박사 학위논문

Ph. D. Dissertation

무선네트워크 상에서 위치정보를 악용한 분산서비스거부공격 및 완화방법

DDoS Attack Exploiting Location Information and its Mitigation over Wireless Networks

이 현 록 (李 炫 錄 Lee, Hyunrok)

전산학과

Department of Computer Science

KAIST

2013

무선네트워크 상에서 위치정보를 악용한 분산서비스거부공격 및 완화방법

DDoS Attack Exploiting Location Information and its Mitigation over Wireless Networks

DDoS Attack Exploiting Location Information and its Mitigation over Wireless Networks

Advisor : Professor Kim, Kwangjo

by

Lee, Hyunrok
Department of Computer Science
KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science . The study was conducted in accordance with Code of Research Ethics[1].

2012. 12. 5.

Approved by

Professor Kim, Kwangjo

[Advisor]

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

무선네트워크 상에서 위치정보를 악용한 분산서비스거부공격 및 완화방법

이현록

위 논문은 한국과학기술원 박사학위논문으로
학위논문심사위원회에서 심사 통과하였음.

2012년 11월 21일

심사위원장	김광조	(인)
심사위원	김대영	(인)
심사위원	김명철	(인)
심사위원	이영희	(인)
심사위원	하정석	(인)

DCS
20045324

이 현 록. Lee, Hyunrok. DDoS Attack Exploiting Location Information and its Mitigation over Wireless Networks. 무선네트워크 상에서 위치정보를 악용한 분산서비스거부공격 및 완화방법. Department of Computer Science . 2013. 54p. Advisor Prof. Kim, Kwangjo. Text in English.

ABSTRACT

The explosive proliferation of the smart mobile devices drives massive growth in the wireless data communication. In order to provide wireless data communication for the smart hand-held devices, a large number of base stations including access points (APs) based on IEEE 802.11 (WiFi) have been densely deployed. While widespread deployment and expansion of wireless network infrastructure brings the wireless data communication ubiquitously, it has also introduced many security challenges such as Denial of Service (DoS), malicious mobile code, *etc.*, surrounding wireless networks.

In this dissertation, we present a novel Distributed Denial of Service (DDoS) attack model exploiting smart mobile device's location information simultaneously with utilizing active scanning vulnerability in wireless network. Also we provide the simulation result of this attack model in order to evaluate a proof-of-concept of our attack model and show that relatively small number of attacker with tiny volume of active scanning traffic is sufficient to totally interrupt wireless communication. Also we propose a location information protection method based on chameleon hash scheme for a mitigation method against our DDoS attack model.

For constructing our DDoS attack model, an attacker who desires to aggress on the wireless network propagates a malicious code in users' smart mobile device. It becomes a zombie device of botnets which is stealthy controlled by C&C servers. After the botnet master issues the attack command, the zombie devices activate their smart active discovery attack, in case of WiFi probe request which is a basic attack unit to evaluate our model. The decisive distinction between existing active discovery attack such as Probe Request Flooding (PRF) and our attack is that there is no clear classification to decide whether normal active discovery protocol or not, due to the tiny volume traffic and reasonable interval of the request message. Furthermore, our attack can be forcefully performed in targeted specific region, where is a primary target of the attacker such as government organization, bank, *etc.*, based on the location information of user without extra device like a jammer.

In order to mitigate our attack model, we need location information protection for user to prevent the attacker from clear target of the region specific setting. Previous researches on location privacy protection method can be largely categorized into location obfuscation and k-anonymity with additional trusted third party (TTP). However, previous work have no rigorous proof of security, then the information can be relatively easy to reveal. Because an additional TTP must be involved in the architecture of location privacy protection methods in most researches, a user's location information should be revealed to the TTP. Thus, we propose secure and effective location information protection method based on chameleon hash which never reveals accurate location of user to anybody, and also we present security and performance analysis of our proposed method.

Contents

List of Tables

List of Figures

Chapter 1. Introduction

The explosive proliferation of the smart mobile devices such as smartphones, tablet PCs, *etc.*, have been increasing the traffic of the wireless data communication. In order to provide the ubiquitous communication, a large number of wireless base stations have been densely deployed. Telecommunications companies have been spreading their base stations for the coverage of the cellular data network such as 3G and 4G. Besides, Internet Service Providers (ISPs), enterprises and individuals also have been installing Access Points (APs) based on IEEE 802.11 standard (WiFi) [7]. While dense deployment and expansion of wireless network infrastructure brings high accessibility, it has also introduced many security challenges surrounding wireless network environments. The wireless network, which typically has narrow bandwidth, is to be more vulnerable to eavesdropping, illegal access, jamming, denial of service and so on. Also the smart mobile devices, which have more powerful computing resources then legacy mobile devices, suffer from malicious mobile codes for stealing user information and constructing botnet.

In this chapter, we firstly introduce the current status of WLAN deployment, and then describe discovery protocols in Media Access Control (MAC) layer used by the IEEE 802.11 standard with its well-known vulnerabilities. The motivation and contributions, and organization of this dissertation will be followed.

1.1 Status of WLAN deployment

In order to support wireless data communication, APs based on IEEE 802.11, called Wireless Area Network (WLAN) or hotspot, have been densely deployed in convenient public locations such as a department store, crowded street, coffee shop, restaurant, airport, university campus, *etc.* According to the control method of APs, WLAN can be categorized largely into uncontrolled and controlled WLAN; the important properties of each type of WLAN are as follows:

- **Uncontrolled WLAN**
  - Type of deployment: Non-planned deployment of AP.
  - SSID assignment: Arbitrary named SSID.
  - Users: Individual, small company.
  - Channel assignment: Arbitrary radio channel. Users can freely tune to the channel.
  - Management: Individual manages his own AP policy. A simple management firmware embedded on AP is used.

- **Controlled WLAN**
  - Type of deployment: Planned deployment of group of APs.
  - SSID assignment: Single named SSID.
  - Users: University campus, big company.
  - Channel assignment: Planned radio channel. Users cannot tune to the channel.
  - Management: Network administrator manages all AP policy including user access privilege and channel control. Normally an automated central network management software is used.

In the past, network experts could manage the dense deployments of wireless networks. However, the rapid deployment of low cost devices including AP and smartphone is changing quickly the wireless environment from controlled to uncontrolled. Both of WLAN types can adopt own network policy either open or closed network. In case of open network, any device of certain user can freely join the WLAN without any restriction. Closed network, on the other hand, cannot allow unauthorized access. A user who wants to protect his network adopt closed network policy utilizing SSID hiding, MAC address authentication, WEP, WPA and WPA2.

A location-based service company, JiWire [47], collects and manages the information about WiFi hotspots of various countries and cities. Table 1.1 shows current number of APs in Republic of Korea, where number of deployed APs are ranked first place all around the world under their survey. Fig. 1.1 illustrates an example of WiFi map including paid and free WiFi hotspots in Seoul, Republic of Korea. Although private hotspots such as enterprise hotspots, personal hotspots, *etc.*, are excluded in this example, the APs are densely deployed.

1.2 Discovery Protocol on IEEE 802.11 and its Vulnerabilities

This section introduces discovery protocols which consist of scanning, authentication and association protocol with the MAC frames. We then identify its vulnerabilities aimed at flooding attack.

1.2.1 MAC layer on IEEE 802.11

The MAC layer of the IEEE 802.11 standards defines the ad hoc mode and the infrastructure mode. The former is that STA communication directly with each other, and the latter is that all communication perform via a fixed AP. In this dissertation, our focus is the infrastructure mode of operation in the IEEE 802.11 ranging from a Basic Service Set (BSS) and Extended Service Set (ESS).

There are three major frame types that are data, control and management frames used in the MAC layer of IEEE 802.11 wireless networks. An actual data can be carried by the data frames from station to station. The control frames provide channel acquisition, carrier sensing, power saving, and reliability of MAC. The management frames perform joining and leaving functions among APs. Table 1.2 shows

Table 1.1: Number of APs in Republic of Korea

Region	No. of APs
Seoul	54,591
Gyeonggi-do	34,149
Incheon	16,062
Busan	15,080
Daegu	10,382
Gyeongsangnam-do	8,177
Gyeongsangbuk-do	7,859
Chungcheongnam-do	7,213
Daejeon	7,032
Jeollabuk-do	6,355
Gangwon-do	5,383
Chungcheongbuk-do	5,037
Jeollanam-do	3,991
Ulsan	2,692
Jeju	2,584
other	1
Total	186,759

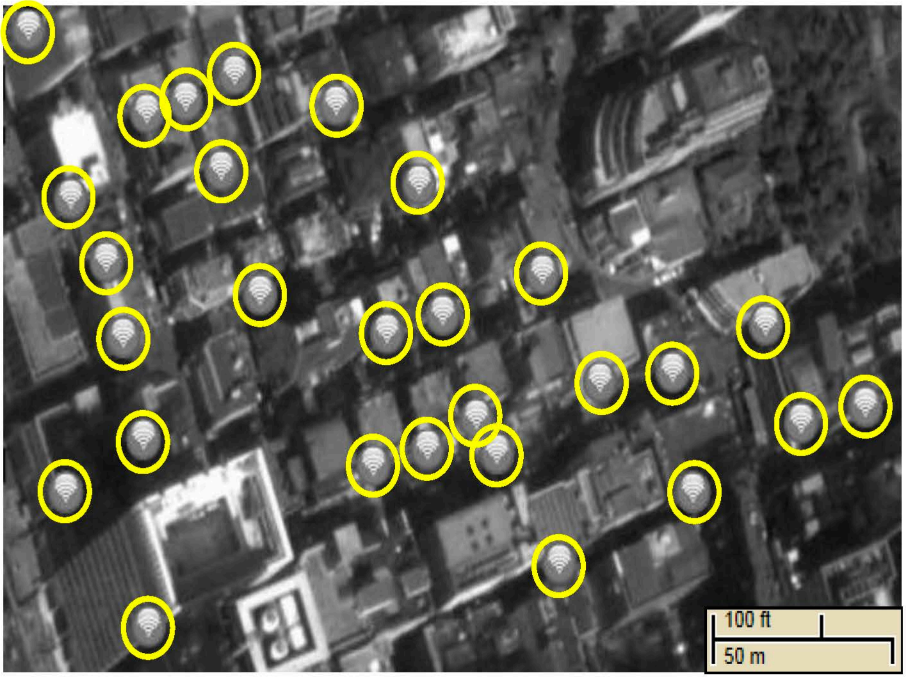


Figure 1.1: An example of street-level WiFi map (Seoul, ROK)

list of major functions of each frames, which can be a candidate to be exploited by adversaries in order to perform DoS & DDoS against the IEEE 802.11 wireless networks. In this dissertation, we focus on the vulnerabilities on the management frames. For the clear understanding for MAC frames in detail, Tables 1.3 and 1.4 show the format and the frame control of IEEE 802.11 packet.

Table 1.2: List of major functions of IEEE 802.11 frames

Data	Control	Management
Data Data+CF-ACK Data+CF-Poll Data+CF-ACK+CF-Poll Null Function CF-ACK (no data) CF-Poll (no data) CF-ACK+CF+Poll	RTS CTS ACK PS-Poll CF-End & CF-End ACK	Beacon Probe Request & Response Authentication Request & Response Association Request & Response Reassociation Request & Response Disassociation Announcement Traffic Indication Message (ATIM)

Table 1.3: MAC Frame Format of IEEE 802.11

	Frame Control	Duration ID	Address1 (Source)	Address2 (Destination)	Address3 (RX Node)	Sequence Control	Address4 (TX Node)	Data	FCS
Bytes	2	2	6	6	6	2	6	0-2,312	4

Table 1.4: MAC Frame Control of IEEE 802.11

Field	Bits	Description
Frame Control	15-14	Protocol version,
	13-12	Type
	11-8	Subtype
	7	To DS.
	6	From DS.
	5	More Frag.
	4	Retry
	3	Power Mgmt.
	2	More Data
	1	WEP
	0	Order

1.2.2 Discovery Protocol

Before establishing wireless data communication channel on IEEE 802.11, user's station (STA) should discover the information of access point (AP), which includes ESSID, BSSID, channel, bit rate, encryption algorithm, *etc.* Discovery protocol of STA involves 3 steps: 1) active or passive scanning, 2) authentication, and 3) association as shown in Fig. 1.2.

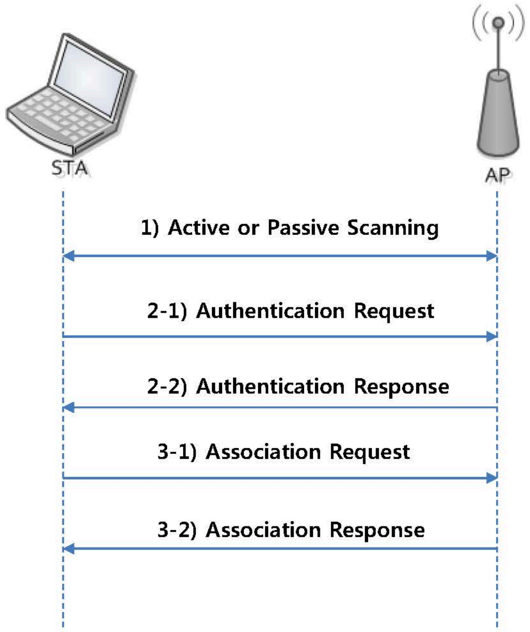

STA
AP
1) Active or Passive Scanning
2-1) Authentication Request
2-2) Authentication Response
3-1) Association Request
3-2) Association Response

Figure 1.2: Discovery protocol on IEEE 802.11

A STA can discover AP information in two methods: 1) passive scanning and 2) active scanning. Firstly, AP broadcasts periodically a beacon frame, and STA receives the information of AP through listening to the beacon in the passive scanning as shown in Fig. 1.3.

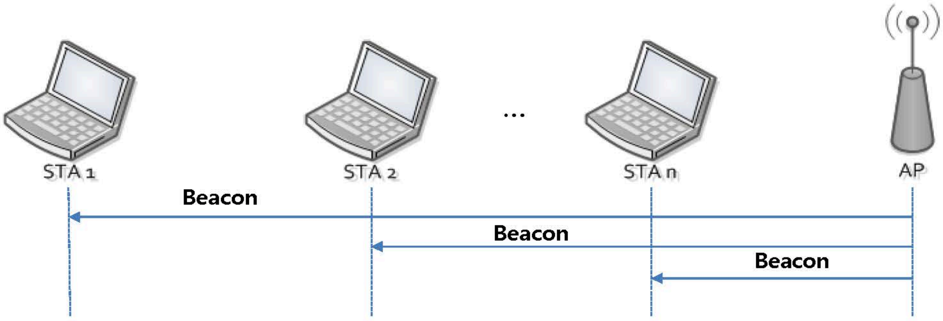


Figure 1.3: Passive Scanning

Second, STAs can directly request the information of APs through probe request & response protocol, called as active scanning. The STA sends a probe request packet, and then the AP sends a probe response packet through available channels. After that the STA receives the response packet, and then confirms the information of the AP. The transmission method of probe request can be divided into directed and broadcast methods. The broadcast probe request is frequently used for scanning all APs by sending a probe request packet with the null value in the SSID field as shown in Fig. 1.4. Fig. 1.5 shows the active scanning with SSID which is performed at re-association request by previously associated STA with a specified AP already having the SSID. After discovering protocol, authentication and association protocols are executed between STA and AP in BSS.

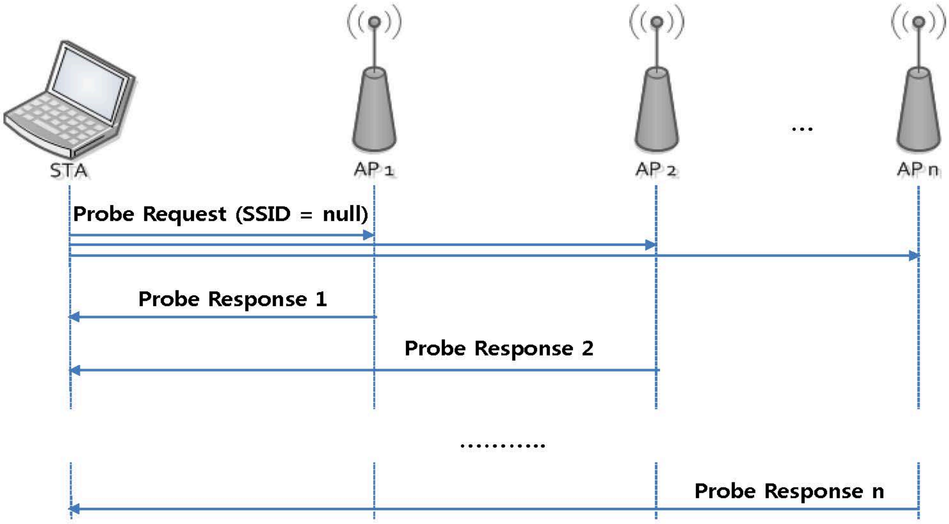


Figure 1.4: Active Scanning without SSID

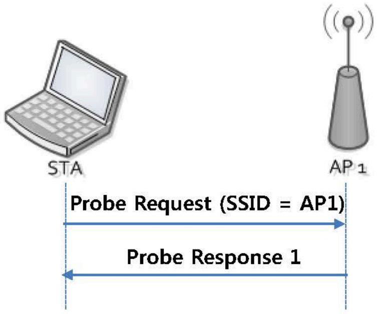


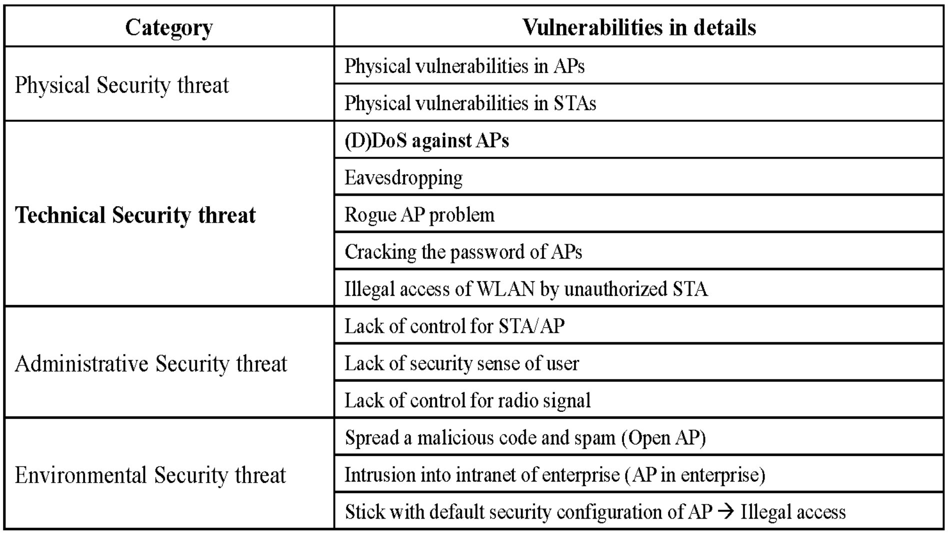
Figure 1.5: Active Scanning with SSID

1.2.3 Vulnerabilities

As shown in Table 1.5, overall vulnerabilities on IEEE 802.11 WLAN include so many security threats including physical, technical, administrative and environmental security threats. In this dissertation, we focus on the technical security threats, especially DoS & DDoS against APs.

As describing before, wireless connection establishment between mobile station (STA) and base station (BS), a discovery or scanning protocol is an essential prerequisite at the first step. The active discovery can be a major cause of the flooding attack, which depletes network bandwidth resources. In case of WiFi, Probe Request Flooding (PRF) in the active scan mode is one of widely known Denial of Service (DoS) attacks. And also there exist the potential possibilities of such flooding attack in any kind of wireless network supporting active discovery, due to STA's demand for receiving quick response to the connection establishment. Also another resource depletion attacks such as Authentication Request

Table 1.5: Vulnerabilities on IEEE 802.11 WLAN



Category	Vulnerabilities in details
Physical Security threat	Physical vulnerabilities in APs
	Physical vulnerabilities in STAs
Technical Security threat	**(D)DoS against APs**
	Eavesdropping
	Rogue AP problem
	Cracking the password of APs
	Illegal access of WLAN by unauthorized STA
Administrative Security threat	Lack of control for STA/AP
	Lack of security sense of user
	Lack of control for radio signal
Environmental Security threat	Spread a malicious code and spam (Open AP)
	Intrusion into intranet of enterprise (AP in enterprise)
	Stick with default security configuration of AP → Illegal access

Flooding (ARF) and Association Request Flooding (ASRF) are possible to exhaust AP's processing and memory resources. For preventing those DoS attacks, IEEE 802.11w [14] standard is established. However, PRF DoS attack is out of scope of IEEE 802.11w standard.

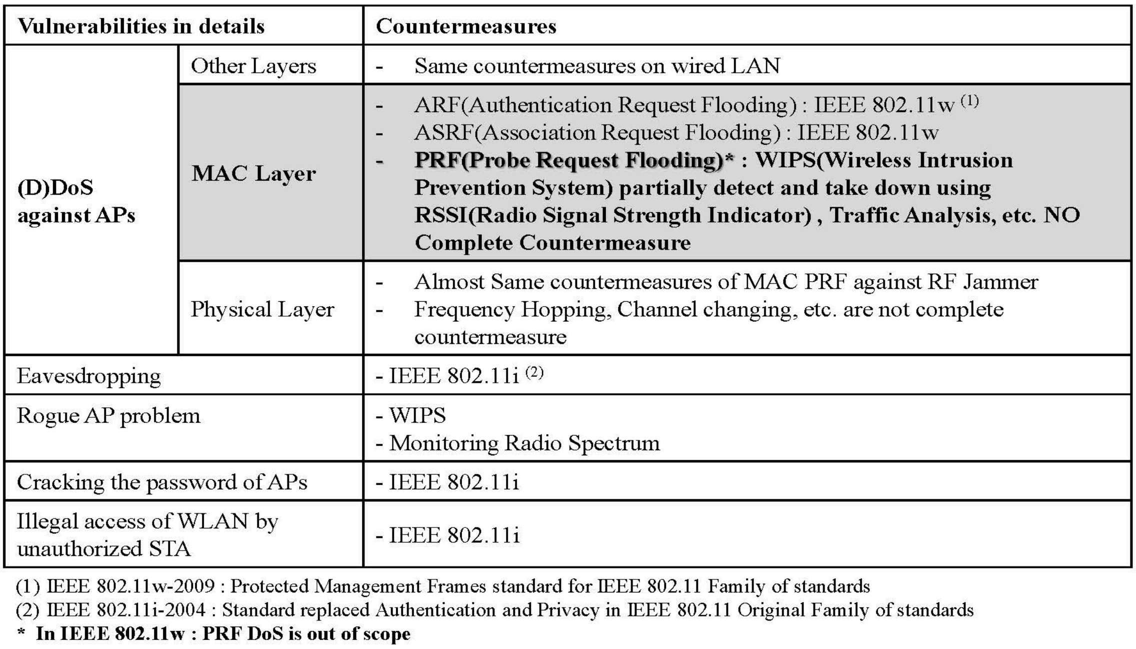
In Table 1.6 we summarize known countermeasures against technical threats on IEEE 802.11 WLAN. 'Eavesdropping', 'Cracking the password of APs' and 'Illegal access of WLAN by unauthorized user' can be protected by IEEE 802.11i [15] which defines more enhanced authentication and authorization method rather than weak WEP between STA and AP, namely WPA2. The RC4 stream cipher in WEP and WPA is replaced by the Advanced Encryption Standard (AES) block cipher in the standard.

Because malicious mobile codes are recently prevalent on the Internet, targeted smart mobile device not only can reveal the privacy information of an owner, but also can be a component of botnet construction by botmasters. While botnets based on wired computing device have been one of serious threats against the Internet, mobile botnets do not come into the spotlight due to the limitations of the smart mobile device such as battery power, network access constraints and computing resources. However, the technical improvement of the device and the wireless infrastructure leads the botmasters and malicious code developers into the main issue. Moreover, the popularity of open source based operating system such as Android [48] introduces the leverage of the most attractive platform for establishing botnets.

1.3 Motivation and Contributions

The vulnerabilities on wireless network and malicious mobile code will give a chance to make the worst-case scenario, which is to construct more powerful Distributed DoS (DDoS) attack rather than previous one. Most researches [2, 9, 10] related to the DoS attack in wireless network have been focusing on the deteriorative availability of single BS with the vulnerabilities of Medium Access Control

Table 1.6: Known countermeasures against technical threats on IEEE 802.11 WLAN



Vulnerabilities in details		**Countermeasures**
(D)DoS against APs	Other Layers	- Same countermeasures on wired LAN
	MAC Layer	- ARF(Authentication Request Flooding) : IEEE 802.11w [(1)] - ASRF(Association Request Flooding) : IEEE 802.11w - **PRF(Probe Request Flooding)* : WIPS(Wireless Intrusion Prevention System) partially detect and take down using RSSI(Radio Signal Strength Indicator) , Traffic Analysis, etc. NO Complete Countermeasure**
	Physical Layer	- Almost Same countermeasures of MAC PRF against RF Jammer - Frequency Hopping, Channel changing, etc. are not complete countermeasure
Eavesdropping		- IEEE 802.11i [(2)]
Rogue AP problem		- WIPS - Monitoring Radio Spectrum
Cracking the password of APs		- IEEE 802.11i
Illegal access of WLAN by unauthorized STA		- IEEE 802.11i

(1) IEEE 802.11w-2009 : Protected Management Frames standard for IEEE 802.11 Family of standards
(2) IEEE 802.11i-2004 : Standard replaced Authentication and Privacy in IEEE 802.11 Original Family of standards
* **In IEEE 802.11w : PRF DoS is out of scope**

(MAC) and physical (PHY) layers. It has been requiring extra jamming device or huge volume of traffic generating devices. Also those attacks happen at off-line and fixed position, moreover, there is no considerations of the trend of distributed manner and the mobility of user. The location of the attacker can be detected and excluded him from the network by wireless security solution such as Wireless Intrusion Protection System (WIPS) based on the Radio Signal Strength Indicator (RSSI) and traffic analysis. Such limitations of previous research can bring new risk of DoS attack, but it does not consider the risk as significant threats.

Therefore, we present a novel Distributed Denial of Service (DDoS) attack model exploiting smart mobile device's location information simultaneously with utilizing active scanning vulnerability in wireless network. Also we provide the simulation result of this attack model in order to evaluate a proof-of-concept of our attack model and show that relatively small number of attacker with tiny volume of active scanning traffic is sufficient to totally interrupt wireless communication. Also we propose a location information protection method based on chameleon hash scheme for a mitigation method against our DDoS attack model.

For constructing our DDoS attack model, an attacker who desires to aggress on the wireless network propagates a malicious code in users' smart mobile device. It becomes a zombie device of botnets which is stealthy controlled by C&C servers. After the botnet master issues the attack command, the zombie devices activate their smart active discovery attack, in case of WiFi probe request which is a basic attack unit to evaluate our model. The decisive distinction between existing active discovery attack such as Probe Request Flooding (PRF) and our attack is that there is no clear classification to decide whether normal active discovery protocol or not, due to the tiny volume traffic and reasonable interval of the request message. Furthermore, our attack can be forcefully performed in targeted specific region, where is a primary target of the attacker such as government organization, bank, *etc.*, based on the location information of user without extra device like a jammer.

In order to mitigate our attack model, we need location information protection method for user to prevent the attacker from clear target of the region specific setting. Previous researches on location privacy protection method can be largely categorized into location obfuscation and k-anonymity with additional trusted third party (TTP). However, previous work have no rigorous proof of security, then the information can be relatively easy to reveal. Because an additional TTP must be involved in the architecture of location privacy protection methods in most researches, a user's location information should be revealed to the TTP. Thus, we propose secure and effective location information protection method based on chameleon hash which never reveals accurate location of user to anybody, and also we present security and performance analysis of our proposed method.

1.4 Organization of Dissertation

The rest of this dissertation is organized as follows: Chapter 2 discuss DDoS attacks, threats, and mobile malicious codes for the background knowledge. We present DDoS attack model exploiting location information of user with its evaluation in Chapter 3. In Chapter 4, we propose a mitigation method against our DDoS attack model with performance and security analysis. And finally conclusion and open problems of this dissertation will be made in Chapter 5.

Chapter 2. Background

In this chapter, we present DDoS attacks in other layers and security threats on smart mobile devices for the background knowledge of this dissertation.

2.1 DDoS Attacks in Other Layers

For clear understanding DDoS attack in MAC layer over wireless networks, we firstly require to investigate well-known DDoS attacks in other layers such as network, transport, and application layer.

2.1.1 DDoS Attack in Network Layer

The DDoS attacks in network layer can be categorized into Internet Control Message Protocol/Internet Group Management Protocol (ICMP/IGMP) flooding and IP flooding. First, we summarize ICMP/IGMP flooding as follows:

(1) ICMP Smurf

An adversary forges the source IP address of ICMP packet into the IP address of targeted victim server, and then he broadcasts the ICMP PING packet over the network. All servers who receive the ICMP packet send a response message into the targeted victim server. So the network traffic of the target server is exponentially increased, then the server cannot provide a service. ICMP Smurf also is called a broadcast flooding.

(2) ICMP Ping of Death

An adversary sends massive fragmented packets and abnormal out of bands (OOB) to a victim server. The victim server exhausts system resources, and the server can be ultimately crashed. Due to the depletion of the resources, the degradation of the network performance will be decreased.

(3) ICMP Unreachable Storm

An adversary continuously sends the port-unreachable frame of ICMP packet into the target server. The server will be degrading the performance or halting the system.

(4) Ping of Death

Normal size of a ping packet is 56 bytes long. However, an adversary, who want to execute the ping of death, generates larger size of IP packet rather than its maximum size 65,535 bytes. He then sends the generated packets to the victim server and leads to interrupt the server.

(5) Ping Flooding

Due to sending continuous ICMP packet to targeted system, an internal service queue counter of the targeted system is exhausted, then the server is denial of service. Also the network of the server can be overloaded by the attack.

(6) Ping Sweep

An adversary manipulates the IP address of ICMP packet into broadcast IP. Then he get the list of available servers over the networks by analyzing received packets. Furthermore, the adversary can induce Over Load of whole networks.

Second, we describe the IP flooding summarized as follows:

(1) Teardrop Attack

The teardrop attack is called another name as IP Fragment Packet Flooding. Normally, the sender fragments IP datagram and the receiver reassembles the datagram in the TCP/IP communications, whereas an adversary, who want to perform teardrop attack, generates the datagram arbitrarily. The adversary sends the forged datagram to the targeted server which will be halted. Also this attack can exploit a bug in the reassembly code using invalid fragmented IP. This vulnerability can be occurred a damage on the target OS, then the server can be down. The adversary also can send deceptive information of IP fragmentation without real data, it is called open teardrop attack. The open teardrop is more powerful in point of the system overloading.

(2) Multicast Flooding

Multicast flooding utilizes generating IP broadcast packet in order to get targeted MAC address of a switch device.

2.1.2 DDoS Attack in Transport Layer

The DDoS attacks in transport layer can be largely divided into TCP and UDP flooding, those are described as follows:

- TCP Traffic Flooding

  (1) TCP Port Flooding

  An adversary sends large volume of packets to a specific TCP port, the specific port of the targeted server is denial of service. Moreover, this flooding has an effect on all deployed routers and switches.

  (2) TCP SYN Flooding

  When a client attempts to connect with a sever in TCP communication, TCP header with SYN flag is sent by the client. And then the client sends ACK after receiving SYN-ACK from the server. If an adversary just sends a large number of TCP SYN packets without reply the final ACK, the targeted server is depleting its resources for waiting the ACK.

(3) TCP SYN-ACK Flooding

Manipulated TCP SYN packet which holds the address of the victim server is sent to the clients. All clients who received the packet sends TCP ACK to the victim server. So, the resources of the server is depleted.

(4) TCP URG Flooding

An adversary sets the flag value of TCP header to URG(0x20) and sends a large number of TCP packets with URG header flag. The victim server falls into denial of service.

(5) TCP ACK Flooding

An adversary sets the flag value of TCP header to ACK(0x10) and sends a large number of TCP packets with ACK header flag. The victim server depletes almost resources and falls into denial of service.

(6) TCP PUSH Flooding

An adversary sets the flag value of TCP header to PUSH(0x08) and sends a large number of TCP PUSH header packets. The victim server falls into denial of service.

(7) TCP RESET Flooding

An adversary sets the flag value of TCP header to RESET(0x04) and sends a large number of TCP packets with RESET header flag. The victim server cannot provide normal services.

(8) TCP FIN Flooding

An adversary sets the flag value of TCP header to FIN(0x01) and sends a large number of TCP FIN header packets. The victim server is denial of service.

(9) TCP NULL Flooding

An adversary sets the flag value of TCP header to NULL(0x00) and sends a large number of TCP NULL header packets. The victim server falls into denial of service.

(10) TCP XMAS Flooding

An adversary sets the flag value of TCP header to URG(0x20), PUSH(0x08), RST(0x04), FIN(0x01), *etc.*, and then sends a large number of TCP packets with various header flags. The victim server is denial of service.

(11) Land Attack

An adversary arbitrarily synchronize his IP address and Port with the server's IP address and Port. If this condition is satisfied, the server performance can be decreased and interrupted.

- UDP Traffic Flooding

(1) UDP Port Flooding

An adversary sends large volume of packets to a specific UDP port, the specific port of the targeted server is denial of service. Moreover, this flooding has an effect on all deployed routers and switches.

(2) UDP Loopback

An adversary identically manipulates the port of source and destination to Echo (port number: 7), Quote of the day (port number: 17) and Chargen (port number: 19), and then the adversary sends a packet which can be occurred endless communications. If the UDP Loopback is successfully executed, the resources of the server and the network will be overloaded.

(3) DNS Flooding

An adversary generates and sends huge volume of DNS request packets to a DNS server. The DNS server cannot response the request packets.

(4) DNS Query Flooding

An adversary generates and sends huge volume of modified DNS query packets to a DNS server. Then, the DNS server can be down.

(5) DNS Reply Flooding

The reply address of DNS query is changed into the victim server by an adversary. The server can be a denial of service.

2.1.3 DDoS Attack in Application Layer

The most of recent DDoS attacks are targeted to an application service such as Web service. We describe the representative DDoS attacks in application layer as follows:

(1) HTTP GET Flooding

HTTP GET flooding attack use huge volume of HTTP GET requests by sending the requests to the targeted server. The server depletes their resource, then an adversary successes to halt the server. This attack can be performed whether valid or invalid HTTP GET requests.

(2) HTTP GET Flooding with Cache-Control

The cache server can be installed for the efficiency of a Web server. When the clients request same URL or service, the clients retrieve the Web data from the cache server. So an adversary, who want to skip the cache server during HTTP GET flooding attack, attaches Cache-Control in the HTTP GET requests in order to avoid the cache server. The the flooding attack can affect directly into the targeted Web server.

(3) HTTP Session depletion

An adversary opens many sessions of HTTP and then he keeps the sessions as long as possible. Then, the resources of server is depleted.

(4) Fragmented HTTP Header Attack

An adversary generates a valid fragmented HTTP header which can be sufficient to make a connection with the victim server. The fragmented HTTP header is more efficient attack rather than full HTTP protocol in point of adversary view.

(5) Telnet Flooding

An adversary generates and sends repeatedly some control characters that are valid in TCP telnet session. The server is denial of service.

(6) FTP PASV DoS

After the connection of the FTP service is established by an adversary side, the adversary sends a large number of FTP PASV commands before responding the server. The FTP server then can be downed.

2.2 Threats and Malicious Codes on Smart Mobile Devices

The popularity of the smart mobile devices such as smartphones, tablet PCs, *etc.*, have been increasing sharply. Recently, storing the private information of the user into the smart mobile devices is a general tendency. Because the devices can keep connection with the Internet for 24 hours with its mobility, new security challenges on the devices have been raised.

In this section, we describe security threats of the smart mobile devices with the trends of mobile malicious code.

2.2.1 Security Threats

We can summarize the security threats on smart mobile devices as follows:

(1) Open Architecture

The main difference between smart mobile devices and legacy mobile devices is whether adopts open architecture or not. Typical open architecture of smart mobile devices is Android OS platform [48] developed by Google. The Android support open Application Programming Interface (API), which can access internal resources of smart mobile devices, for developing APPs. Also various interfaces for the communications also have open architecture. The open accessibility gives us a convenient environment for the developing APPs and the communications. However, the open architecture is also main reason of security threats on smart mobile devices. An adversary can develop his malicious mobile code easily by using the open API. Also the malicious code or malware can be propagated rapidly through various communication interfaces.

(2) Open APP market

Anyone can develop their APPs using open API, and uploads them for sales in the open APP market. Also an attacker can easily develop and insert his malicious code in the APP. Not only the open APP market, but also the third party market can be a good distribution channel of the malicious code.

(3) Wireless network environment

The smart mobile devices can utilize various wireless networks such as 3G, LTE, BlueTooth, WiFi, Near Field Communication (NFC), and so on. Therefore, an adversary who wants to establish botnets can use the various communication channels for the distribution and command&control (C&C) channel of malicious code. Also the various communication channels are opened to access that can increase the possibility for the intrusion of the adversary or hacker.

(4) Physical threats

Due to the properties of the smart mobile devices such as mobility, daily use, *etc.*, that can have more drawbacks of lost, stolen, and breakage rather than legacy information devices. Even if the physical threats are not technical threats, we should focus on the threats which can revel important user's privacy data stored in the devices.

2.2.2 Malicious codes

We use the malicious code also called malware which is a compound noun of malicious and software in this dissertation. The mobile malicious code is a mobile software which intends to malicious act such as destroys the system, leaks the information, and so on. The distribution and infection channels of mobile malicious codes are diverse. Fig. 2.1 depicts the percentage ratio of the channels.

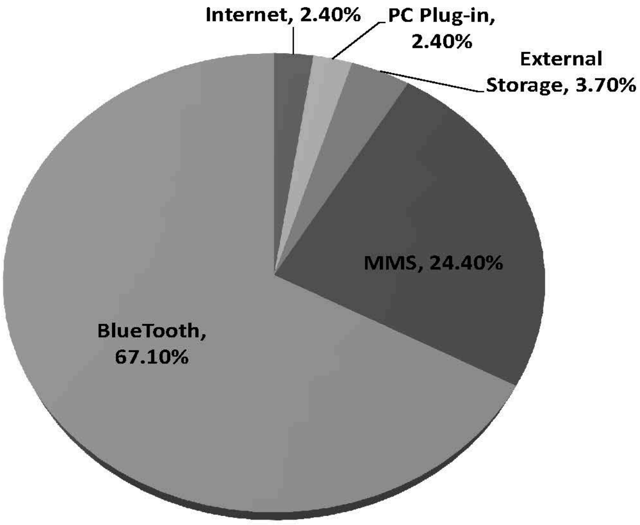


Figure 2.1: The ratio of the infection channels of mobile malicious code [Refer to KISA Report]

Current increasing trend of mobile malicious code including botnet depicts in Fig. 2.2.

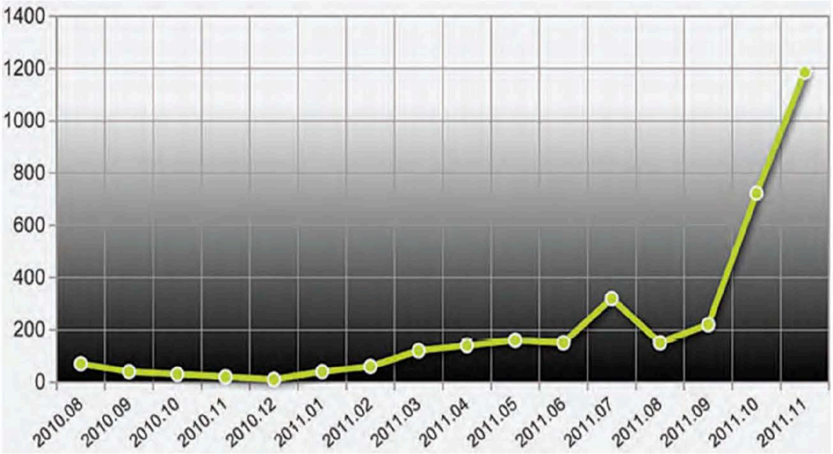


Figure 2.2: Trend of mobile malicious code
[Refer to Ahn LAB Report 2011.11]

The trend of the occurrence ratio of mobile malicious codes is going to move Android platform as shown in Figs. 2.3 and 2.4.

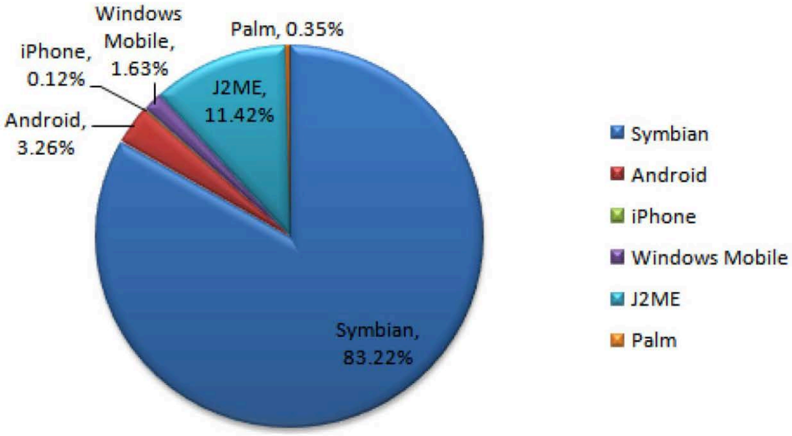


Figure 2.3: Malicious Codes Occurrence Ratio in 2010.12 [Refer to KISA Report, 2011]

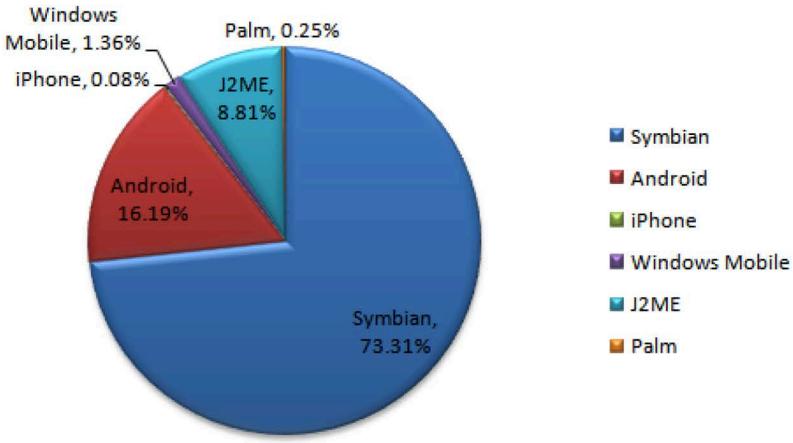


Figure 2.4: Malicious Codes Occurrence Ratio in 2011.02 [Refer to KISA Report, 2011]

Chapter 3. DDoS Attack Exploiting Location Information of User

3.1 Related Work

There has been a large number of works related to DoS and DDoS attacks, detection methods, mitigation approaches and countermeasures in the general Internet services [4, 5, 6, 11, 12, 21]. Also most of the wireless security studies primarily focus on the weakness of 802.11 and its security extension [1, 3, 8]. Despite the large number of work related to DoS and DDoS issues, relatively a few DoS attacks in wireless network have been researched.

Table 3.1: Summary of previous DoS attacks in WLAN

	BS03 [2]	**BFV08** [9]	**BT08** [10]	**XTZW05** [13]
Attack Method	ARF / Virtual CS	PRF/ARF/ASRF	Survey	Jamming
Attack Type	DoS	DoS	DoS	DoS (Jamming)
Experiment	1 Attacker 1 AP 4 STAs	1 Attacker 1 AP 2 STAs	-	1 Attacker Ad hoc 2 STAs
Traffic Volume	Huge	Huge	Huge	Huge
Additional Information	No	No	No	No

As summarizing in Table 3.1, Bellardo and Savage [2] investigated the DoS vulnerabilities on 802.11 MAC layer, and proposed deauthentication and virtual carrier-sense attacks. Also they implemented the attacks in commodity 802.11 devices with experimental analysis. Bernashi *et al.* pointed out APs vulnerabilities and examined the DoS attack with various commercial AP products. Then, they claimed that a single malicious station with huge volume of traffic could hinder legitimate communications [9]. In [10], the authors surveyed, classified and compared DoS vulnerabilities in 802.11 networks with available countermeasures. Wu *et al.* [13] suggest four jammer attack models that are constant, deceptive, random and reactive jammer. They also propose detection method utilizing RSSI and pre-installed location information of jammer. However, previous work target into single AP with one powerful attacker, which generates huge volume of traffic. Because previous attacks need to install the attacker device in fixed location, those technically are not characterized as DDoS attack. Fig. 3.1 depicts the typical DoS attack model in previous researches.

Recently, researches on mobile botnets and malwares have been attracting a big concern. In [19],

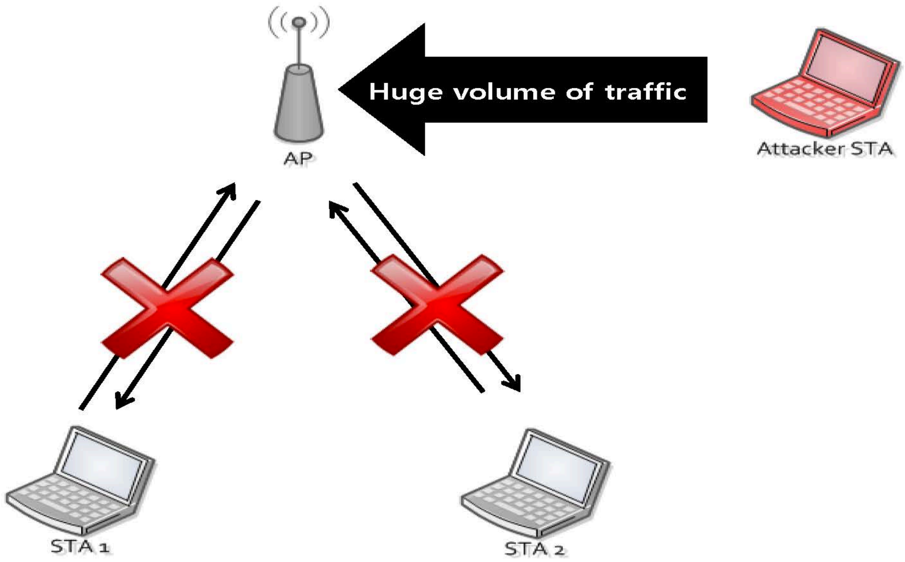


Figure 3.1: Typical DoS attack model

the authors surveyed the current status of mobile malware and suggested future directions in various smartphone operating systems. For constructing botnet, Hua and Sakurai [20] presented a botnet based on Short Message Service (SMS) as Command and Control (C&C) channel with the simulation of botnet propagation. Other C&C strategy utilizing URL was introduced by [18] and the authors implemented the prototype over Android platform. Knysz *et al.* [22] designed mobile WiFi botnets and evaluated it by utilizing only open WiFi networks. They claimed that their botnet model could successfully mount a DDoS attack against unprotected server. But, the goal of previous research on botnets is to disturb targeted server system in the Internet. That means they just changed the zombie PC into the mobile device, and the wired C&C channel into the wireless channel.

3.2 Overview

For constructing our DDoS attack model, an attacker who desires to aggress on the wireless network propagates a malicious code in users' smart mobile device. It becomes a zombie device of botnets which is stealthy controlled by C&C servers. After the botnet master issues the attack command, the zombie devices activate their smart active discovery attack, in case of WiFi probe request which is a basic attack unit to evaluate our model. The decisive distinction between existing active discovery attack such as PRF and our attack is that there is no clear classification to decide whether normal active discovery protocol or not, due to the tiny volume traffic and reasonable interval of the request message. Furthermore, our attack can be forcefully performed in targeted specific region, where is a primary target of the attacker such as government organization, bank, *etc.*, based on the location information of user without extra device like a jammer.

3.3 Attack Model

3.3.1 Notations

We will use the notations as shown in Table 3.2 in order to describe our DDoS attack model.

Table 3.2: Notations

Symbol	Description
X_i	A device X has its identity i
STA_x	Normal user device
STA'_y	Zombie device
BS_z	Base Station
CS_t	C&C Server
BM	Botnet Master
cmd	Command and control message
m	Normal message
m'	m including mobile malicious code
$A(cmd) \Rightarrow B$	A sends cmd to B via C&C channel
$A(m) \rightarrow B$	A sends m to B
$A \rightsquigarrow B$	Active scanning attack from A to B
$m\|\|m^*$	Concatenation m and m^*

3.3.2 Botnet Architecture

The overall architecture of our botnet is illustrated in Fig. 3.2. The architecture consists of BM, CS_t, STA_x and STA'_y. The BM commands and controls all the distributed CS_t. A CS_t manages their bots directly or indirectly via various C&C channels. The C&C channel can be smartly changing according to the usage pattern by users. If the user frequently uses a social network service such as Twitter, Facebook and others, those will be a main C&C channel. If the user sticks into a messaging service (*i.e.* SMS and network messenger), CS_t adaptively switch the channel into the service protocol. Also the authenticated and encrypted cmd between STA'_y and CS_t causes to decrease the detection probability of intrusion detection system. In order to construct a practical botnet model, we utilize Android platform which is not only the most popular operating system for smart mobile device but also the easiest way to construct bot, due to their open source policy.

3.3.3 Construction and Distribution Protocols

The botnet construction of our attack model starts with generating malicious mobile code. The general process of generating malicious mobile code is shown in Fig. 3.3. This technique already applied GingerMaster [45] and DroidDream [46]. The owner of APP in Android is 'system account (uid=1000)',

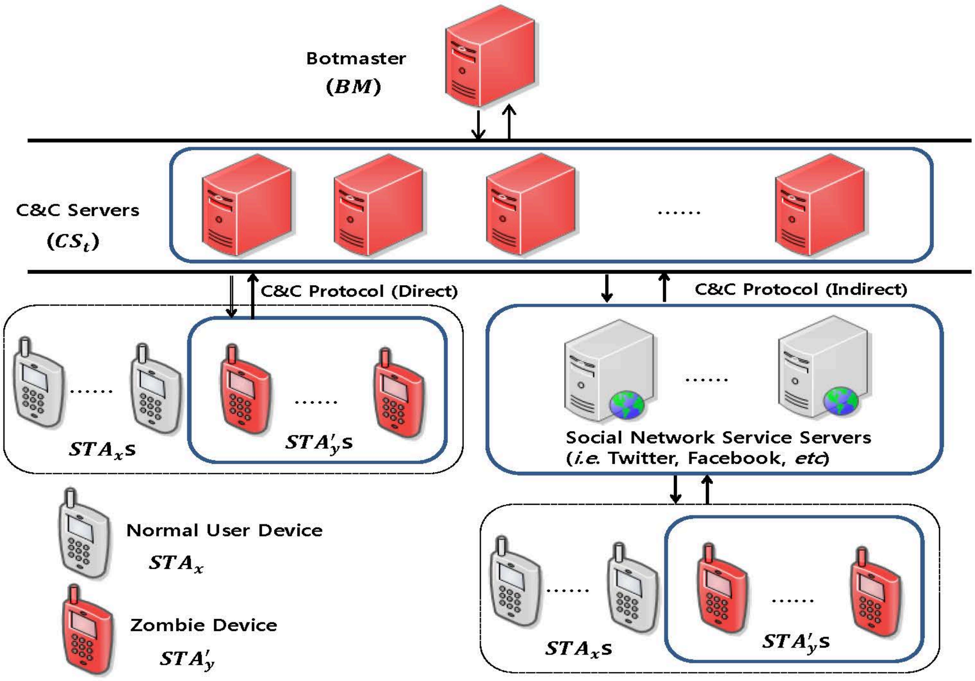


Figure 3.2: The overall architecture of our botnets

however the problem is to open readable permission to everyone. So the attacker can easily get the original APP file, then he decompiles it easily caused by the property of java class file [17]. After the optimization to decrease the complexity of decompiled file, the attacker can insert his own mobile malicious code. Then, the processes of repacking and redistribution are following.

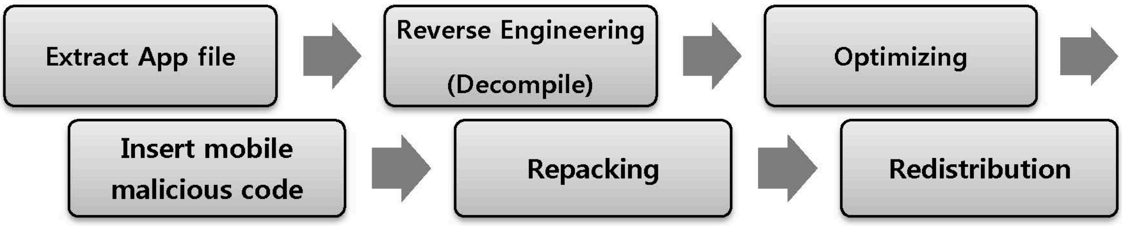


Figure 3.3: The process of malicious mobile code generation

The distribution protocol can be divided into two parts: one is a bootstrapping protocol, and the other is an infection protocol for spreading the malicious mobile code by previous infected zombie devices.

As shown in **Protocol 1**, bootstrapping protocol is a starting point to modify normal user's smart mobile device, STA_x into zombie device, STA'_y. Also this protocol can be a part of our infection protocol.

Protocol 1 Bootstrapping protocol

```
1: $STA_\alpha(APP\ request) \rightarrow$
   $S = \{S|$Server: Open App market, SNS, Web, blog$\}$
   ▷ $BM$ already published $m'$ into $S$
2: $S(m') \rightarrow STA_\alpha$
3: $STA_\alpha$ to be a $STA'_\alpha$
```

In **Protocol 2**, we describe the procedure of infection mechanism of our bot. First, BM sends an infection command to all CS_t for expanding the size of his botnets. Second, CS_t transmits the command to the group of STA'_y. STA'_y, then send the message, which includes the link of malicious APP, to STA_x. STA_x can be chosen by the contact list of STA'_y. Also the communication channel will be selected by the source of the list, in case of a social network service(SNS) channel can be selected from the friend list of STA'_y. After that, STA_x performs **Protocol 1**. Finally, STA_x will be another STA'_y.

Protocol 2 Infection protocol

```
1: $BM(cmd) \Rightarrow$ all $CS_t$
   ▷ $cmd = \{$Infection Command$\}$
2: $CS_t(cmd) \Rightarrow STA'_y$
3: $STA'_y(m) \rightarrow STA_x$ via SMS, SNS, or Short URL
   ▷ $STA_x$ based on the contact list of $STA'_y$
   ▷ $m =$ recommendation of $(m')$
4: $STA_x$ jump to **Protocol 1**
```

After execution of construction and distribution protocols, STA'_y has malicious mobile code in his smart devices as shown in Fig. 3.4. Normally the malicious APP is located in the application area as third party APP. More advanced stealthy malicious mobile code can be placed into the malware runtime and rootkit. In case of malware runtime, the user information collector can gather the information of location, IMSI, and UUID. Then the core C&C libraries can synchronized with original OS runtime. The synchronizer can guarantee the stealthiness of the mobile malicious code, because it sends the user's information only OS runtime execution. Also the rootkit with native call jump table makes the mobile malicious codes on stakeout until the kernel update.

3.3.4 Attack protocol

We present attack protocol for DDoS attack exploiting STA'_y's location as shown in **Protocol 3**.

The BM firstly issue 'Discover Location Command' for aggregating the location of STA'_y. If the BM already has the location information of STA'_y by updating the information periodically, it can be skipped from steps 1 to 4 in **Protocol 3**. After finding the location of STA'_y out, BM selects group of STA'_y within target region. And then BM gives the attack command with the interval of active discovery attack and duration time for the attack. The interval can be assigned by BM. If BM assigns the interval between MinChannelTime and MaxChannelTime for full scanning latency of radio channel, then the attack can be hard to detect. In case of IEEE 802.11, BM can choose the value between MinChannelTime = 10ms and MaxChannelTime = 50ms [16]. Also the interval can be increased more than MaxChannelTime by depending on the size of botnet in the target region. After receiving the attack command, STA'_y executes active discovery attack (Probe Request Flooding in case of WiFi) during the time of duration. Finally, STA'_y send 'Report end of attack' to BM.

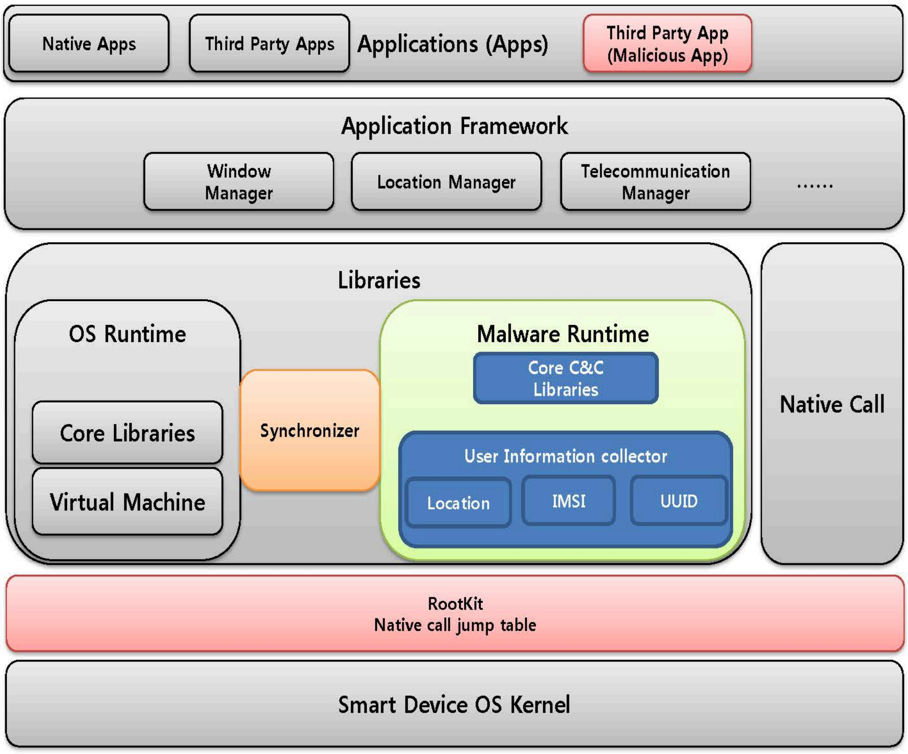


Figure 3.4: An example structure of malicious mobile code in smart mobile device

Protocol 3 Attack protocol

```
1:  $BM(cmd) \Rightarrow$ all $CS_t$
      $\triangleright$ $cmd$ = {Discover Location Command}
2:  $CS_t(cmd) \Rightarrow STA'_y$
3:  $STA'_y(cmd) \Rightarrow CS_t$
      $\triangleright$ $cmd$ = Current Location
4:  $CS_t(cmd) \Rightarrow BM$
      $\triangleright$ $cmd$ =
      {Report Command||Location Information of $STA'_y$}
5:  $BM$ selects group of $STA'_y$ within target region
6:  $BM \Rightarrow CS_t(cmd)$
      $\triangleright$ $cmd$ = {(Attack Command||Interval||Duration)}
7:  $CS_t(cmd) \Rightarrow STA'_y$ based on Step 6.
      $\triangleright$ $cmd$ = {Execute Attack||Interval||Duration}
8:  **while** Attack time $\leqq$ Duration **do**
9:      $STA'_y \rightsquigarrow BS_z$ : Corresponding Interval
10: **end while**
11: $STA'_y(cmd) \Rightarrow CS_t$
      $\triangleright$ $cmd$ ={Report end of attack}
12: $CS_t(cmd) \Rightarrow BM$
      $\triangleright$ $cmd$ ={Report end of attack}
```

3.4 Evaluation

In order to evaluate the proof-of-concept of our DDoS model, we assume that BM has own botnet which has a reasonable size to perform the active scanning attack. Also BM already have the location information of STA'_y. And the target region is assumed by BM's primary target. BS_z is densely deployed APs, denoted AP_z based on IEEE 802.11 standards.

3.4.1 Simulation Model

For evaluation of our DDoS attack, we simulate it in two models: 1) Simulation Model I under ideal grid topology and 2) Simulation Model II under considering real topology of 'WELCOME_KAIST' WLAN in N5 building at KAIST campus, Republic of Korea. We summarize our simulation environment as follows:

- Discrete network simulator:
  - Simulation model I: EXata simulator [49].
  - Simulation model II: NS3 simulator [50].
- Evaluation: Traffic information in terms of throughput, packet loss ratio, end to end delay, number of dropped packets in MAC layer.
- Services: The most popular services in smart mobile devices.
  - Simulation model I: Video streaming and mVoIP.
  - Simulation model II: Video streaming.
- Network Topology:
  - Simulation model I: Grid deployment.
  - Simulation model II: Real deployment.

[Simulation Model I]

We evaluate the traffic information of the our DDoS attack using a EXata simulator [49] in terms of throughput, packet loss rate, end to end delay, packet drops in MAC. We also consider the most popular services like video streaming and mVoIP in APPs of the smartphone. The simulated grid topology consists of 16 APs as illustrated in Fig. 3.5. Fig. 3.6 shows the channel allocation of the simulation topology. We consider separate channel allocation to minimize interference and to focus on effectiveness of our DDoS attack. We also consider a typical and well-planned dense wireless network topology. If our DDoS attack model can be valid in our topology, our attack model in a complicated and unplanned dense wireless network is more highly effective. Because the unplanned dense wireless network has more interference effects and unexpected management frame traffic, our DDoS attack can be successfully achieved bandwidth depletion with the minimum effort.

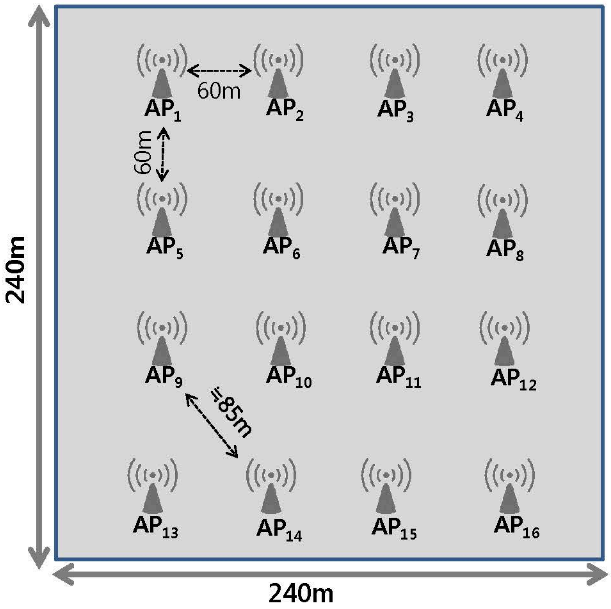
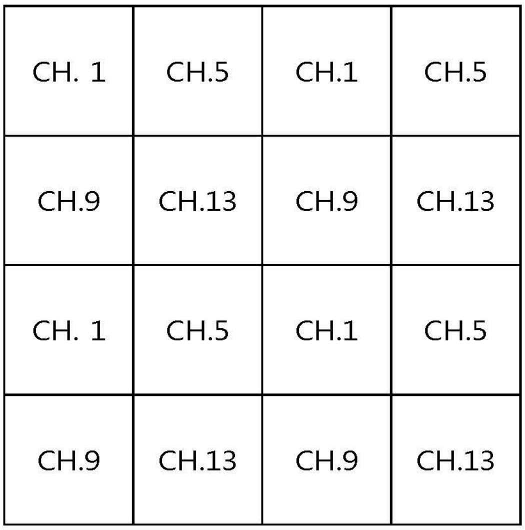


Figure 3.5: [Simulation Model I]: Simulation topology of the attack model



CH. 1	CH.5	CH.1	CH.5
CH.9	CH.13	CH.9	CH.13
CH. 1	CH.5	CH.1	CH.5
CH.9	CH.13	CH.9	CH.13

Figure 3.6: [Simulation Model I]: Channel allocation of the simulation topology

The simulation scenarios consist of 16 stationary STA_x and varying number of attacker STA'_y. Each attacker STA'_y can be controlled by BM who manages C&C servers, and the number of attacker STA'_y are increasing whenever it has been infected by malicious mobile code. A unique number of identity is assigned to each BS_z and the BS_z of our simulation is IEEE 802.11 AP. The distance between two adjacent APs is configured to 60 m while the radius of the radio range is 75 m. Four channels in 2.4 GHz are assigned to each AP. An attacker STA'_y moves around the simulated region randomly at the pedestrian speed of 1 m/sec to 2 m/sec. The radio type of AP, STA_x and STA'_y is IEEE 802.11g with

54 Mbps link rate. The simulation time is 600 sec and the pause time of each STA'_y is set to 30 sec.

The packet drop ratio of the probe request management frame in MAC layer is compared with constant PRF attack intervals of 10 ms and 50 ms. In case of the video streaming traffic, a pair of randomly chosen STA_x transmit and receive 400 Kbps constant bit rate (CBR) flows with the packet size of 1024 Bytes for each flow. In case of the mVoIP traffic, a pair of randomly chosen STA_x transmit and receive 160 bytes packet under assumption of using G.711 codec with 20 ms interval. Table 3.3 summarizes our simulation parameter for [Simulation Model I] in detail.

Table 3.3: Simulation Parameters for [Simulation Model I]

Parameters	Values
Radio type	802.11a/g
Propagation	TwoRayGround
Antenna	Omni-directional
Data rate	54 Mbps
RTS/CTS	Disabled
No. of channels	4 channels in 2.4 GHz bandwidth (1, 5, 9, 13)
Attack interval	10, 50 ms
Mobility	Random Way Point (Pause time: 30sec)
Applications	Video Streaming, mVoIP
# of Attackers	Varying
Simulation time	600sec

[Simulation Model II]

We evaluate the traffic information of the our DDoS attack using a NS3 simulator [50] in terms of throughput, packet loss rate, end to end delay, packet drops in MAC. We also consider the most popular services like video streaming in APPs of the smartphone. The case of mVoIP is excluded due to the limitation of NS3 simulator. The simulated real topology consists of 9 APs as illustrated in Fig. 3.7. Fig. 3.8 shows the channel allocation of the simulation topology. We assign real channel allocation of APs in 'WELCOME_KAIST' WLAN. This deployment is also well-planned dense wireless network by network administrators of KAIST. However, we can have chance to evaluate our attack model under one of real deployment of wireless network.

The simulation scenarios consist of 9 stationary STA_x and varying number of attacker STA'_y. Each attacker STA'_y can be controlled by BM who manages C&C servers, and the number of attacker STA'_y are increasing whenever it has been infected by malicious mobile code. A unique number of identity is assigned to each BS_z and the BS_z of our simulation is IEEE 802.11 AP. The distance between two adjacent APs is configured to 30 m while the radius of the radio range is 35 m. Three channels in 2.4

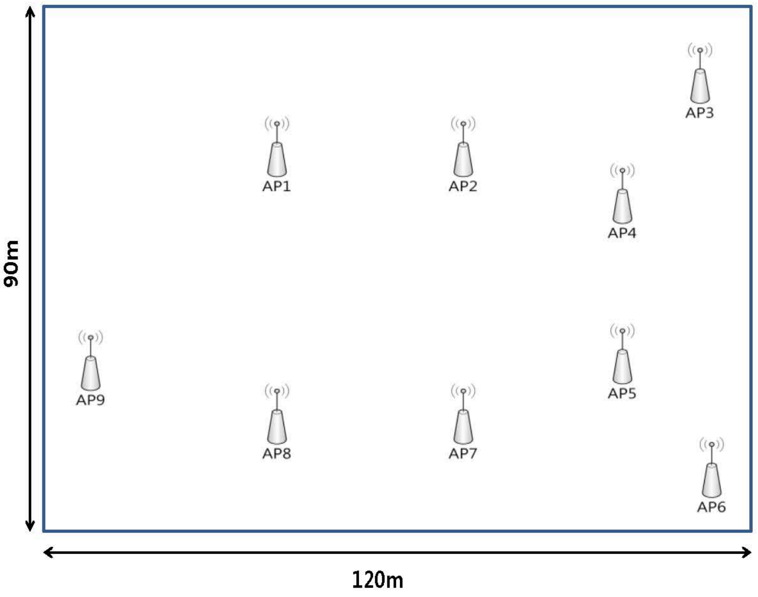


Figure 3.7: [Simulation Model II]: Simulation topology of the attack model

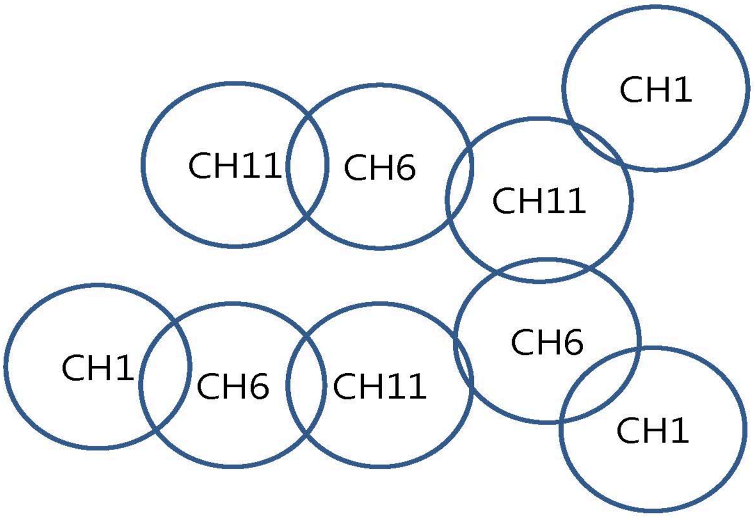


Figure 3.8: [Simulation Model II]: Channel allocation of the simulation topology

GHz are assigned to each AP. An attacker STA'_y moves around the simulated region randomly at the pedestrian speed of 1 m/sec to 2 m/sec. The radio type of AP, STA_x and STA'_y is IEEE 802.11g with 54 Mbps link rate. The simulation time is 600 sec and the pause time of each STA'_y is set to 30 sec.

The packet drop ratio of the probe request management frame in MAC layer is compared with constant PRF attack intervals of 10 ms and 50 ms. In case of the video streaming traffic, a pair of randomly chosen STA_x transmit and receive 400 Kbps constant bit rate(CBR) flows with the packet size of 1024 Bytes for each flow. The case of mVoIP traffic is excluded due to impossibilities of G.711 codec assumption in NS3 simulator. Table 3.4 summarizes our simulation parameter in detail.

Table 3.4: Simulation Parameters for [Simulation Model II]

Parameters	Values
Radio type	802.11a/g
Propagation	TwoRayGround
Antenna	N/A
Data rate	54 Mbps
RTS/CTS	Disabled
No. of channels	3 channels in 2.4 GHz bandwidth (1, 6, 11)
Attack interval	10, 50 ms
Mobility	Random Way Point (Pause time: 30sec)
Applications	Video Streaming
# of Attackers	Varying
Simulation time	600sec

3.4.2 Simulation Results and Analysis

We define the evaluation parameters [unit] as follows:

- **PRI** : Probe Request Attack Interval [ms]
  - 'No' means that there is no attack
- **PDR** : Packet Drop Ratio [%]= $\frac{L}{L+S} \times 100$
  - L = # of lost packets
  - S = # of packets received successfully
- **TP** : Throughput [bps]
- **E2ED** : End to end delay between normal communication of STA_x [ms]
- **#PD** : Number of packet drops in MAC layer

Our simulation results of two models show that relatively small number of attacker (20 zombie devices in case of [Simulation Model I], 11 zombie devices in case of [Simulation Model II]) with tiny volume of active scanning traffic is sufficient to totally interrupt wireless communication. Our results and analysis of each simulation model in detail are following.

[Simulation Model I]

Tables 3.5 and 3.6 show that the average of evaluated value from every APs in case of 10 and 20 zombie STA'_y, respectively. We could observe that almost half of packets loss are from attack interval

50 ms with 10 zombie STA'_y in case of video streaming (VS), then attack interval 10 ms is enough to disturb the service in the same number STA'_y. In case of mVoIP, we also decide that attack interval 10 ms is critical point to decline quality of voice. From Table 3.6, the attack interval does not have meaning any further in both cases of video streaming and mVoIP. Figs. 3.9 and 3.10 depict the outline of Tables 3.5 and 3.6, respectively.

Table 3.5: The simulation result of 10 STA'_y

	PRI	PDR	TP	E2ED	#PD
VS	No	3.7	394661	2.9	1101
	50ms	46.8	217651	1249.7	18682
	10ms	75.1	101998	2414.3	23364
mVoIP	No	1.9	62775	1.6	577
	50ms	10.7	57166	1223.0	18726
	10ms	68.4	20225	6383.7	24798

Table 3.6: The simulation result of 20 STA'_y

	PRI	PDR	TP	E2ED	#PD
VS	No	3.7	394661	2.9	1101
	50ms	88.0	45726	1128.9	26374
	10ms	88.9	48944	4672.9	27211
mVoIP	No	1.9	62775	1.6	577
	50ms	90.6	738	1163.5	27049
	10ms	98.8	5978	2844.3	27236

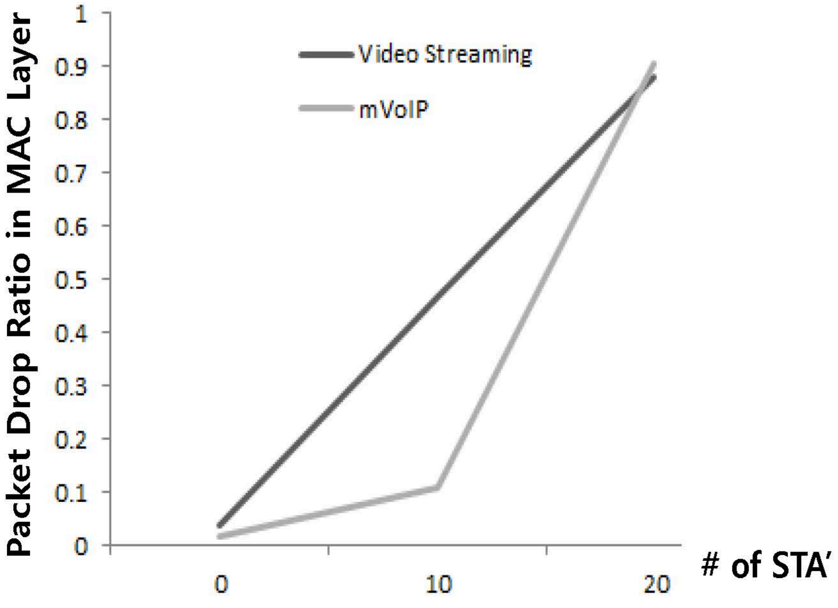


Figure 3.9: Average of PDR within 50 ms PRI

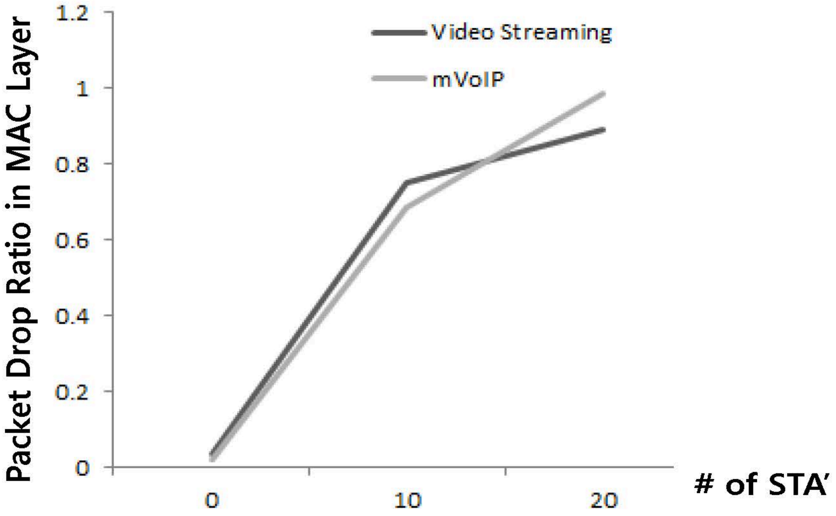


Figure 3.10: Average of PDR within 10 ms PRI

Figs. 3.11 and 3.12 illustrate the evaluation results of packet drop ratio in MAC layer at each identifier of AP based on video streaming application. Our DDoS attack increases the resource of each AP overhead by 1% - 10% to more than 50% in Fig. 3.11 and to almost 90% in Fig. 3.12, respectively.

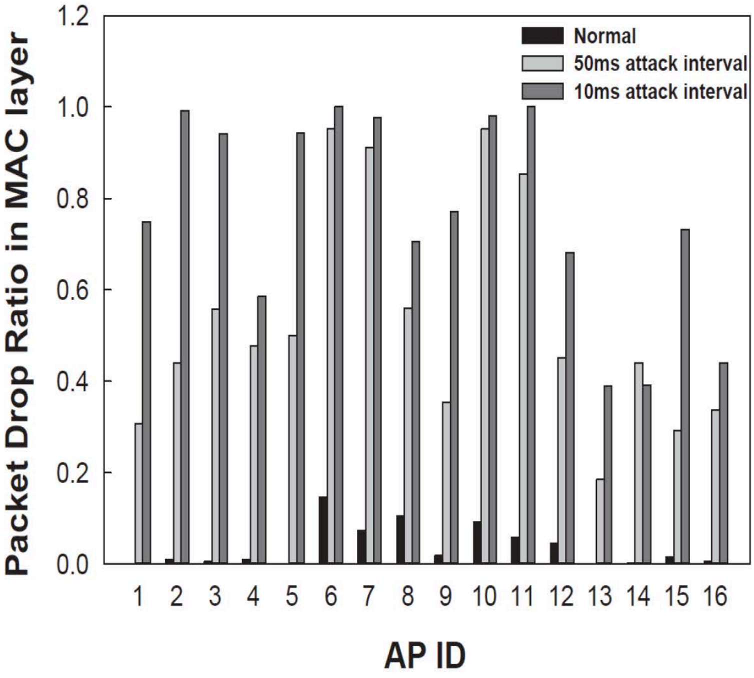


Figure 3.11: Packet drop ratio (VS/10 STA'_y/each AP).

Figs. 3.13 and 3.14 depict the evaluation result of packet drop ratio in MAC layer at each identifier of AP based on mVoIP application. Our DDoS attack increases the resource of each AP overhead by 1% - 10% to more than 50% in Fig. 3.13 and to almost 99% in Fig. 3.14, respectively.

From these figures, we could more easily decide that the threshold number of STA'_y to execute a

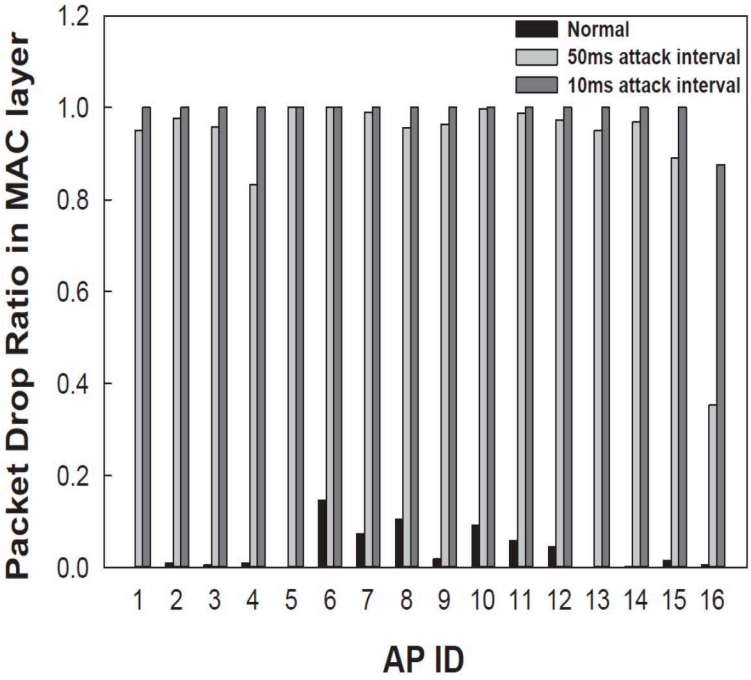


Figure 3.12: Packet drop ratio (VS/20 STA'_y/each AP).

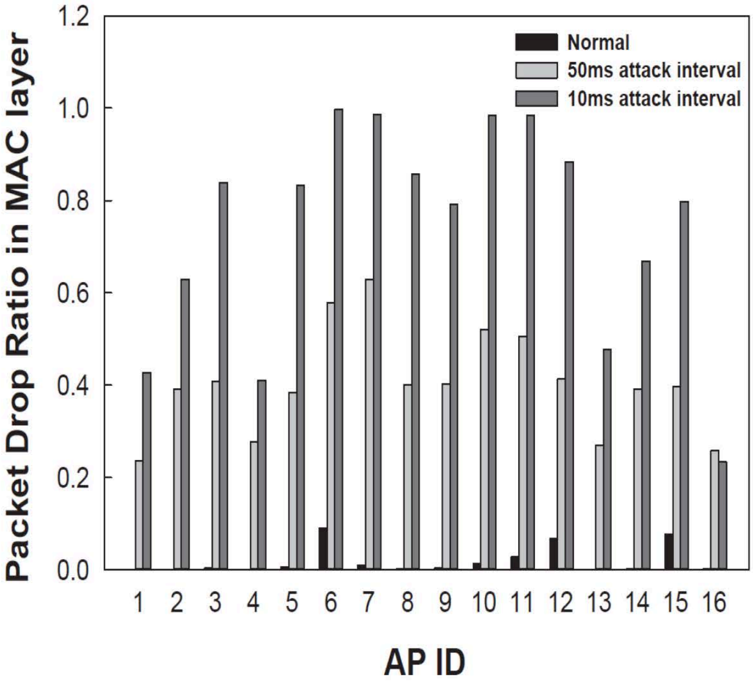


Figure 3.13: Packet drop ratio (mVoIP/10 STA'_y/each AP).

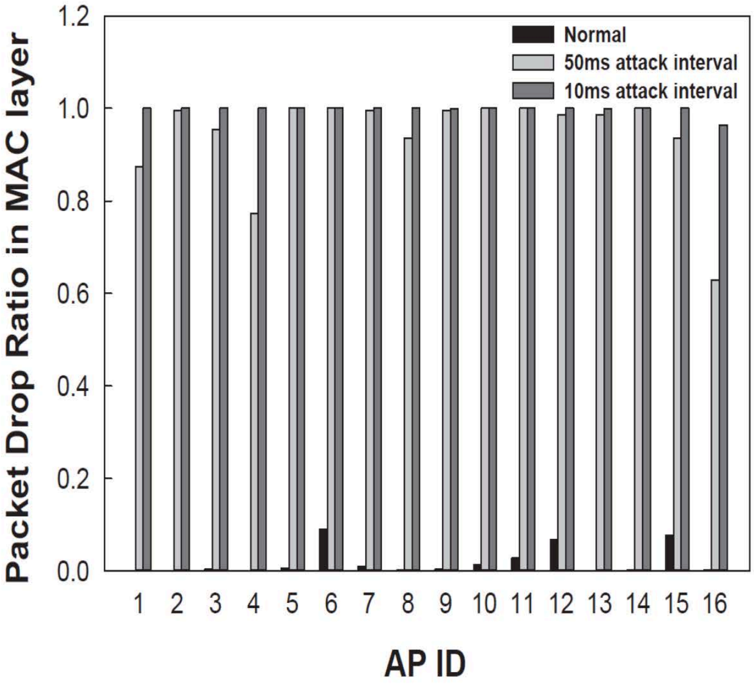


Figure 3.14: Packet drop ratio (mVoIP/20 STA'_y/each AP).

successful DDoS attack. If there exist more than 20 STA'_y in dense wireless networks, all the APs in the specific region are interrupted under our simulation setting. Additionally, the 50 ms and 10 ms interval of probe request flooding is relatively tiny volume of traffic to achieve the goal of DDoS attack.

[Simulation Model II]

Table 3.7 shows that the average of evaluated value from every APs in case of 11 zombie STA'_y. We could observe that almost 90% of packets loss are from attack interval 50 ms with 11 zombie STA'_y in case of video streaming (VS), then attack interval 10 ms is sufficient to totally interrupt the service in the same number STA'_y. Fig. 3.15 depict the average of PDR within 50 ms and 10ms PRI by increasing the number of STA'_y from 0 to 11.

Table 3.7: The simulation result of 11 STA'_y

	PRI	PDR	TP	E2ED	#PD
VS	No	5.1	386544	3.99	1577
	50ms	90.2	43218	1157.3	26984
	10ms	95.1	101998	4998.1	28761

Fig. 3.16 shows the evaluation result of packet drop ratio in MAC layer at each identifier of AP based on video streaming application. Our DDoS attack increases the resource of each AP overhead by 1% - 18% to more than 90% in 50 ms PRI and to almost 95% in 10 ms PRI, respectively.From these figures, we could more easily decide that the threshold number of STA'_y to execute a successful

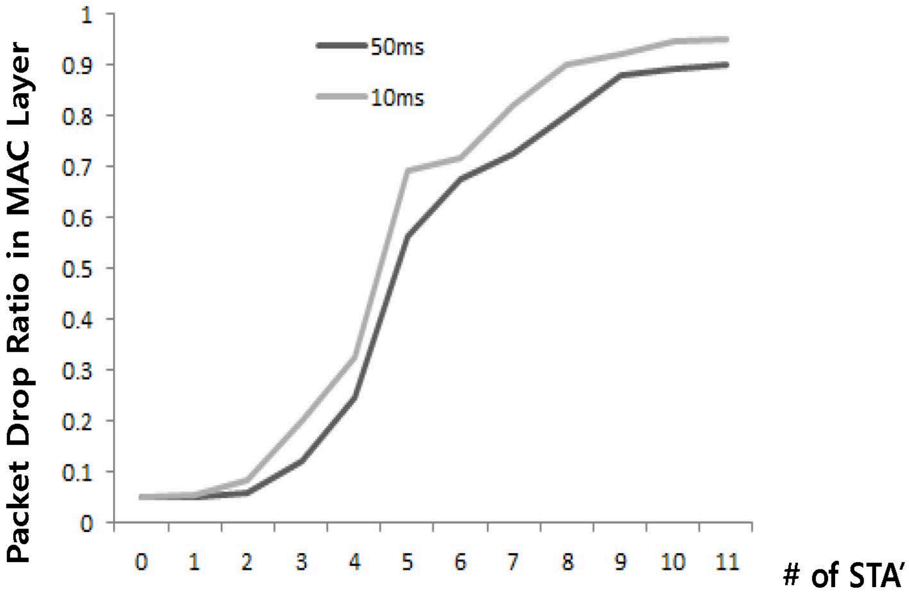


Figure 3.15: Average of PDR within 50 ms and 10ms PRI

DDoS attack. If there exist more than 11 STA'_y in dense wireless networks, all the APs in the specific region are interrupted under our simulation setting. Additionally, the 50 ms and 10 ms interval of probe request flooding is relatively tiny volume of traffic to achieve the goal of DDoS attack. We here note that both Probe Request and Probe Response frames are transmitted at a basic rate (*i.e.* 6 Mbps in 802.11g) because a STA does not know the supported data rate of an AP before association. Therefore, the distributed PRF attack is more severe as the data rate is higher.

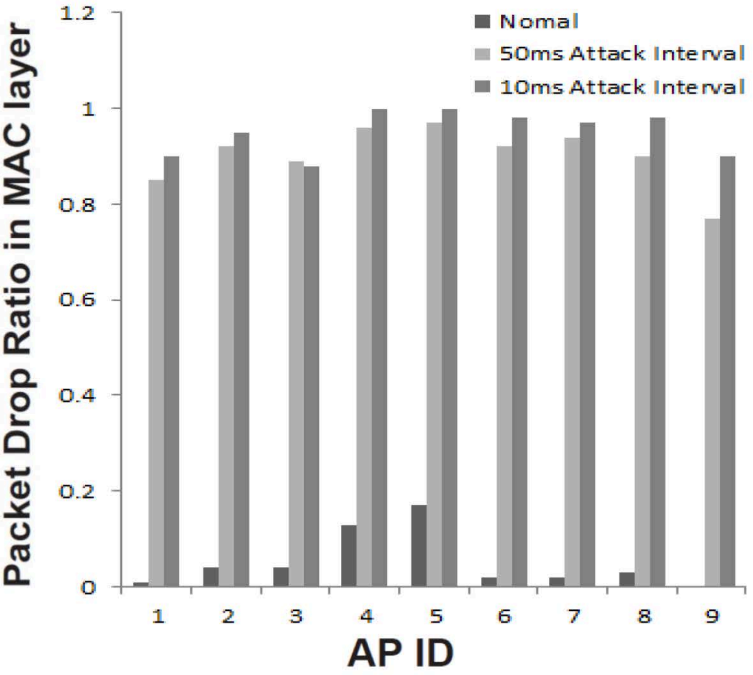


Figure 3.16: Packet drop ratio (VS/11 STA'_y/each AP).

3.5 Summary

In this chapter, we have investigated the first DDoS attack model exploiting location information of smart mobile device and utilizing active discovery vulnerability with unrecognizable traffic in wireless network. Without any costly devices, we have evaluated the proof-of-concept of the regional targeted DDoS attack model using simulation and its analytical result. Our simulation results show that 20 and 11 attacker STAs with tiny volume of traffic PRF is sufficient to totally halt wireless communication under our two topologies. Our attack approach DDoS is stronger method rather than DoS targeted specific AP. If the BM has sufficient zombie devices around the world, the DDoS will play havoc with the communication.

Chapter 4. Mitigation Method against our DDoS Attack Model

In this chapter, we start with some mathematical backgrounds and then introduce previous researches on location privacy protection method including location obfuscation and k-anonymity with their limitations. After that we propose a location information protection method for user to prevent the attacker from clear target of the region specific setting. Our mitigation method is secure and effective location information protection method based on chameleon hash scheme which never reveals accurate location of user to anybody. And we also present performance and security analysis of our proposed method. Lastly, we will summarize our mitigation method.

4.1 Mathematical Background

In this section, we briefly introduce mathematical backgrounds such as number theoretic problems, proofs of knowledge based on discrete logarithms, chameleon hash and attack model on pseudorandomness used in this dissertation.

4.1.1 Number Theoretic Problems

We firstly define groups, finite groups, order and generators as follows:

Definition 4.1. Groups
A *group* is a set $\mathbb{G}$ of elements together with a binary operation '$\cdot$' such that satisfies following four conditions.

1. **Closure**: $a, b \in \mathbb{G} \rightarrow a \cdot b = c \in \mathbb{G}$.
2. **Associativity**: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. **Identity**: $\forall a \in \mathbb{G}, i \cdot a = a \cdot i = a \rightarrow i \in \mathbb{G}$ is an identity element.
4. **Inverse**: $\forall a \in G$, $\exists a^{-1}$ such that $a \cdot a^{-1} = i$

Definition 4.2. Finite Groups
A *group* $(\mathbb{G}, \cdot)$ is ***finite*** if it has a finite number of g elements. We denote the cardinality of $\mathbb{G}$ by $|\mathbb{G}|$.

Definition 4.3. Order
The ***order*** of an element $a \in \mathbb{G}$ is smallest positive integer n such that $a \cdot a \cdots a = a^n = i$.

Definition 4.4. Generators
A *group* $\mathbb{G}$ which contains elements g with maximum order is equal to $|\mathbb{G}|$, then it is said to be ***cyclic***. The elements are called ***generators***.

Let $\mathbb{G} = \langle g \rangle$ be a cyclic multiplicative group with generator g and ordered prime number q. We present the hard problems in a group $\mathbb{G}$ as follows:

- Discrete Logarithm Problem (**DLP**) : Given $h \in \mathbb{G}$, to compute $x \in \mathbb{Z}_q^*$, such that $h = g^x$.
- Computational Diffie-Hellman Problem (**CDHP**) : Given $g^x, g^y \in \mathbb{G}$ for $x, y \in \mathbb{Z}_q^*$, to compute g^{ab}.
- Decisional Diffie-Hellman Problem (**DDHP**) : Given $g^x, g^y, g^z \in \mathbb{G}$ for $x, y, z \in \mathbb{Z}_q^*$, to determine whether $z = xy$ or not.
- Gap Diffie-Hellman Problem (**GDHP**) : Given $g^x, g^y \in \mathbb{G}$ for $x, y \in \mathbb{Z}_{\shortparallel}{}^*$, to compute g^{xy} using **DDH Oracle**.
  - If **CDHP** can be solved, it is also possible to be solved **DDHP**. However, the equivalent of two problems is not proven yet. Okamoto and Pointcheval [25] suggest the possibility of a signature existence, if there is the difference of solving difficulty between **CDHP** and **DDHP**. They coined a group $\mathbb{G}$ is a **GDH** group: if **DDHP** can be solved in polynomial time, the **CDHP** cannot be solved in polynomial time with non-negligible probability. Also they coined the name of the problem is **GDHP**. There exists no GDH group except hyper or super singular elliptic curve over finite fields based on Weil paring. For more concrete concepts about **GDH**, see [25, 24, 23].

4.1.2 Proofs of Knowledge

The proof of knowledge of a discrete logarithm is due to basically Schnorr [26]. Let $H : \{0,1\}^* \to \{0,1\}^k$ be a collision-resistant hash function (***CRHF***). A prover (P) whose input is (g, q, y, x) wants to prove a verifier (V) whose input is (g, q, y). Only the P holds a secret $x \in \mathbb{Z}_q$ and requires to prove the V that $x = \log_g y$ without revealing x. More precisely, the P interacts with the V on a message (g, y) as follows:

1. The P randomly chooses a number $r {\in}_R \mathbb{Z}_q$.
2. The P computes $c = H(g, y, g^r)$ and $s = r - cx$.
3. The V outputs 'yes' or 'no' depending on whether holds or not $c = H(g, y, g^s y^c)$.

The equality of two discrete logarithms, which is originally proposed by Chaum and Pedersen [31]. Let $H : \{0,1\}^* \to \{0,1\}^k$ be a ***CRHF***. The P whose input it (g, h, q, y, z, x) requires to prove the V whose input is (g, h, q, y). Only the P holds a secret $x \in \mathbb{Z}_q$ and requires to prove the V that $x = \log_g y$ $= \log_h z$ without revealing x. More precisely, The P interacts with the V on a message $(g, y), (g, z)$ as follows:

1. The P randomly chooses a number $r {\in}_R \mathbb{Z}_q$.

2. The P computes $c = H(g, h, y, z, g^r, h^r)$ and $s = r - cx$.

3. The V outputs 'yes' or 'no' depending on whether holds or not $c = H(g, h, y, z, g^s y^c, h^s z^c)$.

The formal definition of a proof of knowledge of the discrete logarithm and the equality of two discrete logarithms as follows:

Definition 4.5. If a pair $(c, s) \in \{0, 1\}^k$ satisfies $c = H(g, y, g^s y^c)$, it is called the the proof of knowledge of the discrete logarithm of every y in a cyclic group $\langle g \rangle$.

Definition 4.6. If a pair $(c, s) \in \{0, 1\}^k$ satisfies $c = H(g, h, y, z, g^s y^c, h^s z^c)$, it is called the proof of knowledge of the discrete logarithm of every y, z in cyclic groups $\langle g \rangle$ and $\langle h \rangle$.

From **Definition 4.6**, we can easily determine whether $\langle g, y, h, z \rangle$ is a valid Diiffie-Hellman tuple of (c, s) or not.

4.1.3 Chameleon Hash

The chameleon hash function, originally introduced by Krawczyk and Rabin [32], is a trapdoor ***CRHF*** with key pair (K_T, K_H), where K_T is a secret trapdoor key, K_H is a public hash key. For a given input, the chameleon hash [28, 29, 32, 34, 27, 33] is allowed to compute the hash value by anyone who has K_H, but the collision value by only the holder of K_T. First of all, we present an informal description of the chameleon hash as follows:

- Generator g and sufficient big prime number q are generated for security parameters, then generate trapdoor key $K_T = x(x \in \mathbb{Z}_p^*)$, after that we generate hash key $K_H = y = g^x \pmod p$

- Chameleon Hash Computation: $CHash_y(r_0, m_0) = g^{r_0} y^{m_0} \pmod p$

- Collision Computation: $r_1 = r_0 + x(m_0 - m_1) \pmod p$

$$\begin{aligned}
&CHash_y(r_1, m_1)\\
&= g^{r_1} y^{m_1} \pmod p\\
&= g^{r_0 + x(m_0 - m_1)} g^{x m_1} \pmod p\\
&= g^{r_0} g^{x m_0} \pmod p\\
&= CHash_y(r_0, m_0)
\end{aligned}$$

From the informal description of the chameleon hash, we present a formal definition of a chameleon hash as follows:

Definition 4.7. A chameleon hash consists of three efficient probabilistic polynomial-time (PPT) algorithms ($\mathcal{SG}, \mathcal{KG}, \mathcal{HG}$) and one efficient deterministic polynomial-time (DPT) algorithm ($\mathcal{CG}$):

- **System-parameters Generator** $\mathcal{SG}$: A PPT algorithm that inputs security parameters k, and outputs the system parameters SP.

- **Key Generator** $\mathcal{KG}$: A PPT algorithm that inputs SP, and outputs key pair of trapdoor & hash (K_T, K_H).

- **Hash Generator** $\mathcal{HG}_n$: A PPT algorithm that inputs K_H, a customized identity I, a message m_n, and a random value r_n, and outputs the chameleon hash value $h_n = CHash_n(I, m_n, r_n)$.
- **Collision Generator** $\mathcal{CG}_n$: A DPT algorithm that inputs K_T, a message m_n, a random value r_n, a valid chameleon hash value $\mathcal{HG}_n$, and another message $m_{n+1} \neq m_n$, and outputs a value $r_{n+1} = \mathcal{CG}_n(h_n, K_T, I, m_n, r_n, m_{n+1})$ such that $h_{n+1} = CHash_{n+1}(I, m_{n+1}, r_{n+1}) = CHash_n(I, m_n, r_n) = h_n$.

If the value r_n has the uniform random distribution in a finite space $\mathcal{R}$, then the distribution of r_{n+1} satisfies computational indistinguishability from the uniform random distribution in the finite space $\mathcal{R}$.

Also chameleon hash should hold collision resistance, semantic security and key exposure freeness properties as follows:

- **Collision resistance**: Without the knowledge of the trapdoor key K_T, there is no efficient algorithm that satisfies $CHash_{n+1}(I, m_{n+1}, r_{n+1}) = CHash(I, m_n, r_n)$ with non-negligible probability.
- **Semantic security**:
  - $H[X]$: The entropy of a random variable X
  - $H[X|Y]$: The conditional entropy of a random variable X under Y.
  - If $H[X] = H[X|Y]$ is hold, it is called semantically secure. (*i.e.* Let m be a message, and h be a chameleon hash value, then if $H[m] = H[m|h]$ is hold, it is semantically secure chameleon hash.
- **Key exposure freeness**: Although the attacker has the random oracle to access $\mathcal{CG}$ and is permitted to send many selectable queries on triples (I_k, m_k, r_k) where $I_k \neq I$, there is no efficient collision finding algorithm for a given $CHash(I, m, r)$ without the information of K_T.

4.1.4 Affine Space

In order to overcome a vector space in a geometry, we define the affine space which consists of a vector and a point for expressing geometrical region as follows:

Definition 4.8. Affine Space $\mathbb{A}(X)$ to be the set:

$$\mathbb{A}(X) = \left\{ \sum_{i=1}^{l} \lambda_i x_i : x_i \in X, \sum_{i=1}^{l} \lambda_i = 1 \right\}$$

Also we define the convex to express certain points in restricted affine space as follows:

Definition 4.9. Convex $\mathbb{C}(X) \subset \mathbb{A}(X)$ to be the set:

$$\mathbb{C}(X) = \left\{ \sum_{i=1}^{l} \lambda_i x_i : x_i \in X, \sum_{i=1}^{l} \lambda_i = 1, \lambda_i \geq 0 \right\}$$

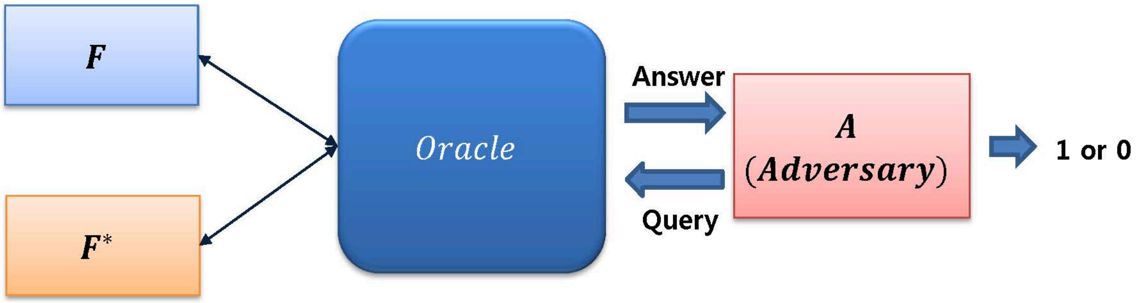


Figure 4.1: Random Oracle Model on Pseudorandomness.

4.1.5 Attack Model on Pseudorandomness

The attack model on pseudorandomness can be depicted as Fig. 4.1. And the formal definition of pseudorandomness as follows:

Definition 4.10. For distinguishing between function $\mathcal{F}$ and complete random function $\mathcal{F}^*$ within same domain, the advantage $Adv_A(\mathcal{F}, \mathcal{F}^*)$ of an attacker defines as follow:

Let $\mathcal{O}^{PR}$ be a *Oracle* for pseudorandomness,

$$Adv_A(\mathcal{F}, \mathcal{F}^*) = |Pr(A = 1|\mathcal{O}^{PR} \leftarrow \mathcal{F}^*) - Pr(A = 1|\mathcal{O}^{PR} \leftarrow \mathcal{F})|.$$

Definition 4.11. Given function $h : \mathbb{N} \to \mathbb{R}$ is *negligible* such that $h(n) < \frac{1}{n^c}$, where constant $c > 0$ and sufficient big natural number n.

Definition 4.12. If $Adv_A(\mathcal{F}, \mathcal{F}^*)$ is *negligible*, function $\mathcal{F}$ satisfies *pseudorandomness*.

4.2 Related Work

Fig 4.2 shows general location based service model between LBS provider and mobile user. With improvement of the location-acquisition technology of smart mobile device, location-based service (LBSs) have been increasing rapidly. While the growth of the LBSs provides convenient services such as point of interests, business localization, customized entertainment, local information, social networks, *etc.*, that have also raised new security challenges about user location privacy surrounding LBS environment. A sensitive information can be revealed from user's location information which contains hospital, bank, lawyer and others location information. Also an adversary can be tracking and profiling of user which include daily routines, favorite places, and so on. Moreover, a user can be a victim of crime at the scene of the incident.

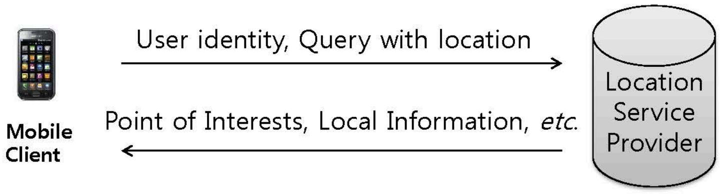

User identity, Query with location
Point of Interests, Local Information, etc.
Mobile Client
Location Service Provider

Figure 4.2: A typical location based service model.

There are a lot of research related to addressing the location privacy protection [36, 37, 38, 39, 40, 41, 42, 43, 44]. The researches can be representatively categorized two types: one is location obfuscation by [44] and the other is k-anonymity by [36]. The location obfuscation provides a cloaking method which modifies original location point $P(x, y)$ into obfuscated region L utilizing three basic operations that is shift (St), enlarge (En), shrink (Sr) and combinations of the operations (*i.e.* $St \cdot En$ means that shift then enlarge operation). Fig. 4.3 shows an example of St operation used in location obfuscation. A user who has original location $P(x, y)$ wants to cloak his location into obfuscated region L using St operation. The original point $P(x, y)$ will be moved into $L(x', y') = St(P(x, y))$. An example of En operation used in location obfuscation is shown in Fig. 4.4. In this example, a user who has original location $P(x, y)$ wants to cloak his location into obfuscated region L using En operation. The original point $P(x, y)$ will be expanded to $L(P(x, y)) = En(P(x, y))$.

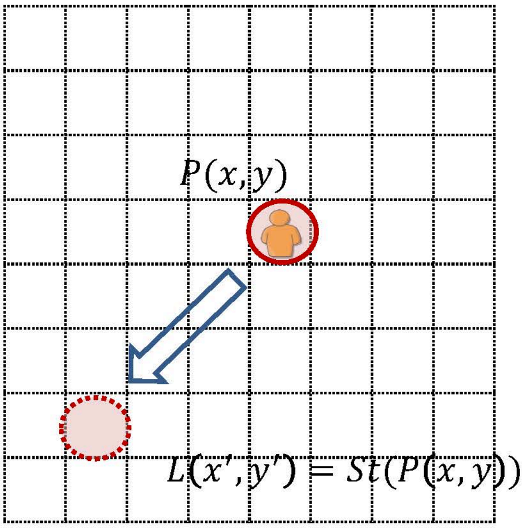


Figure 4.3: An Example of Shift operation used in Location Obfuscation.

Another type of research, namely k-anonymity, is originally used in anonymizing for the record of database. The authors [35] claim that an anonymized record of database should keep unlinkability with another data subject under the assumption of adversary's accessibility to background data such as quasi-identifiers. For the unlinkability, they use the concept of k-anonymity to ensure that each data record could be corresponding to at least k individuals. After the growth of location based services, LBS query has been considered as an record in the database under the concept of k-anonymity. In [36], the authors firstly introduced the possibility of k-anonymity for location privacy. As shown in Fig. 4.5, a user can choose his threshold k for ensuring his level of anonymity. If the user chooses $k = 4$ in a specific location $P(x, y)$, the cloaked region is expanded into $L_1(P(x, y), k = 4)$. In this case the cloaked region L_1 covered at least $k - 1 = 3$ other requests. If the user chooses $k = 8$ in the same location $P(x, y)$, the cloaked region is expanded more into $L_2(P(x, y), k = 8)$. The cloaked region L_2 covered at least $k - 1 = 7$ other requests.

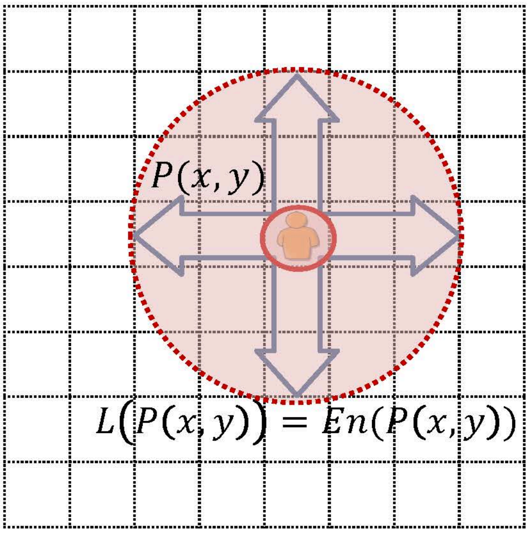


Figure 4.4: An Example of Enlarge operation used in Location Obfuscation.

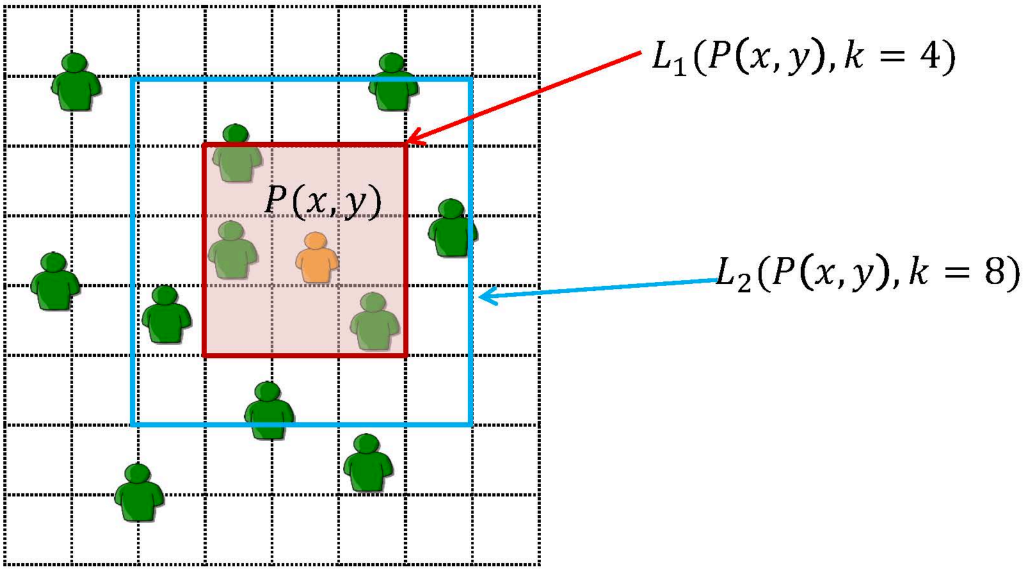


Figure 4.5: An Example of k-Anonymity Methods.

For constructing the system for most previous work in this area, they should adopt anonymizing proxy server (*APS*) which consists of three parts: a cloaking engine, result refining engine, and cloaked location database in their system as an additional trusted third party (TTP). Typical previous architecture of location privacy protection methods can be depicted as Fig. 4.6. A mobile client (*MC*) firstly requests authentication and key agreement to the *APS*, then the *APS* sends those responses to the *MC*. After that the *MC* issues encrypted query with accurate location of the *MC* into the *APS*. The *APS* receives and decrypts the query, and then processes cloaking or obfuscation of the accurate location using a cloaking engine whether location obfuscation, k-anonymity, and so on. The *APS* sends a generated query, which has cloaked region or set of cloaked regions, to location-based service provider (*LBS*) in the next step. The *LBS* chooses an appropriate result, then sends back to the *APS*. After receiving the result from the *LBS*, the *APS* refine the result using the accurate location of the *MC*. After refining, the *APS* encrypts and forwards the refined result to the *MC*. Lastly, the *MC* decrypts the refined result and enjoys the location-based service.

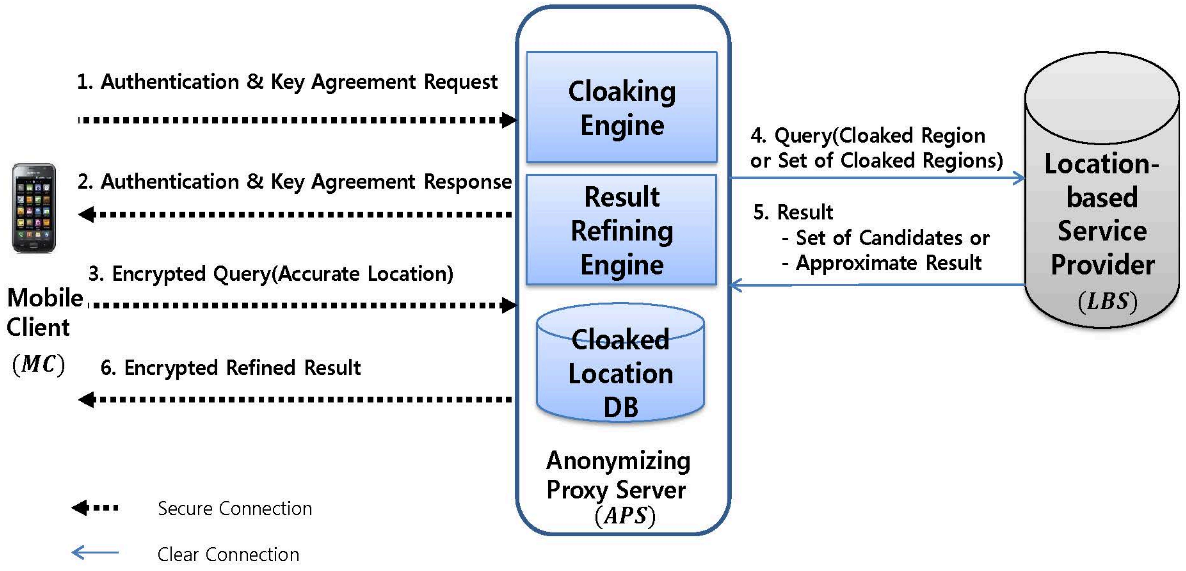

1. Authentication & Key Agreement Request
2. Authentication & Key Agreement Response
3. Encrypted Query(Accurate Location)
6. Encrypted Refined Result
Mobile Client (MC)
Cloaking Engine
Result Refining Engine
Cloaked Location DB
Anonymizing Proxy Server (APS)
4. Query(Cloaked Region or Set of Cloaked Regions)
5. Result - Set of Candidates or - Approximate Result
Location-based Service Provider (LBS)
Secure Connection
Clear Connection

Figure 4.6: Previous Architecture of Location Privacy Protection Methods.

However, previous work have no rigorous proof of security, then the information can be relatively easy to reveal. In case of [44], they proposed a family of adversary models which hold their basic operations and combinations. Even if they claim that a success probability of the de-obfuscation level of the adversary provides better protection than the simple enlargement used by other solutions, the probability is relatively high to be sufficient to reveal the location of the user. In case of [36], it is hard to determine the parameter k because it needs sufficient users in a specific region. Also they just focus on the user's identity, not location information itself. Because an additional TTP must be involved in the architecture of location privacy protection methods in most researches, a user's location information should be revealed to the TTP which can become another big brother. Also encryption and decryption are always executed in every queries between the *MC* and the *APS*. Thus, we propose secure and effective location information protection method based on chameleon hash which never reveals accurate location of user to anybody, and also we present security and performance analysis of our proposed method.

4.3 Proposed Location Information Protection Method

4.3.1 Proposed System Architecture

In order to overcome previous work, we eliminate an additional TTP from the system architecture as shown in Fig. 4.7. A mobile client (*MC*) firstly requests authentication and key agreement to the location-based service provider (*LBS*), then the *LBS* sends those responses to the *MC*. After that the *MC* issues query which contains cloaked region or set of cloaked regions to the *LBS*. The *LBS* chooses an appropriate result which can be set of candidates or approximate result, and sends back to the MC. Lastly, the MC refines the result and enjoys the location-based service. The main differences between previous architecture and ours are that the TTP is eliminated from the architecture and only one secure connection is required.

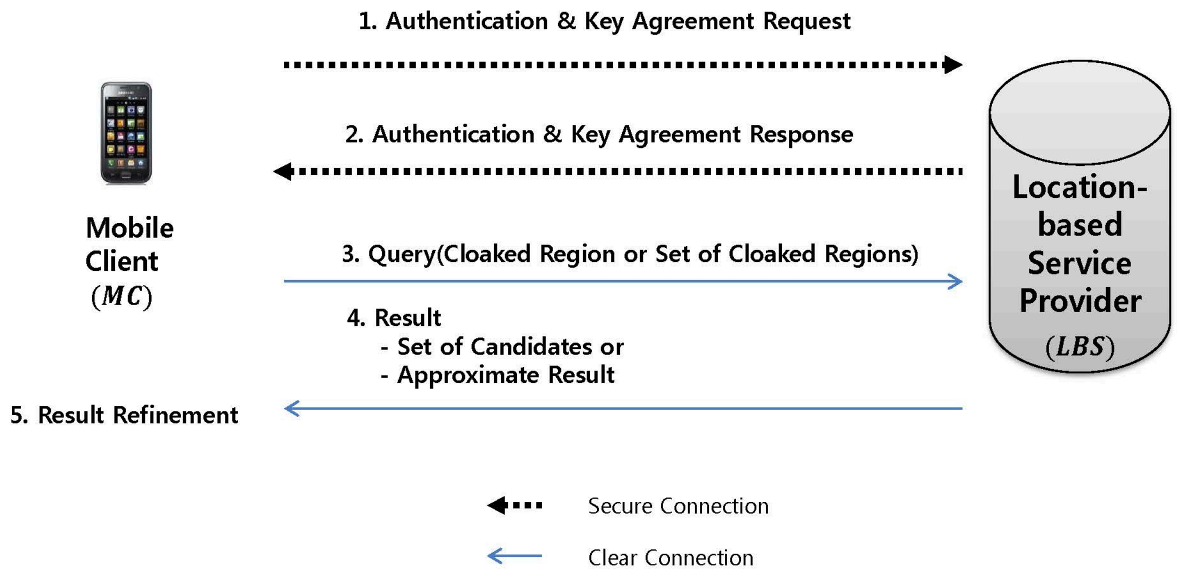

1. Authentication & Key Agreement Request
2. Authentication & Key Agreement Response
Mobile Client (MC)
3. Query(Cloaked Region or Set of Cloaked Regions)
4. Result
- Set of Candidates or
- Approximate Result
5. Result Refinement
Location-based Service Provider (LBS)
Secure Connection
Clear Connection

Figure 4.7: Our Architecture of Proposed Location Information Protection Method.

4.3.2 Setup Phase

We propose setup phase for constructing our location information protection method based on chameleon hash in the **GDH** groups. We refer to [28] chameleon hash constructions. For the construction of key exposure free chameleon hash scheme, the double trapdoor keys must be involved in the constructions. The x is the master trapdoor key and the $h^x = H(y, I)^x$ for each transaction is the ephemeral trapdoor key. The concept of double trapdoor and customized identity I was firstly introduced by Atenies and Medeiros [27].

We describe chameleon hash constructions for the smart mobile device MC and location-based service provider (LBS) as follows:

- **Chameleon hash constructions for the** $\boldsymbol{MC}$ $(\mathcal{SG}, \mathcal{KG}, \mathcal{HG}_n, \mathcal{CG}_n)$:
  - **System-parameters Generator** $\mathcal{SG}$:
    - **Input**: Security parameters $k = \{\mathbb{G}$ (**GDH** group, generator g, ordered prime q) and $H : \{0,1\}^* \rightarrow \mathbb{G}^*$ (CRHF)$\}$
    - **Output**: $SP = \{\mathbb{G}, q, g, H\}$
  - **Key Generator** $\mathcal{KG}$:
    - **Input**: $SP = \{\mathbb{G}, q, g, H\}$

    - ∗ **Output**: Trapdoor & hash key pairs $(K_T, K_H) = (x_c{\in}_R\mathbb{Z}_p^*,\ y_c = g^{x_c})$

  - **Hash Generator** $\mathcal{HG}_n$:

    - ∗ **Input**: Hash key y_c, customized identity I_c, $h = H(y_c, I_c)$, message m_n, random integer $a_n{\in}_R\mathbb{Z}_p^*$, and $r_n = (g^{a_n}, y_c^{a_n})$
    - ∗ **Output**: $CHash_n(I, m_n, r_n) = g^{a_n}h^{m_n}$

  - **Collision Generator** $\mathcal{CG}_n$:

    - ∗ **Input**: Trapdoor key x_c, valid hash value $\mathcal{HG}_n$, message m_n, random integer $a_n{\in}_R\mathbb{Z}_p^*$, customized identity I_c, and $r_n = (g^{a_n}, y_c^{a_n})$
    - ∗ **Output**: $\mathcal{CG}_n(\mathcal{HG}_n, x_c, I_c, m_n, r_n, m_{n+1}) = r_{n+1} = (g^{a_{n+1}}, y_c^{a_{n+1}})$, where $g^{a_{n+1}} = g^{a_n}h^{m_n - m_{n+1}}$ and $y_c^{a_{n+1}} = y_c^{a_n}h^{x_c(m_n - m_{n+1})}$
      Note that $\mathcal{HG}_{n+1} = g^{a_{n+1}}h^{m_{n+1}} = g^{a_n}h^{m_n - m_{n+1}}h^{m_{n+1}} = g^{a_n}h^{m_n} = \mathcal{HG}_n$

- **Chameleon hash constructions for the *LBS*** $(\mathcal{SG}', \mathcal{KG}', \mathcal{HG}_n', \mathcal{CG}_n')$:

  - **System-parameters Generator** $\mathcal{SG}'$:

    - ∗ **Input**: Security parameters $k = \{\mathbb{G}$ (**GDH** group, generator g, ordered prime q) and $H : \{0,1\}^* \to \mathbb{G}^*$ (CRHF)$\}$
    - ∗ **Output**: $SP' = \{\mathbb{G}, q, g, H\}$

  - **Key Generator** $\mathcal{KG}'$:

    - ∗ **Input**: $SP' = \{\mathbb{G}, q, g, H\}$
    - ∗ **Output**: Trapdoor & hash key pairs $(K_T', K_H') = (x_l{\in}_R\mathbb{Z}_p^*,\ y_l = g^{x_l})$

  - **Hash Generator** $\mathcal{HG}_n'$:

    - ∗ **Input**: Hash key y_l, customized identity I_l, $h = H(y_l, I_l)$, message m_n', random integer $a_n'{\in}_R\mathbb{Z}_p^*$, and $r_n' = (g^{a_n'}, y_l^{a_n'})$.
    - ∗ **Output**: $CHash_n'(I_l, m_n', r_n') = g^{a_n'}h^{m_n'}$

  - **Collision Generator** $\mathcal{CG}_n'$:

    - ∗ **Input**: Trapdoor key x_l, valid hash value $\mathcal{HG}_n'$, message m_n', random integer $a_n'{\in}_R\mathbb{Z}_p^*$, customized identity I_l, and $r_n' = (g^{a_n'}, y_l^{a_n'})$
    - ∗ **Output**: $\mathcal{CG}_n'(\mathcal{HG}_n', x_l, I_l, m_n', r_n', m_{n+1}') = r_{n+1}' = (g^{a_{n+1}'}, y_l^{a_{n+1}'})$, where $g^{a_{n+1}'} = g^{a_n'}h^{m_n' - m_{n+1}'}$ and $y_l^{a_{n+1}'} = y_l^{a_n'}h^{x_l(m_n' - m_{n+1}')}$
      Note that $\mathcal{HG}_{n+1}' = g^{a_{n+1}'}h^{m_{n+1}'} = g^{a_n'}h^{m_n' - m_{n+1}'}h^{m_{n+1}'} = g^{a_n'}h^{m_n'} = \mathcal{HG}_n'$

In the first step of the *MC* Setup $(\mathcal{SG}, \mathcal{KG}, \mathcal{HG}_1)$, the smart mobile client generates system parameters $SP = \{\mathbb{G}, q, g, H\}$ and the trapdoor & hash key pairs $(x_c{\in}_R\mathbb{Z}_p^*,\ y_c = g^{x_c})$. And then the client sets his identity, *LBS* identity, query identity, and his accurate location in the customized identity I as [27]. For example, the customized identity is generated by $I_c = H(ID_c, ID_l, ID_q, latitude, longitude)$. After that the message m_1 is assigned for starting query, the random integer a_1 is selected, and $r_1 = (g^{a_1}, y_c{}^{a_1})$ is computed. The last step of the *MC* Setup is the generation of the chameleon hash value using $CHash_1(I_c, m_1, r_1) = g^{a_1}h^{m_1}$, and then the $\mathcal{HG}_1$ will be sent to the *LBS*.

The location-based service provider executes the *LBS* Setup $(\mathcal{SG}', \mathcal{KG}', \mathcal{HG}_1')$ similar to the *MC* Setup phase. The service provider generates system parameters $SP' = \{\mathbb{G}, q, g, H\}$ and the trapdoor & hash key pairs $(x_l \in_R \mathbb{Z}_p^*,\ y_l = g^{x_l})$. And then the service provider sets his identity, *MC* identity, and query identity in the customized identity I_l. For example, the customized identity is generated by $I_l = H(ID_l, ID_c, ID_q)$. After that the message m_1' is assigned for starting query, the random integer a_1' is selected, and $r_1' = (g^{a_1'}, y_c{}^{a_1'})$ is computed. The last step of the *LBS* Setup is the generation of the chameleon hash value using $CHash_1'(I_l, m_1', r_1') = g^{a_1'}h^{m_1'}$, and then the $\mathcal{HG}_1'$ will be sent to the *MC*.

The setup phases are executed simultaneously during authentication & key agreement request/response between the *MC* and LBS. Within this secure connection, the I_c and $\mathcal{HG}_1$ are sent to the *LBS* and the I_l and $\mathcal{HG}_1'$ are sent to the *MC*. After that the secure connection can be released for improvements in efficiency at the query protocol. The setup phase is depicted as the sequences from the step '0-1' to '5.' in Figs. 4.8 and 4.9.

4.3.3 Query Protocol

[Query Protocol I]: Query with single obfuscated region

After the setup phases, the [Query protocol I] starts with generating a collision and a chameleon hash value as shown in Fig. 4.8. The *MC* generates the collision using $\mathcal{CG}_1 = (\mathcal{HG}_1, x_c, I_c, m_1, r_1, m_2) = r_2 = (g^{a_2}, y_l^{a_2})$, where $g^{a_2} = g^{a_1}h^{m_1 - m_2}$ and $y_l^{a_2} = y_l^{a_1}h^{x_l(m_1 - m_2)}$ and the chameleon hash value using $\mathcal{HG}_2 = CHash_2(I_c, m_2, r_2)$. In this step, we generate obfuscated location with geometrically corresponding a convex from the accurate location of the *MC*. Let $\mathbb{C}$ be a convex which contains the accurate location of the *MC*. The $\mathbb{C}_n$ is denoted a convex pairing with the chameleon hash value $\mathcal{HG}_n$. The *MC* chooses the size of the convex according to the accuracy of the location-based service and the level of location privacy. Let $Y : \{0,1\}^* \rightarrow \mathbb{G}^*$ be a ***CRHF***. The obfuscated location $\mathcal{O}(latitude_n)$ and $\mathcal{O}(longitude_n)$ can be generated by $Y(latitude_n)$ and $Y(longitude_n)$ of the *MC* that should satisfy two conditions as follows:

1. $\{\mathcal{O}(latitude_n) = Y(latitude_n),\ \mathcal{O}(longitude_n) = Y(longitude_n)\} \in \mathbb{C}_n$

2. $m_n = \{\mathcal{O}(latitude_n), \mathcal{O}(longitude_n), \mathbb{C}_n\}$ should satisfy the property of chameleon hash generator $\mathcal{HG}_n = g^{a_n}h^{m_n} = g^{a_{n-1}}h^{m_{n-1}-m_n}h^{m_n} = g^{a_{n-1}}h^{m_{n-1}} = \mathcal{HG}_{n-1}$

In case of single obfuscated region, the parameter n can be assigned 2 as shown in Fig. 4.8. After generating $\mathcal{HG}_2$, the *MC* sends $m_2, g^{a_2}, y_l^{a_2}$, and $\mathcal{HG}_2$ to the *LBS*. Given g^{a_2}, the *LBS* firstly verifies whether $g^{a_2} = y_l^{a_2}$ or not. If the verification is failed, the *LBS* rejects the query of the *MC*. If the verification

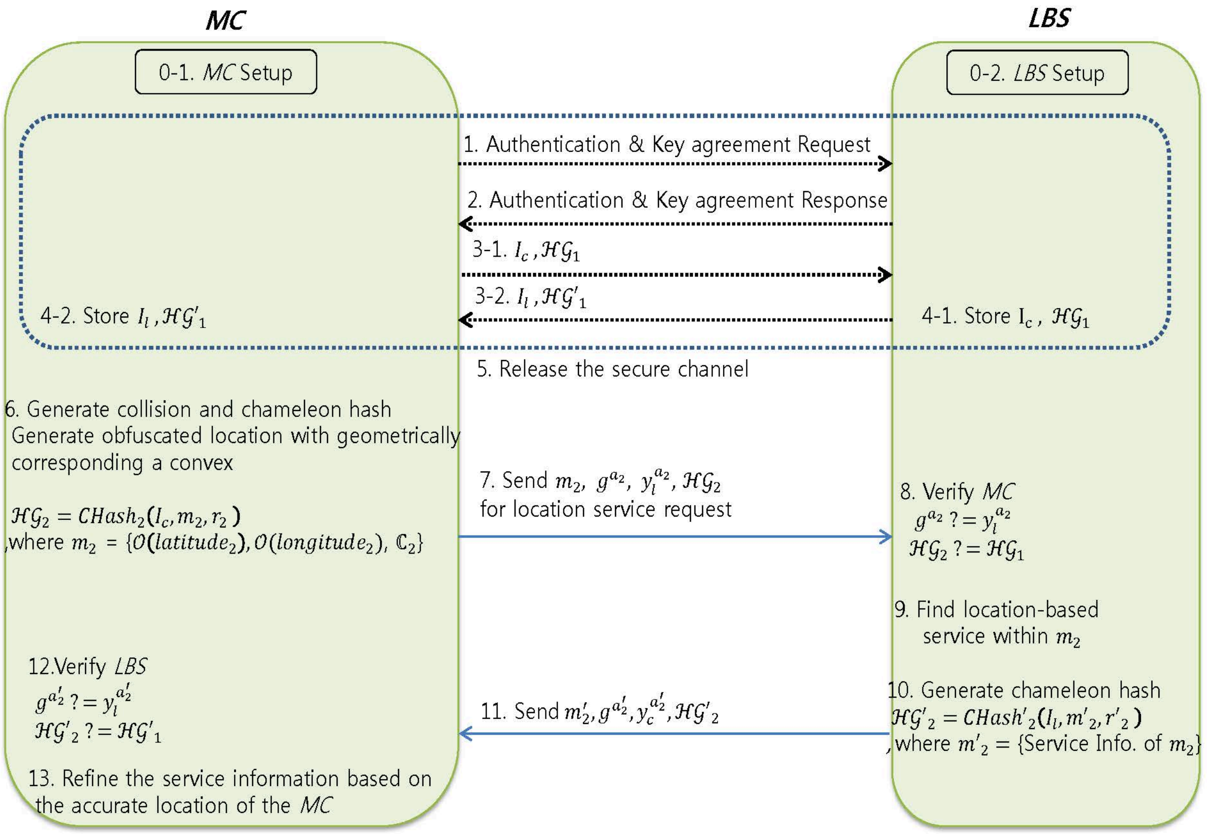

MC
LBS
0-1. *MC* Setup
0-2. *LBS* Setup
1. Authentication & Key agreement Request
2. Authentication & Key agreement Response
3-1. $I_c, \mathcal{HG}_1$
3-2. $I_l, \mathcal{HG}'_1$
4-2. Store $I_l, \mathcal{HG}'_1$
4-1. Store $\mathrm{I_c}$, $\mathcal{HG}_1$
5. Release the secure channel
6. Generate collision and chameleon hash
Generate obfuscated location with geometrically corresponding a convex
$\mathcal{HG}_2 = CHash_2(I_c, m_2, r_2)$
,where m_2 = $\{O(latitude_2), O(longitude_2), \mathbb{C}_2\}$
7. Send m_2, g^{a_2}, $y_l^{a_2}$, $\mathcal{HG}_2$
for location service request
8. Verify *MC*
g^{a_2} ? = $y_l^{a_2}$
$\mathcal{HG}_2$? = $\mathcal{HG}_1$
9. Find location-based service within m_2
10. Generate chameleon hash
$\mathcal{HG}'_2 = CHash'_2(I_l, m'_2, r'_2)$
,where m'_2 = {Service Info. of m_2}
11. Send $m'_2, g^{a'_2}, y_c^{a'_2}, \mathcal{HG}'_2$
12.Verify *LBS*
$g^{a'_2}$? = $y_l^{a'_2}$
$\mathcal{HG}'_2$? = $\mathcal{HG}'_1$
13. Refine the service information based on the accurate location of the *MC*

Figure 4.8: [Query Protocol I]: Query with single obfuscated region

is succeeded, the *LBS* computes the chameleon hash value $\mathcal{HG}_2 = g^{a_2}h^{m_2}$ and then verifies it with previously stored chameleon hash value $\mathcal{HG}_1$. After that the *LBS* finds an appropriate location-based service within m_2, then he generates his chameleon hash value $\mathcal{HG}'_2$, where $m'_2 = \{$Service Information of $m_2\}$. He sends $m'_2, g^{a'_2}, y_c^{a'_2}, \mathcal{HG}'_2$ to the *MC*. The *MC* verifies the *LBS* in the same way of the *LBS*'s verification. Lastly, the *MC* refine the service information based on the his knowledge of the accurate location.

[Query Protocol II]: Query with set of obfuscated regions

The [Query Protocol II] is almost same as [Query Protocol I] except the redundancy of the location information. After the setup phases, the [Query protocol II] starts with generating collisions and chameleon hash values as shown in Fig. 4.9. The *MC* generates the collisions using $\mathcal{CG}_n(\mathcal{HG}_n, x_c, I_c, m_n, r_n, m_{n+1}) = r_{n+1} = (g^{a_{n+1}}, y_c^{a_{n+1}})$, where $g^{a_{n+1}} = g^{a_n}h^{m_n - m_{n+1}}$ and $y_c^{a_{n+1}} = y_c^{a_n}h^{x_c(m_n - m_{n+1})}$ and the chameleon hash value using $\mathcal{HG}_n = CHash_n(I_c, m_n, r_n)$. In this step, we generate obfuscated location with geometrically corresponding set of convex from the accurate location of the *MC*. Let $\mathbb{C}$ be a convex which contains the accurate location of the *MC*. The $\mathbb{C}_n$ is denoted a convex pairing with the chameleon hash value $\mathcal{HG}_n$. The *MC* chooses the size of the convex according to the accuracy of the location-based service and the level of location privacy. Let $Y : \{0,1\}^* \rightarrow \mathbb{G}^*$ be a ***CRHF***. The obfuscated location

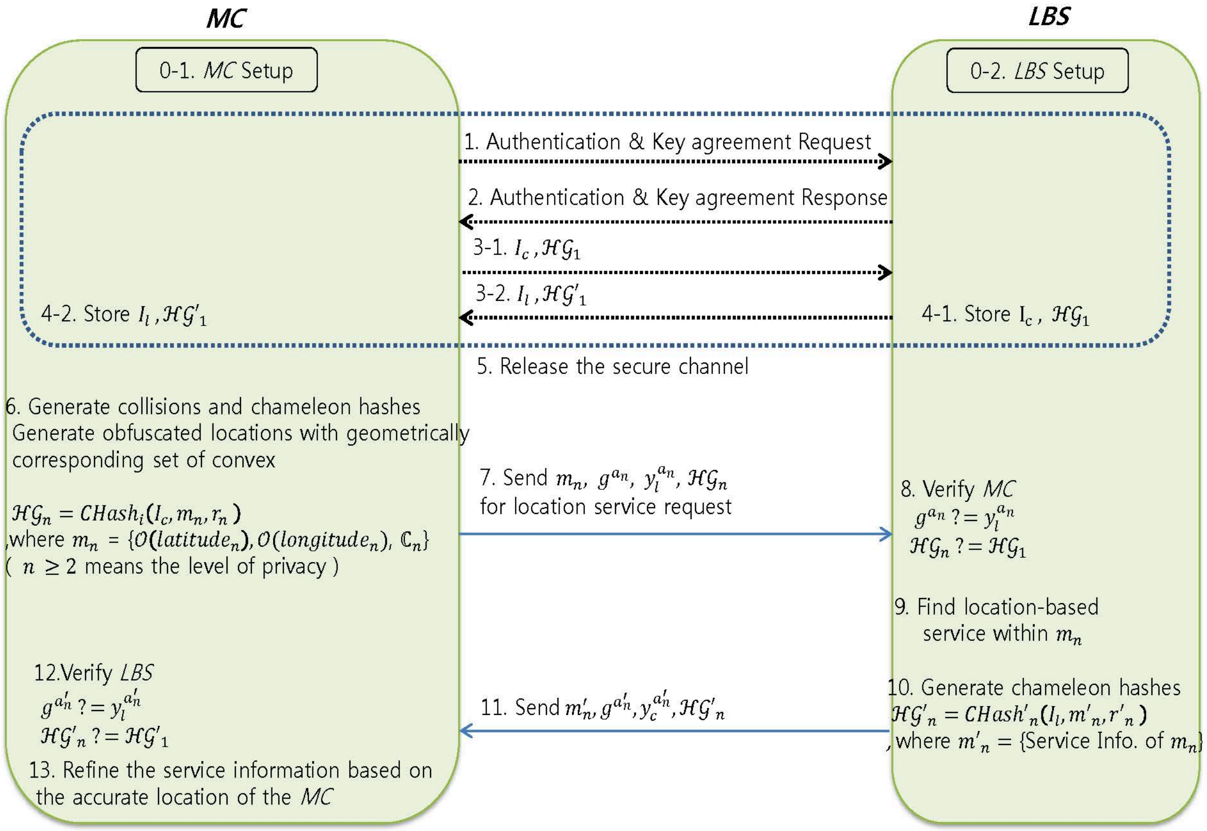

MC
LBS
0-1. MC Setup
0-2. LBS Setup
1. Authentication & Key agreement Request
2. Authentication & Key agreement Response
3-1. I_c , $\mathcal{HG}_1$
3-2. I_l , $\mathcal{HG}'_1$
4-2. Store I_l , $\mathcal{HG}'_1$
4-1. Store I_c , $\mathcal{HG}_1$
5. Release the secure channel
6. Generate collisions and chameleon hashes
Generate obfuscated locations with geometrically corresponding set of convex
$\mathcal{HG}_n = CHash_i(I_c, m_n, r_n)$
,where m_n = {$\mathcal{O}(latitude_n), \mathcal{O}(longitude_n)$, $\mathbb{C}_n$}
($n \geq 2$ means the level of privacy)
7. Send m_n, g^{a_n}, $y_l^{a_n}$, $\mathcal{HG}_n$
for location service request
8. Verify MC
$g^{a_n} ?= y_l^{a_n}$
$\mathcal{HG}_n ?= \mathcal{HG}_1$
9. Find location-based service within m_n
10. Generate chameleon hashes
$\mathcal{HG}'_n = CHash'_n(I_l, m'_n, r'_n)$
, where m'_n = {Service Info. of m_n}
11. Send $m'_n, g^{a'_n}, y_c^{a'_n}, \mathcal{HG}'_n$
12.Verify LBS
$g^{a'_n} ?= y_l^{a'_n}$
$\mathcal{HG}'_n ?= \mathcal{HG}'_1$
13. Refine the service information based on the accurate location of the MC

Figure 4.9: [Query Protocol II]: Query with set of obfuscated regions

$\mathcal{O}(latitude_n)$ and $\mathcal{O}(longitude_n)$ can be generated by $Y(latitude_n)$ and $Y(longitude_n)$ of the *MC* that should satisfy two conditions as same as the [Query Protocol I]. After generating $\mathcal{HG}_n$ where $n \geq 2$ (the level of privacy), the *MC* sends $m_n, g^{a_n}, y_l^{a_n}$, and $\mathcal{HG}_n$ to the *LBS*. Given g^{a_n}, the *LBS* firstly verifies whether $g^{a_n} = y_l^{a_n}$ or not. If the verification is failed, the *LBS* rejects the query of the *MC*. If the verification is succeeded, the *LBS* computes the chameleon hash value $\mathcal{HG}_n = g^{a_n}h^{m_n}$ and then verifies it with previously stored chameleon hash value $\mathcal{HG}_1$. After that the *LBS* finds an appropriate location-based service within m_n, then he generates his chameleon hash value $\mathcal{HG}'_n$, where m'_n = {Service Information of m_n}. He sends $m'_n, g^{a'_n}, y_c^{a'_n}, \mathcal{HG}'_n$ to the *MC*. The *MC* verifies the *LBS* in the same way of the *LBS*'s verification. Lastly, the *MC* choose an appropriate result from set of convex which contains n faked locations based on the *MC*'s level of privacy, and then refine the service information based on the his knowledge of the accurate location.

4.4 Analysis

In this section, we present rigorous performance and security analysis of our proposed method.

4.4.1 Performance Analysis

We make a comparison between previous work and our proposed location information protection method. The number of authentication, key agreement and clear communication are the same number in the comparison. The number of secure communication in previous work is $N * 2$ and only 2 secure communications are sufficient in our protocol. The main differentiation between previous work and our protocols is that the existence of an additional TTP, which connotes providing the complete assurance without revealing the location information of users. Due to the elimination of TTP, only the user manages and controls his location information securely. Even if the mobile device requires more cryptographic operations in a group $\mathbb{G}$ for our proposed method, the smart mobile device can be handled the operations with the improvement of computing power. Moreover, we reduce the communication overheads in our architecture, the remain computing power can be utilized sufficiently in the additional cryptographic computation. The result of comparison is summarized in Table 4.1.

Table 4.1: Performance Comparison with Previous work

	Previous Work	Our Protocol I	Our Protocol II
#. of Authentication	1	1	1
#. of Key Agreement	1	1	1
#. of Clear Communication	$N^{1)} * 2$	$N * 2$	$N * 2$
#. of Secure Communication	$N * 2$	2	2
Additional TTP	Yes	No	No
Repository of User Location	TTP	User's device	User's device
Computation Overhead of User's device	Encryption & Decryption $N * 2 * S^{2)}$	Chameleon Hash $N * (3E + M)$	Chameleon Hash $i^{3)} * N * (3E^{4)} + M^{5)})$

1) N: #. of queries, 2) S: a symmetric cipher operation (*i.e.* AES), 3) i: Level of privacy

4) E: The exponentiation in $\mathbb{G}$, 5) M: The multiplication in $\mathbb{G}$

4.4.2 Security Analysis

We present rigorous security analysis of our proposed method in this subsection. The outline of our proof is that we utilize four lemmas for proving the main theorem which guarantees the security of our method.

Lemma 4.1. *The chameleon hash function in proposed location information protection method is collision resistance under the assumption of the* ***CDHP*** *in a group $\mathbb{G}$ is intractable.*

Proof. We can prove this lemma by contradiction. We assume that there is a PPT algorithm $\mathcal{A}$, with a non-negligible probability, that outputs $\{(m_0, r_0), (m_1, r_1)\}$ pairs which satisfy $\mathcal{HG}_1 = CHash_1(I, m_1, r_1) = CHash_0(I, m_0, r_0) = \mathcal{HG}_0$. In other words, it can be expressed $g^{a_1}h^{m_1} = g^{a_0}h^{m_0}$, then we can efficiently compute $h^x = (\frac{y^{a_1}}{y^{a_0}})^{(m_0-m_1)^{-1}}$. Note that h^x is a signature on a message I (*i.e.,* customized identity) in **GDH** [24], which was proved the security against existential forgery under an adaptive chosen-message attack on the **CDHP** in a group $\mathbb{G}$ is intractable. Thus, the chameleon hash function in proposed location information protection method is collision resistance under the assumption that the **CDHP** in a group $\mathbb{G}$ is intractable. □

Lemma 4.2. *The chameleon hash function in proposed location information protection method provides semantic security.*

Proof. Let I be a customized identity. if there exists '1-to-1 correspondence' between the random value r_i and the chameleon hash value $\mathcal{HG}_i$ for given message m_i, then the conditional probability is $Pr(m_i|\mathcal{HG}_i) = Pr(m_i|r_i)$. Due to the independence between r_i and m_i, the probability can be reduced into $Pr(m_i|\mathcal{HG}_i) = Pr(m_i)$. And then we can prove $H[m_i|\mathcal{HG}_i] = H[m_i]$ as follows:

$$\begin{aligned} H[m_i|\mathcal{HG}_i] &= -\sum_{i=1}^{n}\sum_{\mathcal{HG}_{i=1}}^{n} Pr(m_i, \mathcal{HG}_i)\text{log}(Pr(m_i|\mathcal{HG}_i)) \\ &= -\sum_{i=1}^{n}\sum_{\mathcal{HG}_{i=1}}^{n} Pr(m_i, \mathcal{HG}_i)\text{log}(Pr(m_i)) \\ &= -\sum_{i=1}^{n} Pr(m_i)\text{log}(Pr(m_i)) \\ &= H[m_i] \end{aligned}$$

Therefore, chameleon hash function in proposed location information protection method provides semantic security. □

Lemma 4.3. *The obfuscated location in proposed location information protection method satisfies pseudorandomness.*

Proof. Given obfuscated location pairs $\{\mathcal{O}(latitude_n), \mathcal{O}(longitude_n)\}$ are generated by a full domain collision resistance hash $Y : \{0,1\}^* \in \mathbb{G}^*$. From **Definition 4.10**, the advantage of the adversary A can be defined as $Adv_A(\mathcal{F}, \mathcal{F}^*) = |Pr(A = 1|\mathcal{O}^{PR} \leftarrow \mathcal{F}^*) - Pr(A = 1|\mathcal{O}^{PR} \leftarrow \mathcal{F})|$, which was proved that the advantage of the collision-resistant hash function is equal to the advantage of the random oracle. (*i.e.,* Let F be a SHA$_n$, where $n = 256$ $Adv_A(\mathcal{F}, \mathcal{F}^*) = 1/2 - 1/2^n = 1/2 - 1/2^{256}$ is negligible. Moreover, Boldyreva and Kumar [30] introduced new pseudorandom generator from collision-resistant hash functions. Therefore, the obfuscated location in proposed location information protection method satisfies pseudorandomness.

□

Lemma 4.4. *The chameleon hash function in proposed location information protection method provides key exposure freeness.*

Proof. Although the adversary can make a series of queries on triples $(I_n, m_n, g^{a_n}, y^{a_n})$, there exists no efficient algorithm finding a collision of the chameleon hash value $\mathcal{HG}_0 = CHash_0(I_0, m_0, g^{a_0}, y^{a_0})$, where $I_0 \neq I_n$. Note that h^x is a signature on a message I (*i.e.,* customized identity) in **GDH** [24], and the success probability of finding collision queries is equal to the success probability of breaking the signature. However, it was proved the security against existential forgery under an adaptive chosen-message attack on the **CDHP** in a group $\mathbb{G}$ utilizing random oracle model. □

Theorem 4.5. *If the proposed method satisfies the properties: collision resistance, semantically secure, pseudorandomness, and key exposure freeness, then it is secure.*

Proof. The proposed method satisfies collision resistance (see **Lemma 4.1**), semantically secure (see **Lemma 4.2**), pseudorandomness (see **Lemma 4.3**), and key exposure freeness (see **Lemma 4.4**). Therefore, the proposed method is secure. □

4.5 Summary

For the mitigation method against our DDoS attack model, we propose secure and effective location information protection method which can prevent the attacker from clear target of the region specific setting. The proposed method is based on chameleon hash without an additional TTP. So our proposed method never reveals accurate location of user to anybody, and also we present rigorous security proof with reducing the communication overheads as being described in the performance analysis. Our method can help to prevent exploiting location information against Botnet master in our DDoS attack model.

Chapter 5. Conclusion and Open Problem

Connecting the smart mobile device to the wireless network is a trend that cannot be avoided. While widespread deployment and expansion of wireless network infrastructure brings the wireless data communication ubiquitously, it has also introduced many security challenges such as DoS/DDoS attack, malicious mobile code, *etc.*, surrounding wireless networks.

In this dissertation, we have investigated a novel DDoS attack model exploiting location information of smart mobile device and utilizing active discovery vulnerability with unrecognizable volume of traffic and reasonable interval of the request in wireless network. Furthermore, our attack can be forcefully performed in targeted specific region, where is a primary target of the attacker such as government organization, bank, *etc.*, based on the location information of the user. Without any costly devices, our attack model can be played as jammer-like DDoS attack. We have evaluated the proof-of-concept of the regional targeted DDoS attack model using discrete network simulation and its analytical result. Due to our evaluation, our attack model is hard to detect and prevent under the current countermeasures such as WIPS in wireless networks, and also more powerful on-line attack is possible. Furthermore, our simulation results show that relatively small number of attacker STAs with tiny volume of traffic is sufficient to totally interrupt wireless data communication under our two simulation topologies.

In order to mitigate our attack model, we need location information protection for user to prevent the attacker from clear target of the region specific setting. Previous researches on location privacy protection have no rigorous proof of security, then the information can be relatively easy to reveal. Also an additional TTP is a mandatory component in the previous architecture, then the information can be revealed to the TTP. To overcome these limitations, we propose secure and effective location information protection method based on chameleon hash under **GDHP** which never reveals accurate location of user to the TTP. Also we present rigorous security proof and show the reduction of the communication overheads as being described in the performance analysis.

In order to conclude this dissertation, we discuss open problems in this area. First, we plan to extend and evaluate our attack model into various wireless network models such as cellular, Ad-Hoc, Mesh, and so on. Also we intend to implement our idea on the real smart mobile devices and evaluate it empirically in a practical wireless network. After checking the proof-of-concept in real and various environment, we need to identify our attack model whether to be a potential Advanced Persistent Threat (APT) or not. Second, proposed location information protection method is a completely different approach unlike previous work. So we require to establish an efficient measure of the degree of location privacy between our method and others. Lastly, we need to research related to diverse countermeasures against our DDoS attack.

References

[1] R. Housley and W. Arbaugh, "Security problems in 802.11-based networks," *Communications of the ACM*, 46(5):31-34, 2003.

[2] J. Bellardo and S. Savage, "802.11 Denial-of-Service Attacks : Real vulnerabilities and practical solutions," *In Proceedings of USENIX Security Symposium*, pages 15-28, Washington, D.C., US, August 2003.

[3] A. Mishra, N. L. Petroni Jr., W. A. Arbaugh and T. Fraser, "Security issues in IEEE 802.11 wireless local area networks : a survey," *Wireless Communications and Mobile Computing*, 4:821-833, Wiley InterScience, 2004.

[4] D. Kashiwa, E.Y. Chen, H. Fuji, S. Machida, H. Shigeno, K. Okada and Y. Matsushita, "Active countermeasure platform against DDoS attacks," *IEICE TRANS. INF. & SYST.*, Vol.E85-D, No.12, pp.1918-1928, Dec 2002.

[5] B. Waters, A. Juels, J. A. Halderman and E.W. Felten, "New Client Puzzle Outsourcing Techniques for DoS Resistance," *In Proceedings of ACM Conference on Computer and Communications Security (CCS 2004)*, Washington, D.C., US, October 2004.

[6] A. Matrawy, P.C. van Oorschot and A. Somayaji, "Mitigating Network Denial-of-Service Through Diversity-Based Traffic Management," *In Proceedings of Applied Cryptography and Network Security (ACNS'05)*, New York, US, June 2005.

[7] IEEE LAN/MAN Standards Committee (IEEE 802), "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2007 revision)," *IEEE Standards Association*, 2007.

[8] O.P. Sarmiento, F.G. Guerrero and D.R. Argote, "Basic security measures for IEEE 802.11 wireless networks," *Revista Ingenieria E Investigacion*, 28(2):89-96, 2008.

[9] M. Bernaschi, F. Ferreri and L. Valcamonici, "Access points vulnerabilities to DoS attacks in 802.11 networks," *Wireless Networks*, 14(2):159-169, Kluwer Academic Publishers, 2008.

[10] K. Bicakci and B. Tavli, "Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks," *Computer Standards & Interfaces*, 31:931-941, Elsevier, 2008.

[11] J. Li, N. Li, X. Wang and T. Yu, "Denial of Service Attacks and Defenses in Decentralized Trust Management," *International Journal of Information Security*, 8(2):89-101, Springer-Verlag, 2009.

[12] X. Liu, X. Yang and Y. Xia, "NetFence: Preventing Internet Denial of Service from Inside Out," *In Proceedings of ACM SIGCOMM 2010*, New Delhi, INDIA, August 2010.

[13] W. Xu, W. Trappe, Y. Zhang and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," *In Proceedings of MobiHoc 05*, pp. 46-57, Urbana-Champaign, Illinois, US, May 2005.

[14] IEEE LAN/MAN Standards Committee (IEEE 802), "IEEE 802.11w: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Protected Management Frames. (2009 revision)," *IEEE Standards Association*, 2009.

[15] IEEE LAN/MAN Standards Committee (IEEE 802), "IEEE 802.11i-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Standards Association*, 2004-07-23.

[16] G. Castignani, N. Montavont and A. Arcia-Moret, "Analysis and evaluation of WiFi scanning strategies," *In Proceedings of the 5th Conference on Electrical Engineering*, Merida,2010.

[17] W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, "A study of android application security," *In Proceedings of the 20th USENIX conference on Security (SEC'11).* USENIX Association, Berkeley, CA, USA, 2011.

[18] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi and Z. Tianning, "Andbot: Towards Advanced Mobile Botnets," *In Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats (LEET'11)*, USENIX Association, Berkeley, CA, USA, 2011.

[19] A. Porter Felt, M. Finifter, E. Chin, S. Hanna and D. Wagner, "A survey of mobile malware in the wild," *In Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices (SPSM '11)*, pp.3-14, ACM, New York, NY, USA, 2011.

[20] J. Hua and K. Sakurai, "A SMS-based mobile Botnet using flooding algorithm," *In Proceedings of the 5th IFIP WG 11.2 international conference on Information security theory and practice: security and privacy of mobile devices in wireless communication (WISTP'11)*, LNCS 6633, pp.264-279, Springer-Verlag, 2011.

[21] M. Tanabe, H. Akaike, M. Aida, M. Murata and M. Imase, "Adaptive timer-based countermeasures against TCP SYN flood attacks," *IEICE TRANS. COMMUN.*, Vol.E95-B, No.3, pp.866-875, March 2012.

[22] M. Knysz, H. Xin, Z. Yuanyuan and K.G. Shin, "Open WiFi networks: Lethal weapons for botnets?," *In Proceedings of IEEE INFOCOM 2012*, pp.2631-2635, 25-30, March 2012.

[23] P. Barreto, H. Kim, B. Lynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *In Proceedings of Advances in Cryptology-Crypto 2002*, LNCS 2442, pp.354–368, Springer-Verlag, 2002.

[24] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the Weil pairings," *In Proceedings of Advances in Cryptology-Asiacrypt 2001*, LNCS 2248, pp.514–532 Springer-Verlag, 2001.

[25] T. Okamoto and D. Pointcheval, "The gap-problems: a new class of problems for the security of cryptographic Schemes," *In Proceedings of PKC 2001*, LNCS 1992, pp. 104–118, Springer-Verlag 2001.

[26] C.P. Schnorr, "Efficient signature generation for smart cards," *Journal of Cryptology* 4(3), pp. 239-252, Springer-Verlag 1991.

[27] G. Ateniese and B. de Medeiros, "On the key exposure problem in chameleon hashes," *In Proceedings of SCN 2004*, LNCS 3352, pp. 165–179, Springer-Verlag 2004.

[28] X. Chen, F. Zhang, H. Tian, B. Wei and K. Kim, "Discrete logarithm based chameleon hashing and signatures without key exposure," *Computers and Electrical Engineering*, Vol. 37, pp. 614-623, Elsevier 2011.

[29] X. Chen, F. Zhang, H. Tian, B. Wei, W. Susilo, Y. Mu, H. Lee and K. Kim, "Efficient generic on-line/off-line (threshold) signatures without key exposure," *Information Sciences*, Vol. 178, pp.4192-4203, Elsevier 2008.

[30] A. Boldyreva and V. Kumar, "A new pseudorandom generator from collision-resistant hash functions," *In Proceedings of CT-RSA'12*, pp. 187-202, Springer-Verlag 2012.

[31] D. Chaum and T. Pedersen, "Wallet databases with observers," *In Proceedings of Advances in Cryptology Crypto 1992*, LNCS 740, pp. 89–105, Springer-Verlag 1993.

[32] H. Krawczyk and T. Rabin, "Chameleon hashing and signatures," *In Proceedings of NDSS 2000*, p. 143–154, Internet Society 2000.

[33] G. Ateniese and B. de Medeiros, "Identity-based chameleon hash and applications," *In Proceedings of FC 2004*, LNCS 3110, p. 164–180, Springer-Verlag 2004.

[34] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," *In Proceedings of Advances in Cryptology-Crypto 2001*, LNCS 2139, p. 355–367, Springer-Verlag 2001.

[35] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," *Technical Report SRI-CSL-98-04*, SRI Computer Science Laboratory, 1998.

[36] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking," *In Proceedings of MobiSys '03*, ACM, pp.31-42, New York, NY, USA, 2003.

[37] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model," *In Proceedings of IEEE ICDCS '05*, pp. 620-629, 2005.

[38] H. Kido, Y. Yanagisawa and T. Satoh, "Protection of Location Privacy Using Dummies for Location-Based Services," *In Proceedings of ICPS '05*, 2005.

[39] M.F. Mokbel, C.Y. Chow and W.G. Aref, "The New Casper: Query Processing for Location Services Without Compromising Privacy," *In Proceedings of IVLDB '06*, 2006.

[40] M. Duckham and L. Kulik, "Dynamic & Mobile GIS: Investigating Change in Space and Time," *Location Privacy and Location-Aware Computing*, Taylor & Francis, 2006.

[41] C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati and P. Samarati, "Location Privacy Protection Through Obfuscation-Based Techniques," *In Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security*, LNCS 4602, pp. 47–60, 2007.

[42] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi and K. Tan, "Private Queries in Location Based Services: Anonymizers are not necessary," *In Proceedings of ACM SIGMOD (SIGMOD '08)*, 2008.

[43] H. Lu, C.S. Jensen and M.L. Yiu, "A3D: Anonymity area aware, Dummy-Based Location Privacy in Mobile Services," *In Proceedings of MobiDE '08*, 2008.

[44] C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati and P. Samarati, "An Obfuscation-based Approach for Protecting Location Privacy," *IEEE Transactions on Dependable and Secure Computing*, Vol.8, No.1, pp.13-27, 2011.

[45] GingerMaster, http://www.cs.ncsu.edu/faculti/jiang/GingerMaster/.

[46] DroidDream, http://blog.mylookout.com/blog/2011/03/01/security-alert-malware-found-in-official-android-market-droiddream/.

[47] Jiwire's Wi-Fi Finder. http://www.jiwire.com.

[48] Android, http://www.android.com/.

[49] EXata 2.1, http://www.scalable-networks.com/exata/.

[50] NS3 Simulator, http://www.nsnam.org/.

Summary

DDoS Attack Exploiting Location Information and its Mitigation over Wireless Networks

스마트 모바일 기기들의 폭발적인 보급에 따라 무선 데이터 통신 또한 급속도로 증가하고 있다. 해당 스마트 모바일 기기들의 무선 데이터 통신을 지원하기 위해 IEEE 802.11 (WiFi)에 기반한 액세스 포인트들을 포함한 많은 수의 기지국들이 밀집되어 설치되고 있다. 이런 무선 통신 인프라의 확장과 보급은 언제 어디서나 무선 데이터 통신을 할 수 있게 해주는 반면, 무선 네트워크를 둘러싼 서비스 거부 공격, 악성 모바일 코드 등 여러 가지 보안 문제들 또한 제기되고 있는 실정이다.

본 학위논문에서는 스마트 모바일 기기의 위치정보를 부당하게 이용함과 동시에 무선 네트워크의 능동 스캐닝 취약점을 활용한 새로운 분산 서비스 거부 공격을 보이며, 또한 해당 공격 모델의 명확한 개념 검증을 위해 시뮬레이션 결과를 제시한다. 해당 시뮬레이션 결과로 상대적으로 작은 숫자의 공격자가 매우 적은 양의 능동 스캐닝 통신량을 가지고도 무선 통신을 완전히 방해할 수 있음을 보인다. 또한 해당 공격 모델에 대응하는 완화 방법으로 카멜레온 해쉬에 기반한 위치 정보 보호 방식을 제안한다.

해당 분산 서비스 거부 공격 모델을 구성하기 위해, 무선 네트워크를 공격하려는 공격자는 사용자의 스마트 모바일 기기에 악성 코드를 전파하고, 해당 사용자 기기는 C&C 서버들의 은밀한 제어를 받는 좀비 기기가 된다. 봇넷 마스터가 공격 명령을 내리면 좀비 기기는 그들의 능동 디스커버리 공격을 활성화하게 된다. 해당 공격 모델의 평가를 위한 WiFi 모델의 경우 Probe Request가 기본 공격단위가 된다. 기존 Probe Request Flooding (PRF)와 같은 능동 디스커버리 공격과 본 연구의 공격의 결정적인 차이점은 매우 작은 양의 통신량과 요청 메시지의 합당한 요청 간격 때문에 정상적인 능동 디스커버리 프로토콜과 구별이 되지 않는다는 것이다. 더욱이 해당 공격은 특수한 재머와 같은 장비의 설치 없이도 사용자 위치 정보를 활용하여 공격자가 주요 목표로 하는 정부 기관, 은행 등 특정 지역을 목표로 한 공격이 가능하다.

해당 공격 모델의 완화를 위해 공격자가 특정 지역을 공격하는 것을 방지하도록 사용자의 위치 정보의 보호가 필요하다. 기존의 위치 정보 프라이버시 보호 연구들은 크게 위치 클로킹과 k-익명성 방식으로 나눌 수 있으며, 추가적인 제 3의 신뢰기관 TTP가 필요하다. 하지만 기존 연구들은 보안성에 대한 엄밀한 증명이 없어서 상대적으로 위치 정보가 쉽게 노출될 수 있는 가능성이 있으며, 또한 추가적인 TTP가 반드시 요구되기 때문에 사용자의 위치 정보는 해당 TTP에게 노출될 수 밖에 없는 구조이다. 따라서 본 논문에서는 사용자의 정확한 위치는 누구에게도 절대 노출되지 않는 카멜레온 해쉬에 기반한 안전하고 효율적인 위치 정보보호 방식을 제안하고, 해당 방식의 보안성과 성능을 분석한다.