
Sony C4	23	100%
Xiaomi 5	23	100%
Samsung J7 Prime	23	100%

Based on the results of testing of several android smartphone it can be concluded that the application can run on different smartphones of the type, it's just that there are weaknesses of the Sony E5 smartphone is not supported with a complete API on its Android Lollipop operating system which causes the application to crash when reading Contact information. In addition to the shortcomings it was found that all functional applications can run well on seven android smartphone used when testing.

In addition, testing was also conducted in August 2017 by distributing questionnaires to 17 respondents using smartphones with the Android operating system Lollipop (API Level 22) and above. Respondents provide varied answers, but from the average value obtained can be concluded that the respondents agree with the existence of this application facilitate in monitoring its Android smartphone.

Table 10. Beta Test Results

The first question: Does the presence of remote monitoring application Android smartphone using SMS service can facilitate you in monitoring Android smartphone?				
SS	S	RR	KS	TS
8	8	1	0	0
Average = (75/85) *100% = 88.2%				
The second question: Is the presence of remote monitoring application Android smartphone using SMS service can facilitate you in getting notification of change of SIM card (SIM Card)?				
SS	S	RR	KS	TS
3	12	2	0	0
Average = (69/85) *100% = 81.1%				
The third question: Does the remote monitoring application of Android smartphones use this easy SMS service when in use?				
SS	S	RR	KS	TS

4	13	0	0	0
Average = (72/85) *100% = 84.7%				
The fourth question: Does the remote monitoring application of Android smartphones using this SMS service have an appealing look?				
SS	S	RR	KS	TS
3	11	3	0	0
Average = (68/85) *100% = 80%				

Based on the results of beta test calculations that have been presented in table 10 it can be taken some conclusions of test results are :

a. Users find it helpful in monitoring their Android smartphone. Due to the presence of this application then the limitations of internet media in interacting with smartphones for monitoring purposes can be equipped
b. Users feel helped by the notification change feature of his Android smartphone. Due to the presence of this application then the replacement of an unwanted SIM Card can be avoided
c. This application is easy to use because it uses the same concept as making SMS in general
d. This app has an interesting look

3. CLOSING

3.1 Conclusion

Based on the results of research, analysis, system design, and implementation and system testing, the following conclusions can be drawn :

a. Applications that have been built to facilitate users in performing remote monitoring Android smartphone because with the presence of this application then the limitations of internet media in interacting with smartphones for monitoring purposes can be equipped.
b. Applications that have been built can help in providing a notification when there is a change of SIM Card so that the replacement of unwanted SIM Card can be avoided.

3.2 Suggestion

In this research needs to be done further development and refinement in order to improve system security and user privacy involved in it. The suggestions on the development of this application for the future is as follows :

a. The application development on smartphones that have Android operating system architecture is different from that unveiled by Google because the app is not fully suitable for some Android devices on the market.
b. An adjustment is required if a new operating system has been released because the app may find it hard to reach an API that works to access Android smartphone hardware

ACKNOWLEDGMENTS

The author would like to thank as much as possible to:

a. Friends of alumni of SMAN 27 Bandung include Azhar Hidayatullah, Seftian Irhaf, Anggianto, and especially Regie Aditya Wibawa who has allowed Android device used for application development during this research.
b. Fellow Divisi Hardware include Abdul Rohim, Risqi Windu, M. Azmi Burhan, Dicky Armansyah who has helped succeed in publication of the application process until it can be used by the public.
c. The colleagues of the informatics engineering program include Zamaludin Abdullah, Farid Yahya, Dodi Aprianto, Siddik Tauhid Winarya who has supported during the research.

With the help of colleagues finally the author can complete this research as it should.

REFERENCES

[1] R. Meier and W. , Professional Android Application Development, First Edition ed.

[2] D. Kumar and M. A. Qadeer, "SMS Based Emerging Techniques for Monitoring and Controlling Android Mobiles," *IACSIT International Journal of Engineering and Technology*, vol. 4, p. 6, 2012.

[3] W. Zhang, H. He, Q. Zhang and T.-h. Kim, "Protecting User Privacy on the Android-Based Mobile Platform," *International Journal of Distributed Sensor Networks*, p. 10, 2014.

[4] K. Kuppusamy, R. Senthilraja and G. Aghila, "A MODEL FOR REMOTE ACCESS AND PROTECTION OF SMARTPHONES USING SHORT MESSAGE SERVICE," *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, 2012.

[5] K. Rohitaksha, C. Madhu, B. Nalini and C. Nirupama, "Android Application for Vehicle Theft Prevention and Tracking System," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, p. 3, 2014.

[6] K. Punjabi, P. Bolaj, P. Mantur and S. Wali, "Bus Locator via SMS Using Android Application," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5, p. 2, 2014.

[7] R. Gupta and B. Reddy, "GPS and GPRS Based Cost Effective Human Tracking System Using Mobile Phones," VIEWPOINT, vol. 2, p. 1, 2011.

[8] V. Dave and A. Welekar, "Design of an

Android Application to provide Emergency Service," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 3, 2015.

[9] A. Chandran, "Smartphone Monitoring System," *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 4, p. 4, 2013.

[10] R. Rayarikar, S. Upadhyay and P. Pimpale, "SMS Encryption using AES Algorithm on Android," *International Journal of Computer Applications*, vol. 50, p. 19, 2012.

[11] J. Choi and H. Kim, "A Novel Approach for SMS security," *International Journal of Security and its Applications*, 2012.