



국민대학교  
소프트웨어융합대학  
소프트웨어학부

# 캡스톤 디자인 I

## 종합설계 프로젝트

프로젝트 명	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE
팀 명	04
문서 제목	결과보고서

Version	2.0
Date	2022-05-23

팀원	노용준 (팀장)
	문성찬

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23


#### CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 다학제간캡스톤디자인 수강 학생 중 프로젝트 "SADS: Spoofing Attack Detection System at Indoor Positioning using BLE"를 수행하는 팀 "04"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "04"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

## 문서 정보 / 수정 내역


<b>Filename</b>	팀04-수행결과보고서.pdf
<b>원안작성자</b>	노용준, 문성찬
<b>수정작성자</b>	노용준, 문성찬

수정 날짜	대표 수정자	Revision	추가/수정 항목	내 용
2022-05-18	노용준	1.0	최초 작성	보고서 내용 초안 작성
2022-05-20	문성찬	1.1	내용 추가	수식 및 Preliminary 부분 내용 추가
2022-05-21	노용준	1.2	내용 추가	성능 평가 부분 내용 추가
2022-05-21	문성찬	1.3	내용 수정	Use Case 부분 내용 추가 및 검토
2022-05-22	노용준	1.4	내용 수정	결과 및 기대효과 내용 추가
2022-05-23	문성찬	2.0	최종 작성	성능평가 내용 추가

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

## 목 차

1	Introduction.....	4
1.1	프로젝트 개요 및 필요성.....	4
1.2	프로젝트 목표.....	5
2	Related Work.....	5
2.1	선행 연구 소개 및 한계점.....	5
3	Preliminary.....	7
3.1	활용/개발된 기술.....	7
4	Problem Formulation.....	11
4.1	시스템 구조.....	11
4.2	시스템 개요.....	12
4.3	실험 가정.....	12
5	SADS (Spoofing Attack Detection System).....	13
5.1	연구/개발 내용 및 시스템 기능.....	13
6	Performance Evaluation.....	20
6.1	성능 평가.....	20
7	Limitation.....	25
7.1	현실적 제한 요소.....	25
7.2	해결 방안.....	25
8	Use Case.....	25
8.1	활용방안.....	25
8.2	시스템 확장성.....	27
9	Result.....	27
9.1	결과.....	27
9.2	기대효과.....	27
9.3	자기 평가.....	28
10	Reference.....	30
10.1	참고문헌.....	30
11	부록.....	31
11.1	실험 환경.....	31
11.2	사전 설정.....	32
11.3	실행 매뉴얼.....	33

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23


# 1 Introduction

## 1.1 프로젝트 개요 및 필요성

스마트 도시의 발전에 따라 실내 위치 측위 시스템의 설계는 생활에 포함된 영역에서 다양한 용도로 사용될 수 있습니다 [2][9]. 오늘날 가장 간단한 위치 확인 시스템인 위성위치확인시스템 (GPS)는 가시선 (LoS)에 의존되어 위치 기반 서비스가 작동되어 실내 환경에서 사용할 수 없습니다. 또한 적절한 실내 위치 측위 시스템의 정확도 오차는 1m인 반면, GPS는 최대 5m의 오차가 존재하여 실외에 비해 공간이 비좁은 실내에서 사용하는데 한계가 있습니다 [10]. 실내 위치 측위를 수행하는 일반적인 무선 기술은 와이파이, 블루투스, VLC, RFID, UWB가 있습니다 [11]. 이중 와이파이는 실내 위치 측위를 수행하기 위해 최소한의 하드웨어가 사용되어 가장 간단한 방법일 수 있습니다. 하지만 실내 위치 측위 시스템은 대상을 추적하기 위해 지속적으로 업데이트되는 특징을 가집니다. 이러한 특징은 Wifi에서 많은 양의 전력을 야기하여 빠른 배터리 고갈의 원인이 될 수 있습니다. 따라서 Wifi는 대부분의 실내 위치 측위 시스템에 이상적이지 않을 수 있습니다 [12]. 최근 사물 인터넷의 발전에 따라 비용이 저렴하고 에너지 효율적인 장치들이 많이 개발되었습니다 [12]. 대표적으로 BLE는 대부분 배터리가 필요하다는 단점이 있지만, 배터리의 비용이 저렴하고 낮은 에너지 소비로 오래 사용할 수 있어 [2], 실내 위치 측위 시스템에서 최적의 솔루션으로 채택되었습니다 [13].

Bluetooth 4.0 이전 버전인 Bluetooth Classic은 무선통신 시 과도한 전력을 소비하여 빠른 배터리 소모를 일으키는 큰 단점이 존재했습니다. 이러한 단점을 보완한 Bluetooth Low Energy (BLE)는 Bluetooth Classic에 비해 매우 적은 전력으로 무선 통신이 가능하며, 현재 웨어러블 및 Internet-of-Things (IoT) 기기의 통신 프로토콜로 많이 채택되고 있습니다. BLE 장치는 광고모드와 연결모드로 구분되는데, 광고모드에서 BLE 장치(Peripheral)는 사용자 (Central)와의 연결을 위해 광고 패킷 (비콘 메시지)를 주기적으로 방송합니다 [14]. 사용자 또한 BLE 장치의 광고 패킷을 수신하기 위해 지속적으로 주변의 신호를 스캔합니다. 스캔에 성공하면 BLE 장치와 정상적으로 연결됩니다. 이때 광고모드에서 사용되는 광고 패킷은 수신되는 신호의 강도를 의미하는 Received Signal Strength Indicator (RSSI)를 포함하고 있습니다.

RSSI는 광고 패킷을 방송하는 장치 (송신기)와 수신하는 장치 (수신기)의 거리가 가까울 수록 높은 값을 보이며, 반대로 거리가 멀수록 낮은 값을 보입니다. 이러한 특징으로 RSSI는 삼변측량 기법[1]이나 Particle Filtering-Based Indoor Positioning System (PF-IPS)[2] 등 실내 위치 측위를 위한 방법으로 활발히 연구되고 있습니다. 한편, 유저의 송신기가 방송하고 있는 광고 패킷은 공개되어 있어 비콘 신호 스캔 앱이나 Bluez[3]와 같은 Bluetooth 관련 라이브러리를 사용하여 주변에 방송되고 있는 광고 패킷의 정보를 누구나 쉽게 확인할 수 있습니다. 광고 패킷에는 BLE 장치의 타입, 제조사 등의 정보뿐만 아니라 식별자로 사용되는 Mac Address나 Universally Unique Identifier (UUID)와 같은 정보도 포함되어 있습니다 [4]. 따라서 공격자는 큰 어려움 없이 사용자

 국민대학교 소프트웨어학부 다학제간캡스톤디자인	결과보고서		
	프로젝트 명	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	팀 명	04	
	Confidential Restricted	Version 2.0	2022-05-23

BLE 기기의 식별자를 얻을 수 있어 손쉽게 스푸핑 공격을 수행할 수 있습니다.

## 1.2 프로젝트 목표

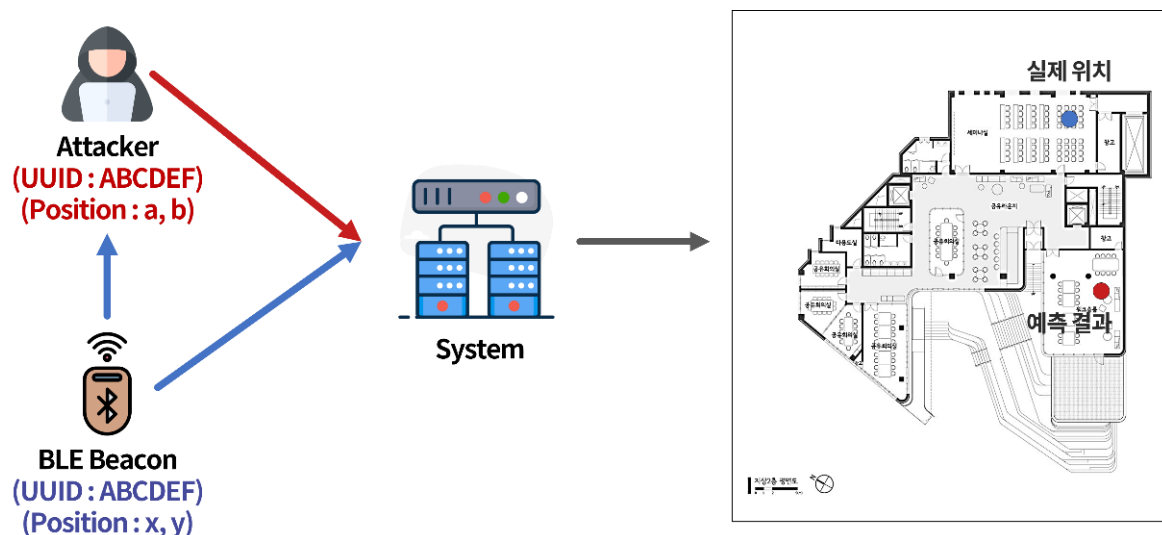


그림 1. 실내 위치 측위에서의 스푸핑 공격 시나리오

대표적인 스푸핑 공격은 BLE 기기의 식별자를 찾아내 공격자가 실제 사용자인 것처럼 식별자를 바꾸어 잘못된 데이터를 전송하는 Twin Evil Attack 방법입니다. 예를 들어, 그림 1은 실내 위치 측위 실험 환경에서 발생할 수 있는 Twin Evil Attack 시나리오입니다. 시스템은 BLE Beacon에서 방송되는 광고 패킷의 RSSI를 수신합니다. 이때 공격자가 BLE Beacon과 다른 위치에서 식별자를 속여 광고 패킷을 System에게 송신하면, System은 정상적으로 BLE Beacon의 위치를 예측할 수 없습니다. 이와 같은 스푸핑 공격은 RSSI를 활용한 실내 위치 측위 정확도에 매우 큰 영향을 줄 수 있습니다 [15]. 따라서 본 프로젝트는 이러한 유형의 스푸핑 공격을 감지하고, 공격자의 정보를 수집할 수 있는 시스템을 제작하는 것이 목표입니다. 다음 섹션에서는 스푸핑 공격을 감지할 수 있는 기존의 연구들과 한계점을 소개합니다.

## 2 Related Work

### 2.1 선행 연구 소개 및 한계점

#### A. The properties of beacons

스푸핑 공격을 예방하는 가장 간단한 방법은 광고 패킷의 속성들을 변경하는 것입니다. 광고 패킷의 UUID (16바이트)와 MAC Address (4바이트)는 값을 바꾸어 다른 사용자와 겹치지 않는 고유 식별자로 사용될 수 있습니다. iBeacon의 경우 2바이트의 Major와 Minor의 조합으로 동일한 회사나 UUID에 속하는 비콘을 더 세밀하게 구분할 수 있습니다 [16]. 하지만 공격자 또한 광고

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

패킷의 속성을 변경할 수 있고, 스마트폰이나 노트북을 통해 주변 광고 패킷들을 스캔하여 쉽게 속성 값을 확인할 수 있습니다 [17]. 따라서 비콘 메시지의 속성을 변경하는 것은 스푸핑 공격을 예방할 수 없고, BLE 기기의 보안을 강화하는데 적합하지 않습니다.

## B. Time Interval

시계열 데이터에 속하는 광고 패킷의 시간 간격은 현재 수신 받은 광고 패킷의 도착 시간과 이전에 수신 받은 광고 패킷의 도착 시간의 차이로 계산할 수 있습니다. 지속적으로 수신되는 광고 패킷에 의해 끊임없이 시간 간격이 계산됩니다. 이렇게 저장된 여러 개의 시간 간격은 약간의 규칙성을 형성하게 됩니다. 만약, 공격자가 우리의 송신기에 광고 패킷을 송신하면 기존의 규칙성을 띄고 있던 시간 간격에 비해 매우 짧은 시간 간격이 계산됩니다. 이처럼 규칙성에 벗어난 이상치가 계산되었을 때 스푸핑 공격임을 감지할 수 있습니다.

한편, 광고 패킷이 수신되는 시간은 매우 불규칙적이어서 공격자의 간섭이 없어도 시간 간격 이상치의 발생 빈도는 매우 높습니다. 따라서 시계열 데이터의 대표적인 예측 방법인 ARIMA 방법론이나 LSTM을 활용하여 다음 시간 간격 값을 예측하거나 이상치를 구분할 수 있는 범위를 계산하는 연구가 진행되고 있습니다 [5][6]. 이러한 기법을 활용한 스푸핑 공격 감지는 매우 효과적일 수도 있습니다. 하지만 시간 간격을 사용한 방법은 스푸핑 공격을 감지만 할 수 있다는 한계점이 있으며, 단지 스푸핑 공격을 감지만 하는 것은 궁극적인 스푸핑 공격 예방의 해결책이 될 수 없습니다. 스푸핑 공격을 제대로 예방하기 위해서는 공격자의 정보나 광고 패킷의 속성 등을 알아야 합니다. 하지만 불규칙적으로 수신되는 광고 패킷 중 유저 BLE 장치의 광고 패킷과 공격자의 광고 패킷을 구분하는 것은 매우 어렵습니다.

## C. Received Signal Strength Indicator (RSSI)

RSSI는 수신기와 송신기의 거리를 대략적으로 나타낼 수 있는 지표로 사용되며 식은 아래와 같습니다 [18].

$$RSSI = -10n \log_{10} d + A$$

위 식에서  $n$ 은 경로 손실 지수이고  $A$ 는 수신기에서 측정된 기준 RSSI 값입니다. RSSI는 스푸핑 공격 예방에 사용되는 대표적인 속성이며 단순히 스푸핑 공격을 감지하는 것이 아닌, 유저의 송신기와 공격자의 거리를 비교하여 공격자의 광고 패킷도 특정할 수 있습니다. 하지만 공격자와 유저의 송신기 사이의 거리가 가깝거나 같은 위치에 있으면 RSSI 값도 비슷해 악의적인 의도를 가진 광고 패킷을 특정할 수 없는 치명적인 단점이 있습니다. 여러 RSSI를 활용한 스푸핑 공격 감지 연구에서 이러한 단점을 찾아볼 수 있으며 [19][20], 현재 이와 같은 단점을 극복하기 위한 여러 연구가 진행되고 있습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

성공적인 스푸핑 감지 시스템 중 하나인 BlueShield[7]는 3개의 수신기를 사용하여 광고 패킷이 방송되는 3개의 채널(37, 38, 39)의 신호를 수신 받습니다. 따라서 공격자는 각각 다른 위치에 있는 수신기들의 RSSI를 모두 속일 수 없습니다. 또한 수신기들은 수신 받을 3개의 채널을 중복되지 않게 지속적으로 변경하므로 공격자는 특정 수신기가 수신 받는 채널을 예상하기 쉽지 않습니다. 이렇게 3개의 수신기를 사용한 스푸핑 공격 탐지는 일반적으로 사용되는 방식에 비해 공격자가 식별될 수 있는 유효 거리에 대한 한계점을 꽤 극복하였습니다. 하지만 여전히 공격자와 유저의 송신기 간의 거리가 매우 가깝거나 동일한 위치에 있을 경우 스푸핑 공격을 잘 탐지하지 못하는 단점이 존재합니다. 본 프로젝트에서는 이러한 한계점에 집중하여 SADS를 개발하였고, 문제의 공식화는 다음과 같습니다.

## 3 Preliminary

### 3.1 활용/개발된 기술

#### A. 타원 곡선 암호화 (Elliptic Curve Cryptography, ECC)

ECC는 RSA 방식과 동일한 기능을 제공하는 타원 곡선 기반의 공개키 암호 방식입니다 [23]. 같은 공개키 암호 방식인 RSA와 비교했을 때 ECC가 더 작은 bit 수의 암호키로 RSA 암호와 동일한 암호 성능을 가집니다 [8]. 본 프로젝트에서는 무작위의 양수 배열을 생성하기 위한 랜덤 시드를 암호화 그리고 복호화하는 방법으로 ECC를 사용합니다. ECC를 이해하기 위해선 타원 곡선 상의 연산과 타원 곡선 상의 유한체 (finite field)를 이해해야 하며 타원 곡선의 방정식은 아래와 같습니다.

$$y^2 = x^3 + ax + b$$

위 식의 형태를 가지는 타원 곡선은  $x$ 축을 중심으로 대칭되고, 비 수직선에 대해 최대 3개 지점에서 곡선과 교차될 수 있습니다. 이러한 특징으로 타원 곡선의 덧셈 연산은 타원 곡선 상의 두 점  $A(x_1, y_1)$ 와  $B(x_2, y_2)$ 를 지나는 직선이 타원 곡선과 만나는 또 다른 교점을  $x$ 축으로 대칭시킨 점  $C(x_3, y_3)$ 를 구하는 것으로 정의하며 식은 아래와 같습니다.

$$\begin{aligned}
 (1) \text{ Addition : } \lambda &= \frac{y_2 - y_1}{x_2 - x_1} & (2) \text{ Doubling : } \lambda &= \frac{3x_1^2 + a}{2y_1} \\
 x_3 &= \lambda^2 - x_1 - x_2 & x_3 &= \lambda^2 - 2x_1 \\
 y_3 &= (x_1 - x_3)\lambda - y_1 & y_3 &= (x_1 - x_3)\lambda - y_1
 \end{aligned}$$

이때 덧셈연산은  $A$ 와  $B$ 가 다른 점에 존재할 때와 같은 점에 존재할 때 2가지의 경우로 나뉘며, 각각 Addition과 Doubling 연산으로 계산됩니다. 다음으로 타원 곡선에서의 곱셈 연산 식은 아래와 같습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

$$k \times P = P + P + \dots + P$$

위 식에서  $k$ 는 곱하는 수로 타원 곡선의 덧셈 연산을  $k$ 번 수행하는 것과 같습니다. 한편 공개 키 형식의 타원 곡선 암호화는 유한체에서 정의될 수 있으며 유한체를 정의하는 식은 아래와 같습니다.

$$y^2 \bmod p = x^3 + ax + b \bmod p$$

유한체의 식은  $0 \sim p-1$ 의 좌표 안에서 정의되며 타원 곡선 방정식의 양변에 모듈러 연산을 취한 형태입니다. 이때  $p$ 는 3보다 큰 소수이며 값이 클수록 대응되는  $y$ 의 값을 구하기 어려워집니다. 이러한 점을 이용한 ECC 메커니즘은 그림 2와 같습니다.

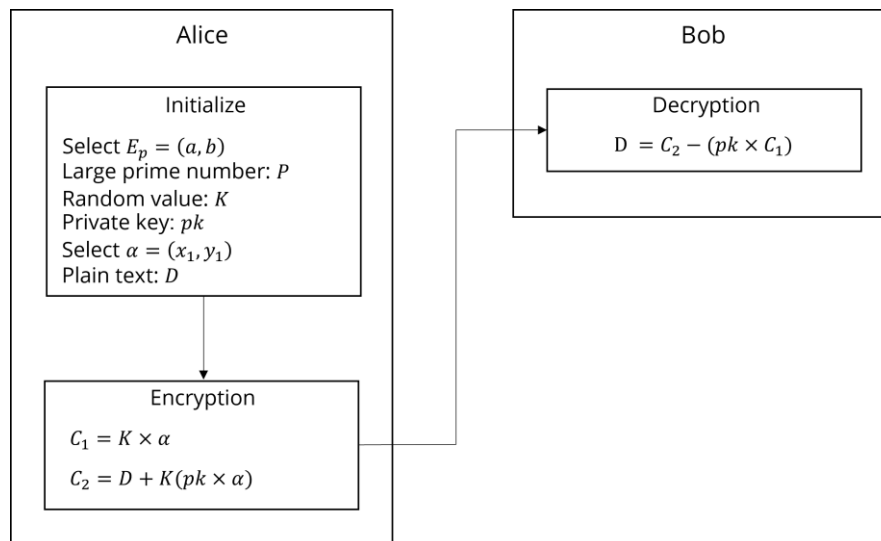


그림 2. 타원 곡선 암호화 메커니즘

데이터를 보내기 위해 암호화하는 Alice는 ECC를 사용하기 위한 값을 초기화합니다. 이때  $E_p$ 와  $P$ 는 타원 곡선 상의 유한체를 정의하기 위한 매개변수이며  $P$ 는 매우 큰 소수입니다.  $K$ 는 암호화에 사용되는 무작위 양수이고,  $pk$ 는 비밀키로 Alice와 Bob에게 알려져 있습니다.  $\alpha$ 는 공개키로 유한체 위의 한 점을 무작위로 선택합니다. 마지막으로  $D$ 는 암호화하려는 평문 (Plaintext)입니다.

값들의 초기화가 끝나면 Alice는  $D$ 를 암호화하여 Bob에게 전송합니다. 그리고 Bob은 비밀키를 사용하여 암호문을 복호화합니다. 이때 암호화와 복호화에 사용되는 계산은 위에서 언급한 타원 곡선 상의 덧셈과 곱셈 연산을 사용합니다.

만약 Alice와 Bob이 독립적으로 운영되는 프로그램이고 Alice에서 생성된 비밀키를 Bob이 알기 위해선 한 번 이상의 통신이 반드시 이루어져야 합니다. 이것을 방지하기 위한 다양한 공개키 교환 알고리즘이 존재하며, 본 프로젝트에서 사용한 Elgamal 공개키 교환 방법은 섹션 5에서 자세



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

히 설명합니다.

## B. 신뢰구간 (Confidence Interval, CI)

신뢰구간은 특정 수준의 신뢰를 가진 모집단의 값을 포함할 가능성이 있는 값의 범위입니다. 주위의 광고 패킷이 충돌하며 불안정해지는 수신 시간은 비이상적으로 높거나 낮은 수신 시간 간격을 출력할 수 있습니다. 이러한 잘못된 값들은 다음 시간 간격을 예측하는데 큰 장애물이 될 수 있습니다. 신뢰 구간은 예측에 사용될 데이터가 포함될 수 있는 구간을 제공하여 이러한 문제를 해결할 수 있으며 [24], 식은 아래와 같습니다.

$$CI = \bar{X} \pm Z_{\alpha/2} \frac{s}{\sqrt{n}}$$

위의 식에서  $\bar{X}$ 는 예측에 사용될 데이터들의 평균 즉, 표본 평균이며,  $Z$ 는 신뢰구간 안에 모집단의 평균이 존재할 확률, 즉 신뢰 수준입니다. 그리고  $n$ 은 표본의 크기,  $s$ 는 표본의 표준편차입니다. 다음은 다음 시간 간격 예측을 위한 홀트 선형 추세 기법을 소개합니다.

## C. 홀트 선형 추세 기법 (Holt Linear Trend Technique)

홀트 선형 추세 기법은 시계열 데이터의 예측 방법 중 하나로 데이터의 주기성과 추세로 다음 값을 예측하며 [25], 식은 아래와 같습니다.

$$\widehat{y_{t+1}} = l_t + b_t$$

위 식에서  $\widehat{y_{t+1}}$ 는 다음 예측 값이며,  $l_t$ 는 시간  $t$ 에서의 시계열 수준 추정 값,  $b_t$ 는 시간  $t$ 에서의 시계열 추세 (기울기) 추정 값으로 두 식의 계산식은 아래와 같습니다.

$$\begin{aligned} l_t &= \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \end{aligned}$$

이때  $y_t$ 는 시간  $t$ 에 대한 실제값이고,  $\alpha$ 와  $\beta$ 는 각각 수준과 추세에 대한 매개변수로 0과 1사이의 값을 가집니다. 즉, 홀트 선형 추세 기법은  $l_{t-1}$ 과  $b_{t-1}$ 을 사용하여 시간이  $t$ 일때의 값을 예측하는 one-step-ahead training forecast입니다 [25]. 그리고 다음 예측을 위해 사용되는 각각의 추정값들은  $t$ 일 때의 실제값으로 갱신됩니다. 한편, 예측값은 스푸핑 공격 여부를 확인하기 위해 이상치를 탐지하는 홀트 신뢰 구간을 형성하게 되는데, 이것은 위에서 소개된 신뢰구간의 식을 약간 변형한 형태로 섹션 5에서 자세히 설명될 예정입니다. 다음은 칼만 필터를 소개합니다.

## D. 칼만 필터 (Kalman Filter, KF)

칼만 필터는 가우스 노이즈가 존재하는 선형 시스템에서 신호를 처리하고 값을 예측하는 최적의 기법 중 하나입니다 [26]. RSSI는 시간 간격과 마찬가지로 매우 불규칙적인 모습을 보이는데,

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

KF를 사용하여 RSSI를 평활화하여 노이즈를 줄일 수 있습니다. KF는 잡음이 포함된 측정치를 바탕으로 다음 측정값을 추정하는 재귀 필터로 예측과정과 추정과정으로 나뉩니다. 예측과정에선 추정값과 오차 공분산을 계산합니다. 아래는 추정값을 구하는 식으로 이전 단계에서 계산된 추정값을 사용하여  $A$ 와의 계산을 통해 새로운 추정값을 예측합니다.

$$\hat{x}_{\bar{t}} = A\hat{x}_{t-1}$$

위 식에서  $\hat{x}_{t-1}$ 은 시간  $t-1$ 에서 계산한 추정값이며,  $A$ 는 추정값을 예측할 때 사용하는 행렬입니다. 아래는 예측한 값이 평균을 기준으로 어느정도 분포되어 있는지 이전 오차 공분산을 사용하여 새로운 오차 공분산을 예측하는 식입니다.

$$P_{\bar{t}} = AP_{t-1}A^T + Q$$

위 식에서  $P_{t-1}$ 는 시간  $t-1$ 에서 계산한 오차 공분산이며,  $Q$ 는 시스템의 노이즈입니다. 이렇게 예측된 추정값과 오차 공분산은 추정과정에서 사용되며 추정과정은 칼만 이득 (Kalman Gain), 추정값, 오차 공분산 계산 순서로 진행됩니다. 칼만 이득은 오차 공분산의 예측값과 측정값의 노이즈로 계산되며 식은 아래와 같습니다.

$$K_t = P_{\bar{t}}H^T(HP_{\bar{t}}H^T + R)^{-1}$$

$H^T$ 는 측정값의 형태로 변환할 때 필요한 행렬이며,  $R$ 은 측정값의 노이즈입니다. 위 식으로 계산된 칼만 이득은 새로 측정값이 입력되었을 때 추정값을 계산하기 위해 사용되며 추정값의 계산 식은 아래와 같습니다.

$$\hat{x}_t = \hat{x}_{\bar{t}} + K_t(Z_t - H\hat{x}_{\bar{t}})$$

$Z_t$ 는 측정값이며 칼만 이득  $K_t$ 는 측정값과 이전에 구했던 추정값의 가중치를 결정합니다. 즉, 추정값을 구할 때 측정값의 노이즈가 작아 칼만 이득이 크다면 측정값에 많은 가중치를 부여해 계산하고, 오차 공분산의 예측값이 작아 칼만 이득이 작다면 예측값에 더 많은 가중치를 부여해 계산합니다. 위 식을 통해 추정값  $\hat{x}_t$ 가 예측되면 새로운 오차 공분산을 계산하며 식은 아래와 같습니다.

$$P_t = (1 - K_tH)P_{\bar{t}}$$

이렇게 구해진 추정값과 새로 계산된 오차 공분산은 다음 예측 과정에서 사용되며 위와 같은 계산 과정이 재귀적으로 수행됩니다.

## 4 Problem Formulation

### 4.1 시스템 구조

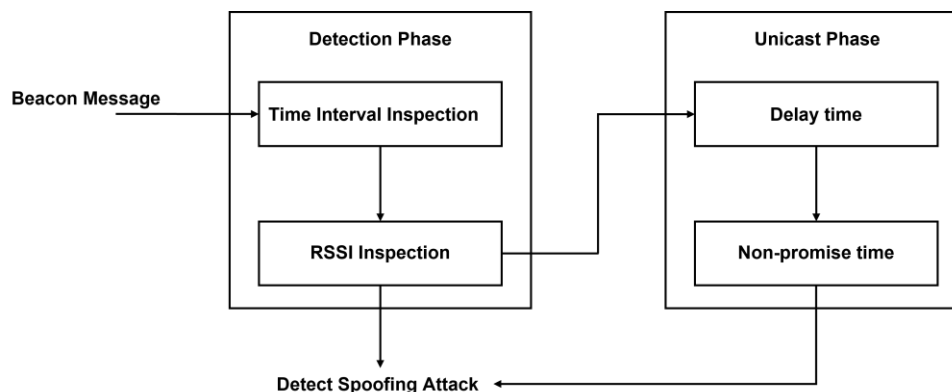


그림 3. SADS 서버 시스템 구조


본 프로젝트에서는 그림 3과 같은 스푸핑 공격 감지 시스템을 제안합니다. BLE 형식의 웨어러블 또는 IoT 기기, 즉 송신기는 주기적으로 광고 패킷 (비콘 메시지)을 방송합니다. 수신기는 송신기가 보내는 광고 패킷을 수신할 수 있습니다. 광고 패킷이 수신되면 서버는 Detection Phase에서 연속적으로 수신되는 광고 패킷의 시간 간격을 검사합니다. 이후 스푸핑 공격이 의심되는 상황이 발생되면 RSSI 검사를 진행하며 이때 발생할 수 있는 경우의 수는 아래와 같이 2가지로 구분됩니다.

- *Detect Spoofing Attack*: 유저의 송신기와 공격자의 거리가 멀어 RSSI 검사에서 공격자의 광고 패킷이 특정된 경우입니다.
- *Non – Detect Spoofing Attack*: 유저의 송신기와 공격자의 거리가 가까워 RSSI 검사에서 공격자의 광고 패킷이 특정되지 않은 경우입니다.

만약 *Non – Detect Spoofing Attack* 상황이 발생된다면 Unicast Phase를 통해 공격자의 광고 패킷을 검출할 수 있으며 2단계로 나누어 실행됩니다.

- *Delay Time*: 서버가 *Non – Promise Time*으로 바뀌는 타이밍을 모르게 하기 위해 설정하는 무작위 시간입니다. 유저의 송신기는 광고 패킷을 송신하지만 수신기는 광고 패킷을 수신하지 않습니다.
- *Non – Promise Time*: 유저의 송신기와 수신기가 광고 패킷을 송수신하지 않겠다고 약속한 시간으로 수신기는 공격자의 광고 패킷만을 수신할 수 있습니다.

위와 같은 순서로 스푸핑 공격을 탐지하며 스푸핑 공격이 감지되지 않았다면 현재 수신된 광고 패킷을 기반으로 다음 스푸핑 공격의 탐지를 위해 각각의 검사 모델을 재귀적으로 갱신합니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

## 4.2 시스템 개요

그림 3과 같이 제안된 스푸핑 공격 시스템은 다음과 같은 순서로 작동합니다.

- 일대일 통신 (Unicast Communication)을 위해 양수로 이루어진 배열을 무작위로 생성하는 랜덤 시드를 암호화해야 합니다. 따라서 유저의 송신기 측에서 타원 곡선 암호를 이용해 공개키와 비밀키를 생성합니다.
- 서버는 랜덤 시드를 설정하고 양수를 무작위로 추출해 배열을 생성합니다. 이후 생성된 공개키를 사용해 랜덤 시드를 암호화하고, 암호화된 랜덤 시드를 유저의 송신기에 전송합니다.
- 유저의 송신기는 암호화된 랜덤 시드를 비밀키로 복호화 한 후, 랜덤 시드를 사용하여 양수를 무작위로 추출해 배열을 생성합니다. 이때 유저의 송신기와 서버는 같은 랜덤 시드를 사용하므로 동일한 배열은 가집니다.
- 생성된 배열을 사용하여 *Delay Time*과 *Non – Promise Time*을 정의하고 광고 패킷의 송수신을 시작합니다.
- 서버는 광고 패킷이 수신되는 시간 간격을 계산하고, 다음 시간 간격 예측에 사용되는 시간 간격 데이터를 수집합니다.
- 한편, 광고 패킷의 RSSI는 Kalman Filter (KF)에 의해 평활화되며, 이상치를 구분하는 범위를 계산하기 위해 수집됩니다.
- 예측을 위한 시간 간격 데이터와 평활화된 RSSI가 충분히 수집되었으면 광고 패킷의 다음 시간 간격을 예측합니다. 그리고 시간 간격과 평활화된 RSSI 정보의 이상치 범위를 계산합니다.
- 현재 수신된 광고 패킷의 시간 간격이 예측된 시간 간격보다 비정상적으로 낮다면 스푸핑 공격을 의심하며 RSSI 검사를 진행합니다.
- RSSI 검사를 통해 이상치를 가진 광고 패킷이 검출되면 공격자의 광고 패킷으로 특정합니다.
- RSSI 검사에서 이상치를 가진 광고 패킷이 검출되지 않는다면 Unicast Phase가 실행되고 *Delay Time*과 *Non – Promise Time*을 이용해 공격자의 광고 패킷을 특정합니다.

## 4.3 실험 가정

SADS는 다음과 같은 가정을 가집니다.

- 스푸핑 공격자는 1명으로 가정합니다.
- Detection Phase에서는 스푸핑 공격이 발생하기 전, 정상적인 광고 패킷이 10개 이상 수신되었다고 가정합니다.

다음 섹션에서는 SADS에 대해 자세히 소개합니다.

## 5 SADS (Spoofing Attack Detection System)

### 5.1 연구/개발 내용 및 시스템 기능

#### A. Time Interval Inspection

공격자의 광고 패킷이 수신됐을 때 광고 패킷의 거의 모든 정보는 유저의 송신기의 광고 패킷과 비슷합니다. 따라서 각각의 정보를 비교하는 방법은 모든 상황에서 유효하지 않을 수 있습니다. 하지만 광고 패킷이 수신된 시간 간격을 사용하면 스푸핑 공격 여부를 확인할 수 있으며, 본 프로젝트에서 정의한 광고 패킷의 시간 간격  $INT$ 는 아래와 같습니다.

$$INT = P_t - P_{t-1}$$

$P_t$ 는 시간  $t$ 에 수신된 광고 패킷의 시간이고  $P_{t-1}$ 는 시간  $t-1$ 에 수신된 광고 패킷의 시간입니다. 일반적으로  $INT$ 는 광고 패킷이 수신되는 시간이 불규칙적이므로 예측하기 매우 어렵습니다.  $INT$ 는 가끔 너무 높은 값을 출력하거나 또는 낮은 값을 출력할 수 있는데 이러한 값들은 다음  $INT$ 를 정확하게 예측하는데 큰 방해가 됩니다. 따라서  $INT$ 를 활용한 스푸핑 공격 감지는 예측에 사용할 데이터를 수집하는 단계와 다음  $INT$ 를 예측하는 단계로 나뉘며 데이터 수집 단계의 메커니즘은 그림 4과 같습니다.

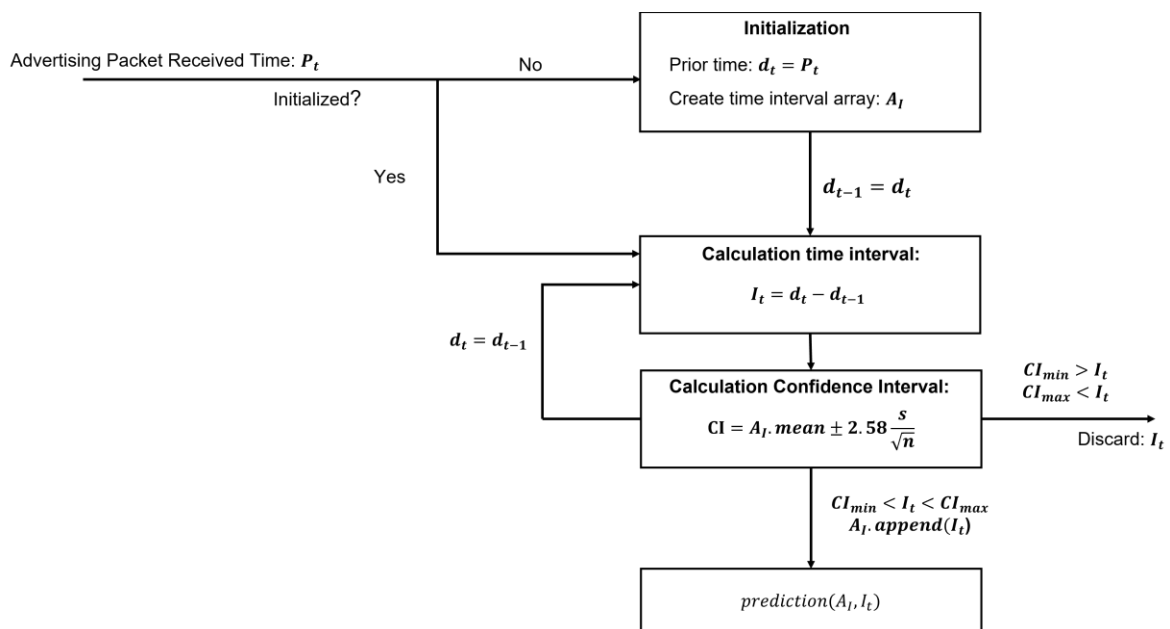


그림 4. 데이터 수집 단계 메커니즘

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

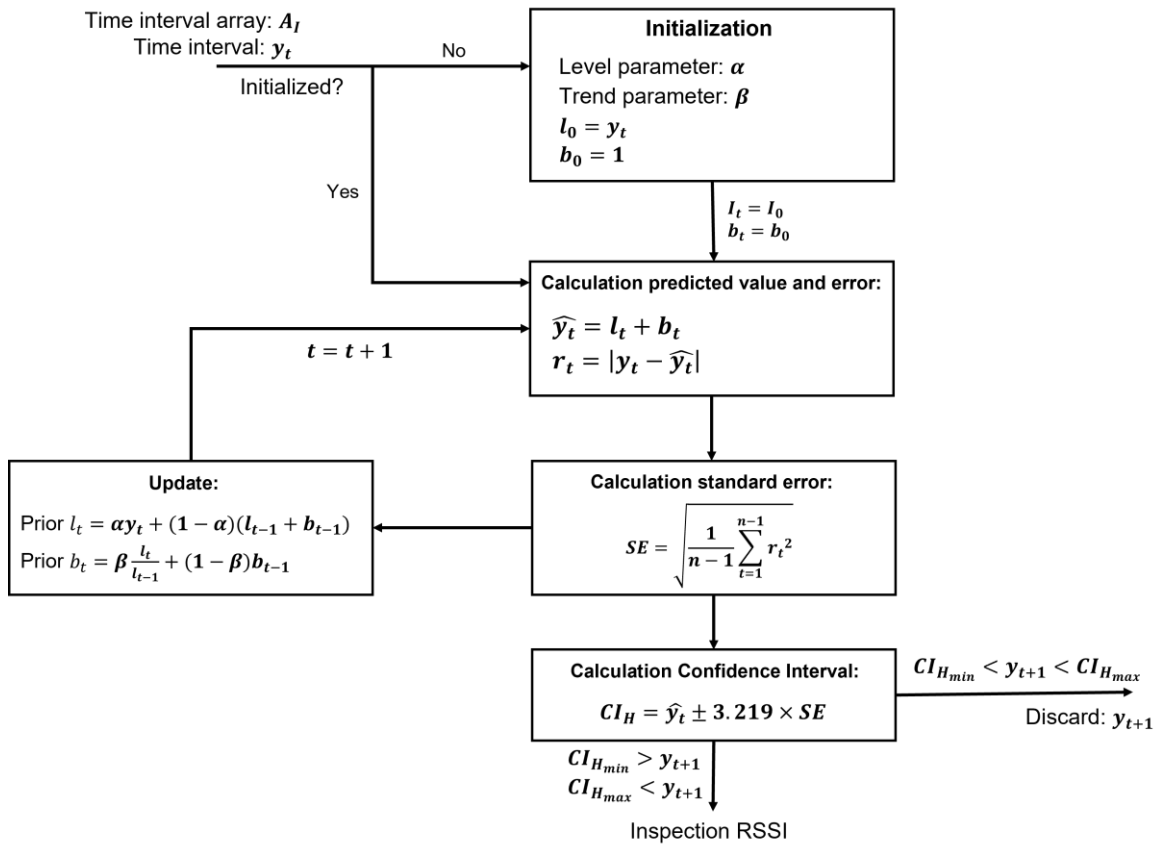


그림 5. 스푸핑 공격 검사 단계 메커니즘

광고 패킷이 수신되었을 때 시간 간격 (INT)  $I_t$ 를 계산하고, 이전에 수집된 INT 데이터가 3개 이상이라면 99%의 신뢰 수준을 적용시킨 신뢰 구간을 생성하며 식은 아래와 같습니다.

$$CI = A_I.mean \pm 2.58 \frac{s}{\sqrt{n}}$$

$s$ 는  $A_I$ 의 표준편차이고,  $n$ 은  $A_I$ 의 길이입니다. 위 식에서 계산된 신뢰구간으로 비정상적인 INT 값들을 필터링합니다. 신뢰구간에 속한 데이터가 최소 10개 이상 수집되었을 때 예측 단계가 수행됩니다. 예측 단계의 메커니즘은 그림 5과 같으며 예측식은 아래와 같습니다.

$$\hat{y}_t = l_t + b_t$$

식에서  $l_t$ 는 시간  $t$ 에서의 시계열 수준 추정값이고,  $b_t$ 는 시간  $t$ 에서의 시계열 추세 (기울기) 추정값으로 예측값의 계산이 끝나면 아래와 같이 갱신됩니다.

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

위 식에서  $\alpha$ 는 수준에 대한 매개변수이고  $\beta$ 는 추세에 대한 매개변수로 각각 0과 1사이의 값을 가집니다. 위와 같은 예측 과정이 재귀적으로 수행되며 예측값의 정밀도를 높입니다. 한편 예측된  $INT$ 와 실제  $INT$  값은 표준 오차 (Standard Error)의 계산에 사용될 수 있으며 식은 아래와 같습니다.

$$SE = \sqrt{\frac{1}{n-1} \sum_{t=1}^{n-1} r_t^2}$$

$n$ 은 값 예측에 사용된 데이터의 개수이고,  $r_t$ 는 시간  $t$ 에 대한 실제  $INT$ 와 예측된  $INT$ 의 차이입니다. 이렇게 계산되고 끊임없이 누적되는 표준 오차는 홀트 신뢰 구간을 생성하며 식은 아래와 같습니다.

$$CI_H = \hat{y}_t \pm 3.219 \times SE$$

위 식은 신뢰구간을 구할 때 표준 오차를 사용하며 실제  $INT$ 의 이상치를 탐지할 수 있는 범위를 계산합니다. 만약 현재 수신된 광고 패킷의  $INT$ 가  $CI_H$ 에 속해 있다면 스푸핑 공격이 아닙니다. 하지만  $CI_H$ 를 벗어난 값이 탐지되었을 때 공격자의 것으로 의심될 수 있는 광고 패킷은 총 두 개로 현재 수신된 광고 패킷과 직전에 수신된 광고 패킷입니다. 이러한 두 개의 광고 패킷은 총 세 가지 경우의 수를 가집니다. 첫 번째는 둘 다 유저의 송신기가 송신한 광고 패킷인 경우, 두 번째는 둘 다 공격자의 광고 패킷인 경우, 마지막은 유저의 송신기가 송신한 광고 패킷이고 다른 한 개는 공격자의 광고 패킷인 경우입니다. 따라서 유저의 송신기가 송신한 광고 패킷과 공격자의 광고 패킷을 구분하기 위해 다음과 같이 RSSI를 활용하여 공격자의 광고 패킷을 탐지하는 감지 모델을 소개합니다.

## B. RSSI Inspection

광고 패킷의 RSSI는  $INT$ 와 마찬가지로 매우 불규칙적입니다. 때문에 RSSI 검사 모델 또한 이상치 판단 범위를 정밀하게 계산하기 위한 전처리 과정과 스푸핑 공격 감지 과정으로 나누어져 있으며 전처리 과정은 아래 그림 6과 같습니다.

 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>다학제간캡스톤디자인</b> </div>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

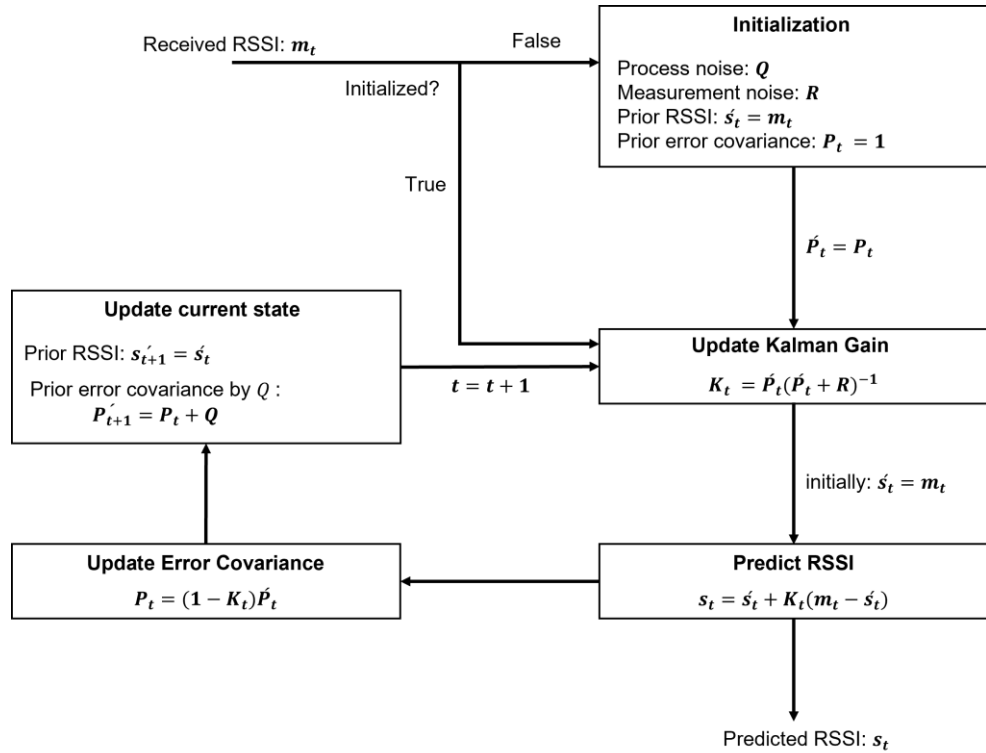


그림 6. RSSI 전처리 단계 메커니즘

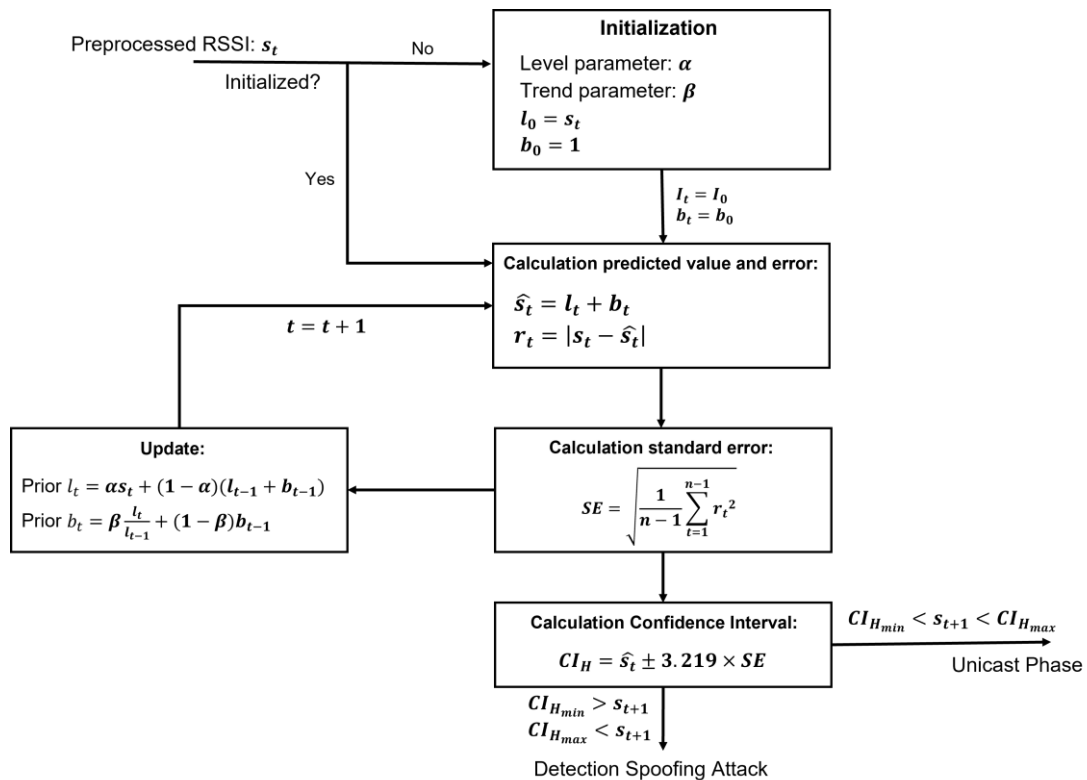


그림 7. 스푸핑 공격 감지 단계 메커니즘



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

전처리 과정은 추정 단계와 갱신 단계로 구분됩니다. 그림 6에서 예측 단계인 다음 RSSI 값을 추정하는 식은 아래와 같습니다.

$$s_t = \hat{s}_t + K_t(m_t - \hat{s}_t)$$

위 식에서  $\hat{s}_t$ 는 현재 수신된 RSSI이고,  $K_t$ 는 Kalman Gain으로 계산식은 아래와 같습니다.

$$K_t = \hat{P}_t(\hat{P}_t + R)^{-1}$$

위 식에서  $R$ 은 측정값  $s_t$ 의 노이즈이며,  $\hat{P}_t$ 는 오차 공분산으로 구하는 식은 아래와 같습니다.

$$P_{\bar{t}} = P_{t-1} + Q$$

위 식에서  $Q$ 는 시스템의 노이즈입니다. 이렇게 추정값이 계산되면 갱신 단계가 실행되며 오차 공분산을 새로 계산합니다. 오차 공분산의 갱신식은 아래와 같습니다.

$$P_t = (1 - K_t)\hat{P}_t$$

이렇게 업데이트 된 오차 공분산은 다음 값을 예측하는데 쓰이며, 전처리 과정에서는 위와 같은 단계가 재귀적으로 수행됩니다. 한편 추정값인 평활화된 RSSI는 탐지 과정에서 공격자의 광고 패킷을 구분하기 위해 사용될 수 있습니다. 자세한 메커니즘은 그림 7과 같으며 시간 간격 검사와 같이 홀트 선형 추세 기법과 신뢰 구간을 사용합니다. 예측식은 아래와 같습니다.

$$\hat{s}_t = l_t + b_t$$

식에서  $l_t$ 는 시간  $t$ 에서의 시계열 수준 추정값이고,  $b_t$ 는 시간  $t$ 에서의 시계열 추세 (기울기) 추정값으로 예측값의 계산이 끝나면 아래와 같이 갱신됩니다.

$$l_t = \alpha s_t + (1 - \alpha)(l_{t-1}b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

위 식에서  $\alpha$ 는 수준에 대한 매개변수이고  $\beta$ 는 추세에 대한 매개변수로 각각 0과 1사이의 값을 가집니다. 위와 같은 예측 과정이 재귀적으로 수행되며 예측값의 정밀도를 높입니다. 한편 예측된 RSSI와 실제 RSSI 값은 표준 오차 (Standard Error)의 계산에 사용될 수 있으며 식은 아래와 같습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

$$SE = \sqrt{\frac{1}{n-1} \sum_{t=1}^{n-1} r_t^2}$$

$n$ 은 값 예측에 사용된 데이터의 개수이고,  $r_t$ 는 시간  $t$ 에 대한 실제 RSSI와 예측된 RSSI의 차이입니다. 이렇게 계산되고 끊임없이 누적되는 표준 오차는 홀트 신뢰 구간을 생성하며 식은 아래와 같습니다.

$$CI_H = \hat{s}_t \pm 3.219 \times SE$$

위 식은 신뢰구간을 구할 때 표준 오차를 사용하며 실제 RSSI의 이상치를 탐지할 수 있는 범위를 계산합니다. 만약 현재 수신된 광고 패킷의 RSSI가  $CI_H$ 를 벗어난 값이면 해당 광고 패킷을 스푸핑 공격으로 특정합니다. 하지만 현재 수신된 광고 패킷의 RSSI가  $CI_H$ 에 속해 있다면 두 가지 경우의 수를 가집니다. 첫 번째는 불안정한  $INT$ 의 수신으로 인해 유저의 송신기만 광고 패킷을 송신함에도 불구하고 해당 광고 패킷을 스푸핑 공격으로 의심하는 경우입니다. 두 번째는 유저의 송신기와 공격자의 거리가 가까울 때 공격자의 광고 패킷을 수신한 경우입니다. 따라서 다음과 같은 두 가지 경우의 수를 확인하기 위해 Unicast Phase가 실행됩니다.

### C. Unicast Phase

본 프로젝트의 시스템은 사전에 정의된 시간에 정확히 광고 패킷의 송수신을 멈추기 위한 *Non - Promise Time*을 정의합니다. 이때 서버는 모든 경우에서 동일한 무작위 양수를 생성하기 위한 랜덤 시드를 생성합니다. 만약 공격자가 랜덤 시드를 알게 된다면 공격자 또한 사전에 정의된 시간을 알 수 있습니다. 따라서 수신기는 랜덤 시드를 생성한 후 암호화하여 유저의 송신기에 전송해야 합니다. 이때 공격자에게 노출되는 것을 최소화하기 위하여 ElGamal 공개키 교환 방법 [22]을 사용합니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

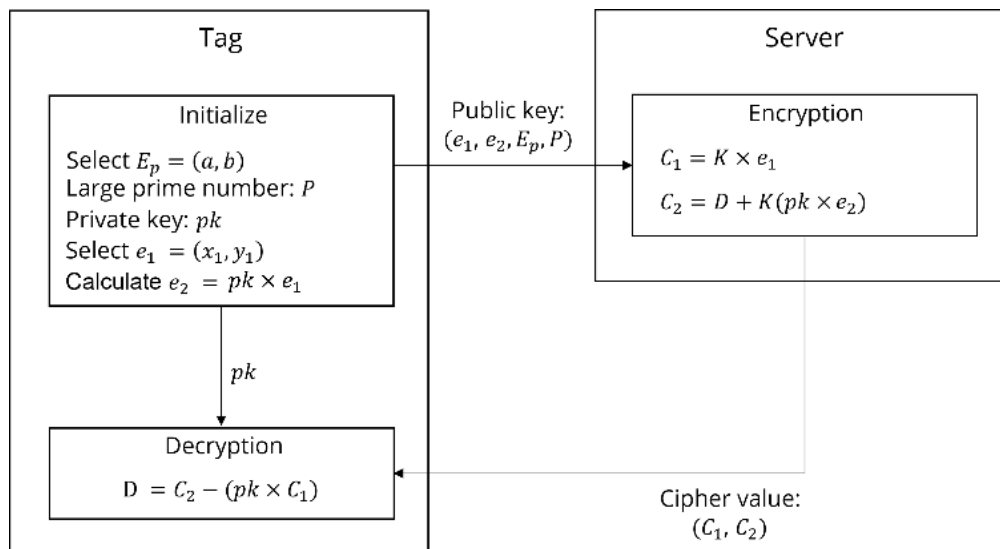


그림 8. ElGamal 공개키 메커니즘

그림 8의 공개키 생성 과정에서  $E_p$ 와  $P$ 는 타원 곡선 상의 유한체를 정의하기 위한 매개변수이며,  $P$ 는 유한체의 범위를 지정하는 매우 큰 소수입니다. 또한  $e_1$ 은 암호화에 사용되는 유한체 위의 한 점이고  $pk$ 는 비밀키이며,  $e_2$ 는  $e_1$ 을  $pk$ 번 덧셈 연산한 값입니다.

생성된 공개키  $(e_1, e_2, E_p, P)$ 는 서버로 전송되고 서버는 무작위로 추출한 랜덤 시드  $D$ 를 아래 식과 같이 암호화합니다.

$$\begin{aligned} C_1 &= K \times e_1 \\ C_2 &= x + K(pk \times e_2) \end{aligned}$$

$C_1$ 와  $C_2$ 는 암호화된 두 점이고  $K$ 는 암호화에 사용되는 무작위 양수입니다. 위 식으로 암호화된 랜덤 시드는 다시 유저의 송신기에 전송됩니다. 그리고 유저의 송신기는 사전에 정의된 비밀키  $pk$ 를 이용하여 복호화를 진행하며 식은 아래와 같습니다.

$$D = C_2 - (pk \times C_1)$$

위의 식으로 복호화된 랜덤 시드를 사용하여 무작위로 추출된 양수의 배열을 생성합니다. 생성된 배열은 아래 그림 9와 같은 메커니즘으로 사용됩니다.

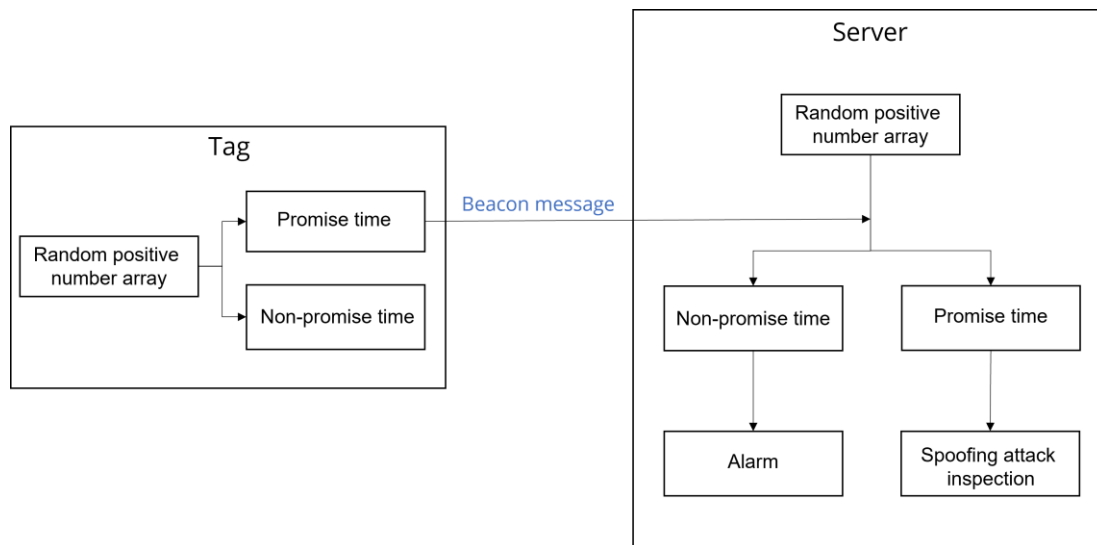


그림 9. Unicast Phase 메커니즘

그림 9에서 사용되는 무작위 양수 배열의 홀수 번째 인덱스는 유저의 송신기가 광고 패킷을 송신하지만 수신기는 광고 패킷을 수신하지 않는 *Delay Time*을 의미하고, 짝수 번째 인덱스는 광고 패킷을 송수신하지 않는 시간, 즉 *Non – Promise Time*을 의미합니다. 공격자는 무작위 양수로 생성된 *Delay time*에 의해 *Non – Promise Time*이 시작되는 시간을 쉽게 예상할 수 없습니다. 따라서 *Non – Promise Time*에 유저와 동일한 식별자로 수신되는 광고 패킷을 공격자의 스푸핑 광고 패킷으로 특정할 수 있습니다.

## 6 Performance Evaluation

### 6.1 성능 평가


#### A. Data

##### a. Stationary Data

송신기가 움직이지 않는 상황에서의 성능 평가에 사용되는 데이터입니다.

Average INT	Fast (0.03s)	Normal (0.13s)	Slow (0.35s)
Distance	0m	1m	2m
Number of Data	50	100	500

그림 10. 스푸핑 공격이 발생할 수 있는 환경

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

SADS의 스푸핑 공격 감지 정확도를 검증하기 위해 그림 10과 같은 9가지의 상황을 가정합니다. 각각의 가정은 실제 BLE Beacon을 활용한 실내 위치 측위에서 현실적으로 발생할 수 있으며, 스푸핑 공격 감지 정확도에 영향을 줄 수 있는 요소들입니다. 본 프로젝트는 성능 평가를 위해 패킷 속도, 유저의 송신기와 공격자 사이의 거리, 데이터 수를 조합하여 27개의 스푸핑 공격 시뮬레이션을 생성했습니다. 이후 각 경우의 수마다 100번의 데이터 수집을 진행하였고 총 2700개의 스푸핑 공격 데이터를 생성했습니다.

#### b. Moveable Data

송신기가 움직이는 상황에서의 성능 평가에 사용되는 데이터입니다.

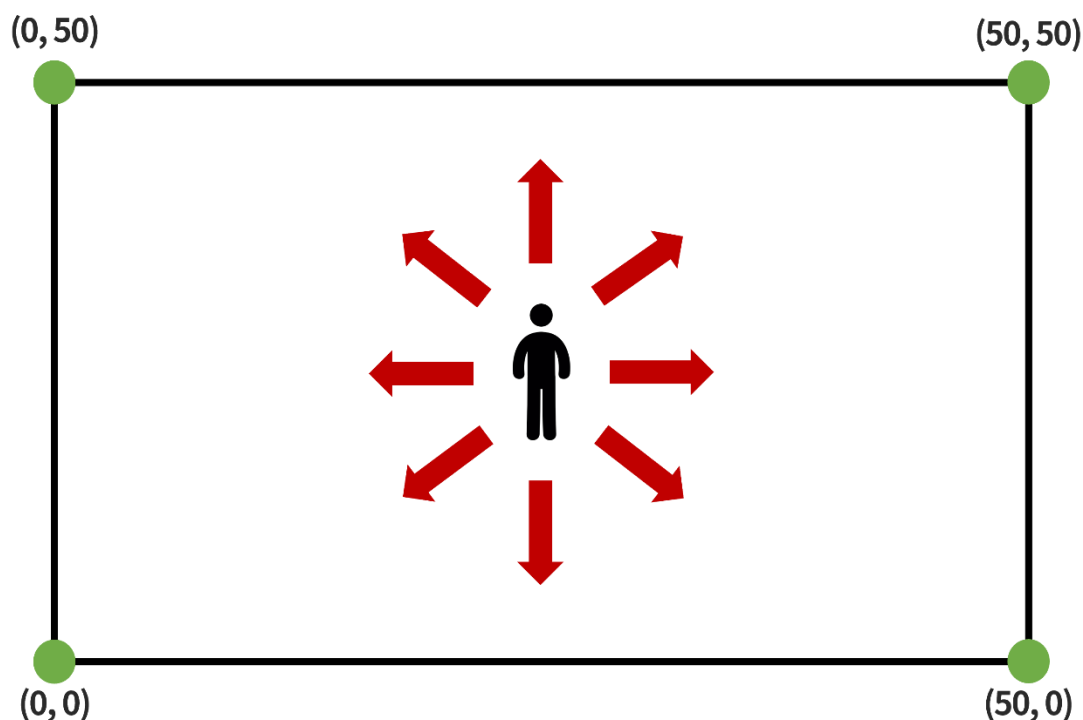


그림 11. 송신기가 움직이는 경우의 데이터 수집

송신기는 최초 (0, 0)의 위치에서 그림과 같이 정의된 8방향 중 하나를 무작위로 선택하여 이동할 수 있습니다. 이때 송신기의 이동 속도와 이동 거리도 무작위로 선택됩니다 (이동 속도는 1~3 초, 이동 거리는 1~5 중 무작위로 선택). 한 개의 데이터 파일은 위와 같은 과정을 천 번 반복합니다. 그리고 송신기와 공격자의 거리를 0m, 1m, 2m로 나누어 스푸핑 공격 상황을 가정합니다. 본 프로젝트는 성능 평가를 위해 송신기와 공격자 사이의 거리별로 3개의 스푸핑 공격 시뮬레이션을 생성했습니다. 이후 각 경우의 수마다 1000번의 데이터 수집을 진행하였고 총 3000개의 스푸핑 공격 데이터를 생성했습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

## B. Detection Phase

### a. Stationary

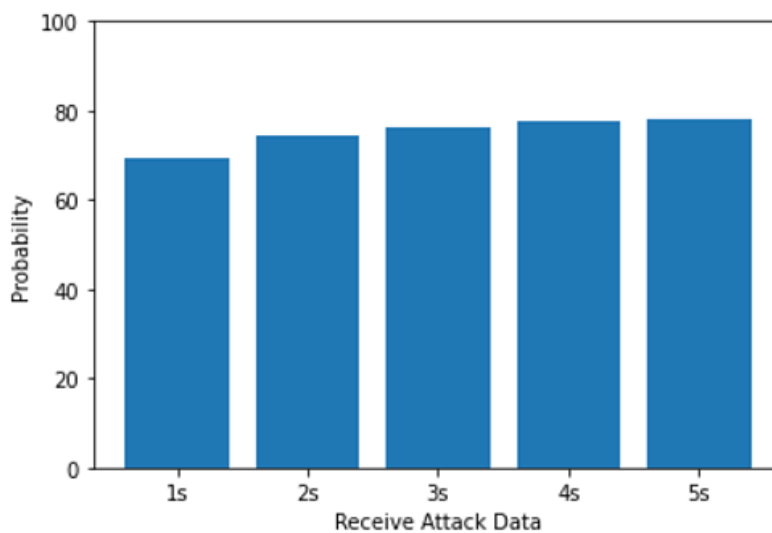


그림 12. Detection Phase의 공격 감지 정확도

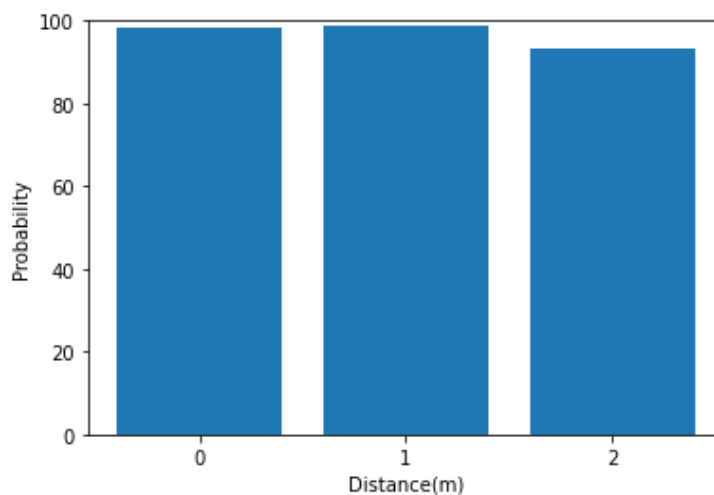


그림 13. 1초간 스푸핑 공격이 진행되었을 때 거리 별 Time Interval 검사의 정확도

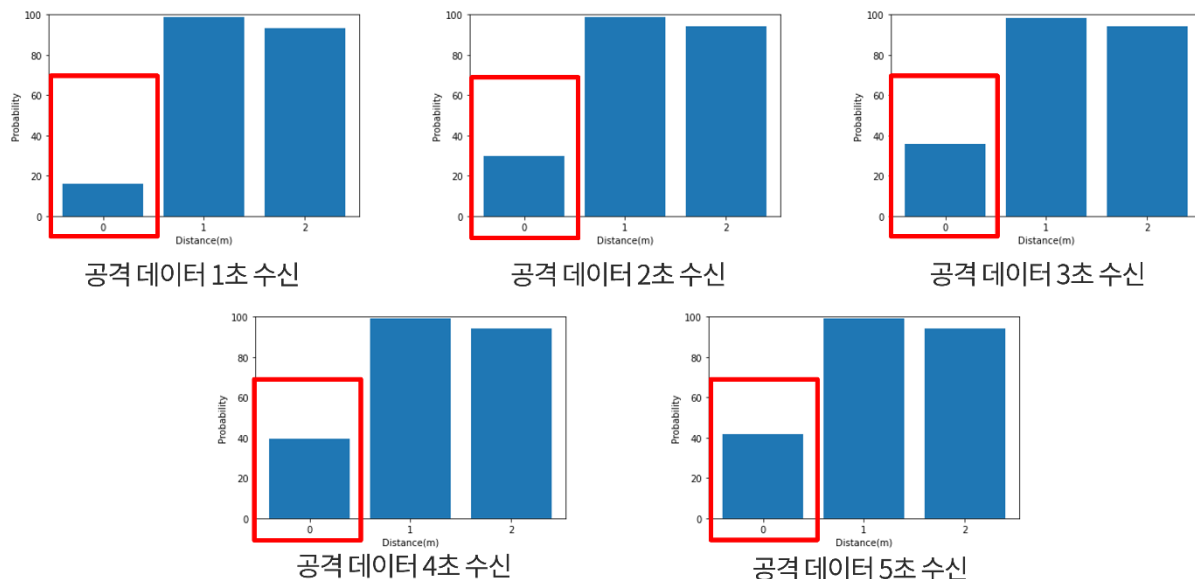


그림 14. 송신기가 움직이지 않는 경우 거리 별 RSSI 검사의 정확도

송신기가 움직이지 않는 상황에서 공격자는 1~5초간 스푸핑 공격을 진행합니다. 스푸핑 공격의 진행 시간이 길어질수록 공격 감지 정확도는 점진적으로 높아집니다. 전체적인 공격 감지 정확도는 70~80%로 높은 수치가 아님을 확인할 수 있습니다. 이러한 이유는 송신기가 움직이지 않는 상황에서 성능 평가 데이터의 모든 경우엔 유저의 송신기와 공격자의 거리가 매우 가까운 경우 (0m)를 포함하고 있기 때문입니다. 위 그림 13을 보시면 Time Interval 검사는 스푸핑 공격이 1초 동안 발생한 상황에서 유저의 송신기와 공격자의 거리에 관계없이 높은 공격 감지 정확도를 보입니다. 하지만 그림 14를 보시면 RSSI 검사에서 유저의 송신기와 공격자의 거리가 가까울 때 공격 감지 정확도가 매우 낮은 것을 보실 수 있습니다.

#### b. Moveable

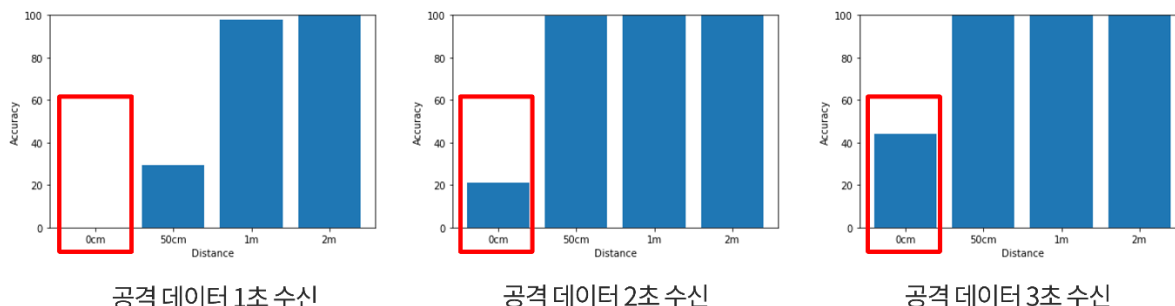



그림 15. 송신기가 움직이는 경우 거리 별 RSSI 검사의 정확도

송신기가 움직이는 상황에서 공격자는 1~3초간 스푸핑 공격을 진행합니다. 그림 15를 보시면

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

Stationary한 경우와 같이 스푸핑 공격의 진행 시간이 길어질수록 공격 감지 정확도는 점진적으로 높아지며, 모든 경우에서 유저의 송신기와 공격자의 거리가 매우 가까울 때 낮은 공격 감지 정확도를 보입니다. 이처럼 Stationary한 경우와 Moveable의 경우에서 공통적으로 발생하는 문제점을 해결하기 위해 SADS는 Unicast Phase를 추가하였습니다.

### C. SADS (Detection Phase + Unicast Phase)

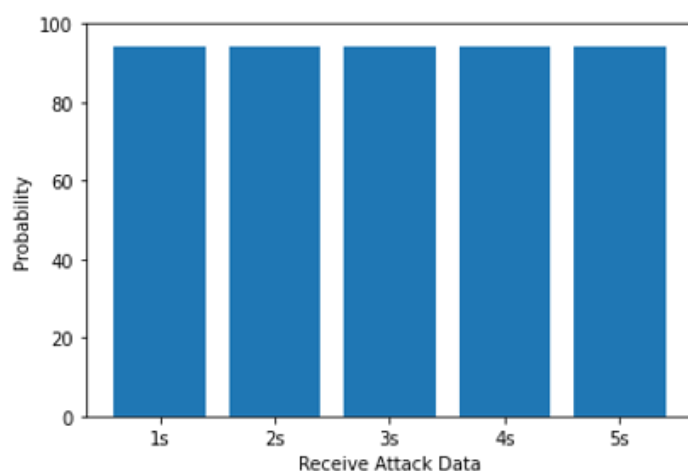


그림 16. SADS의 공격 감지 정확도

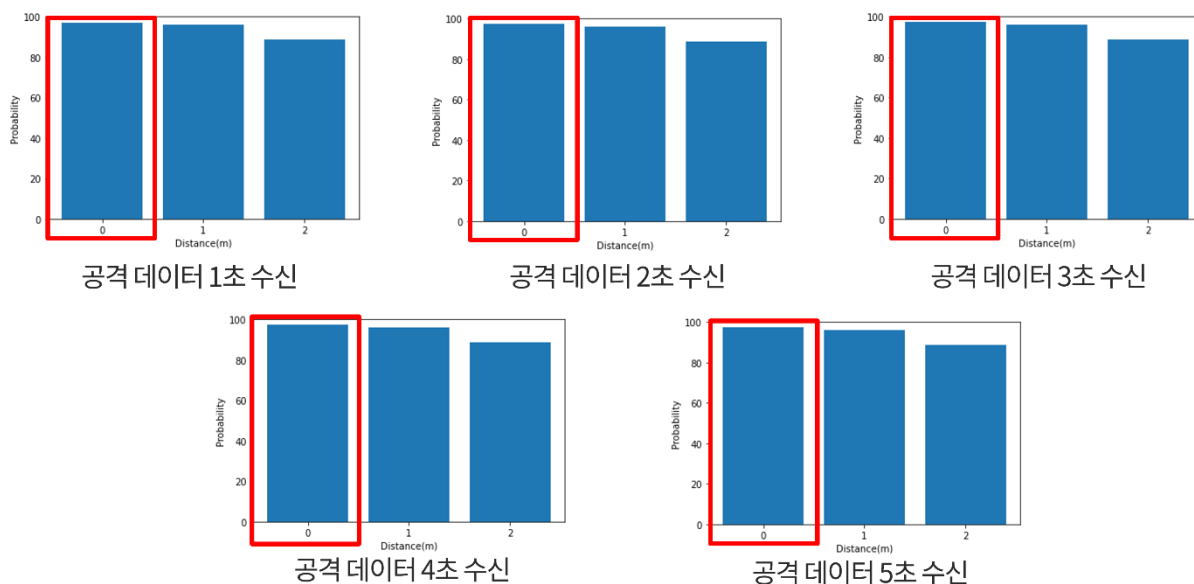


그림 17. 거리 별 SADS 공격 감지 정확도



 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>다학제간캡스톤디자인</b> </div>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

그림 16을 보시면 공격자가 1~5초간 스푸핑 공격을 진행하는 상황에서 Unicast Phase를 결합한 SADS는 유저의 송신기와 공격자의 거리와 관계없이 약 95%의 매우 높은 공격 감지 정확도를 보입니다. 평균적인 Detection Phase의 공격 감지 정확도는 80% 내외로 Unicast Phase를 결합했을 때 약 15% 정도 공격 감지 정확도가 높아진 것을 확인할 수 있습니다.

## 7 Limitation

### 7.1 현실적 제한 요소

Detection Phase가 유효한 스푸핑 공격 감지 정확도를 보이기 위해선 셋업 과정이 필요합니다. 본 프로젝트에서 정한 셋업 과정은 아래와 같습니다.

- 신뢰 구간에 속한 INT 데이터가 최소 10개 이상 있어야합니다.
- KF로 평활화된 RSSI 데이터가 최소 10개 이상 있어야합니다.

따라서 셋업 과정에서 벌어지는 스푸핑 공격은 안정적으로 탐지할 수 없습니다.

### 7.2 해결 방안

Detection Phase는 스푸핑 공격 검사를 위해 기준이 되는 값이 필수로 요구됩니다. 따라서 셋업 과정에서 필요한 데이터의 개수를 줄일 순 있지만, 셋업 과정을 삭제하는 것은 불가능하다고 생각합니다. 하지만 중간에 송신기와 수신기의 통신이 끊기지 않는 일대일 통신이 먼저 진행되고, 이후 Detection Phase를 통해 스푸핑 검사를 한다면 셋업 과정을 삭제할 수 있을 것이라 생각합니다.

## 8 Use Case

### 8.1 활용방안

#### 손흥민 홈구장 '토트넘 핫스퍼 스타디움'에 숨은 최신 기술들

구장 내 와이파이에는 6만명 이상의 관람객 수요를 수용한다. 블루투스 비콘은 관람객들이 스타디움 공식 앱을 통해 다양한 편의시설을 찾기 쉽도록 도와준다. 핫스퍼 스타디움 내부에 자체 소형 양조장과 미술랭 스타를 받은 식당을 포함한 65개의 식당, 스낵코너 등 다양한 상점이 있다.

<축구장>

 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>다학제간캡스톤디자인</b> </div>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

## 스마트한 서비스 신설...교통약자 대중교통 편의성 ↑

지하철 역사에 설치된 9000여개의 블루투스 기기 '비콘(Beacon)'을 활용해 정보를 제공한다.

도착역 알림 서비스는 지하철 탑승 후 내릴 역을 앱에 등록하면 '비콘'이 열차위치를 실시간으로 자동 파악해 해당 역에 도착 시 휴대폰 문자·음성을 통해 알려준다.

서울교통공사가 운영하는 1~8호선 구간 뿐 아니라 코레일 구간도 이용할 수 있다.

<대중교통>

아이비콘은 미국 프로야구 메이저리그(MLB) 구장에도 구축됐다. 지난 2013년 뉴욕 메츠의 '시티 필즈' 구장은 미국 야구장 최초로 서비스를 도입했다. MLB의 '볼파크' 앱과 연동돼 구장 내 판매상품 등 다양한 정보를 제공한다. 이어 LA 다저스와 샌디에고 파드리스의 구장에도 65개의 아이비콘이 설치됐다. MLB는 향후 미국 내 모든 구장에 서비스를 구축할 계획이다.

<야구장>

## 국민건강보험 일산병원, 미래 의료를 디자인하다

의료진을 포함한 일산병원 직원들의 목걸이형 사원증에는 본인 동의하에 비콘(Beacon)이 달려있다. 비콘은 위치 정보를 전달하기 위해 신호를 주기적으로 전송하는 기기를 말한다. 의료진뿐만 아니라 환자와 보호자에도 붙어있다. 이 비콘은 병원 전체(지하2층~지상13층)에 설치된 900여개 감지기에 동선과 위치를 알려준다. 이로 인해 감염병 환자가 발생할 경우 병원 내 동선과 밀접접촉자, 격리대상자를 짧은 시간에 정확히 찾아낼 수 있다.

<병원>

### 그림 18. BLE Beacon을 활용한 실내 위치 측위 사례

그림 18을 보시면 BLE Beacon을 활용한 여러 실내 위치 측위 사례가 있으며, 앞으로 더 많은 BLE Beacon이 설치될 것이란 걸 알 수 있습니다. 축구장이나 야구장의 경우 공격자의 잘못된 광고 패킷 송신으로 인해 팬들의 위치를 혼동시켜 좌석을 찾는데 많은 혼란을 야기할 수 있습니다. 대중교통의 경우 잘못된 위치 정보를 출력하여 승객들의 시간을 허비하게 할 수 있습니다. 특히 병원의 경우 환자의 위치 정보를 잘못 출력한다면, 응급 상황에 잘 대처할 수 없다는 큰 문제점이 있습니다. SADS는 이러한 실내 위치 측위 사례에서 스푸핑 공격을 방어하는데 사용될 수 있습니다.

## 8.2 시스템 확장성

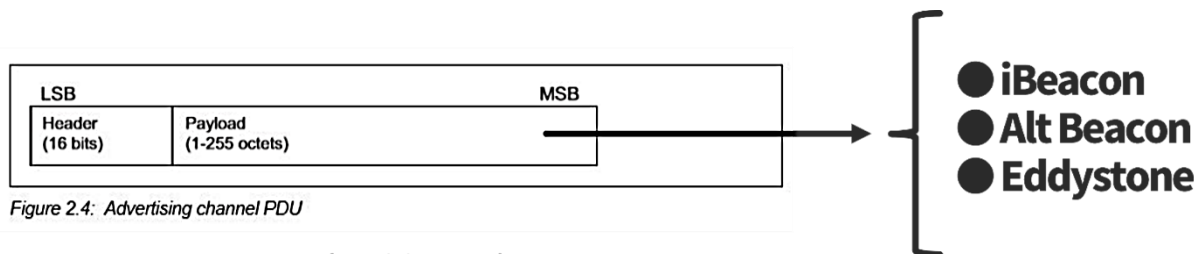


Figure 2.4: Advertising channel PDU

< Advertising Packet >

그림 19. 광고 패킷 내부 구성도

### A. BLE Beacon

송신기가 송신하는 광고 패킷은 그림 19와 같이 Header와 Payload로 구성됩니다. 그리고 Payload의 규격에 따라 애플의 iBeacon, 안드로이드의 Alt Beacon, 구글의 Eddystone 등으로 구분됩니다. SADS는 Payload의 정보가 아닌 광고 패킷의 물리적 요소 (시간 간격, RSSI 등)만을 활용하여 스푸핑 공격을 탐지합니다. 따라서 SADS는 위와 같은 다양한 종류의 모든 비콘에서 발생하는 스푸핑 공격을 감지할 수 있습니다. 비콘 신호는 현재 실내 위치 측위 뿐만 아니라 거리 인식, 물체 인식, 로봇틱스 등의 다양한 분야에서 사용되고 있습니다. 따라서 SADS는 단순히 "Spoofing Attack Detection System at Indoor Positioning using BLE"에만 해당되는 것이 아닌 "Spoofing Attack Detection System on BLE Beacon"으로 확장될 수 있습니다.


### B. BLE Devices

BLE 기기들은 광고 모드와 연결 모드로 구분됩니다. 성공적인 스푸핑 공격을 위해서 공격자는 시스템에 한번 이상 광고 패킷을 방송해야 하므로 SADS는 광고 패킷을 송수신하는 광고 모드에서 발생하는 스푸핑 공격을 유효하게 감지할 수 있습니다. 하지만, 아직 연결 모드에서 발생하는 스푸핑 공격은 감지할 수 없습니다. 추후 연결 모드에서 발생하는 스푸핑 공격을 감지할 수 있는 모델을 개발하여 BLE Beacon 뿐만 아니라 모든 BLE 기기에서 사용할 수 있는 스푸핑 공격 감지 시스템을 제작할 예정입니다.

## 9 Result

### 9.1 결과

본 프로젝트에서는 송신기에서 방송되는 광고 패킷의 물리적인 요소와 송수신기의 일대일 통신을 활용하여 스푸핑 공격을 감지하고 공격자의 광고 패킷을 탐지하는 시스템을 제안하였습니다. SADS는 광고 패킷의 수신 시간 간격과 RSSI를 사용하여 스푸핑 공격을 감지하고 공격자의 광고 패킷을 특정하여 스푸핑 공격의 범위를 제한합니다. 또한 Unicast Phase을 도입함으로써 유저의 송신기와 공격자의 위치가 같더라도 공격자의 광고 패킷을 탐지할 수도 있습니다. 따라서 공격 시뮬레이션에서 SADS는 95% 이상의 스푸핑 공격 감지 정확도를 보이며 유저의 송신기와 공격자

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

사이의 거리에 관계없이 모든 경우에서 스푸핑 공격을 감지할 수 있습니다. SADS는 공격자의 광고 패킷을 특정할 수 있기 때문에 공격자의 광고 패킷만을 수집하여 공격자 정보를 분석할 수 있습니다. 향후 이 정보를 이용해 공격자의 위치를 파악하여 2차 피해를 예방할 수 있는 방법을 연구하고, 더 나아가 모든 BLE 장치에서 사용될 수 있는 스푸핑 공격 감지 시스템을 개발할 예정입니다.

## 9.2 기대 효과

본 프로젝트에서는 SADS가 스푸핑 공격을 효과적으로 탐지할 수 있다는 것을 증명했습니다. 최근 연구에 따르면 기존 BLE 장치의 약 80%가 별도의 인증 절차 없이 평문으로 사용자의 장치와 통신하며 [21], 이러한 BLE 장치는 2025년까지 64억개 이상 생산될 예정입니다 [28]. 실제로 2022년 US San Diego의 연구에서 일반적으로 사용하고 있는 40% 이상의 BLE Beacon과 BLE 장치들의 고유 식별자들이 노출되어 있음을 밝혔습니다 [29]. 그리고 대부분의 BLE 장치들은 입출력 기능에 제한되어 의존되므로 보안 인증 메커니즘을 사용하 기엔 어려움이 있습니다. 따라서 SADS는 위와 같은 보안이 취약한 장치들에서 발생하는 스푸핑 공격을 감 지하여 BLE의 보안을 향상시킬 수 있습니다.


## 9.3 자기 평가

### 노용준 (20171616)

SADS는 목표 달성율을 보았을 때 충분히 성공적인 프로젝트입니다. 정했던 목표를 가볍게 달 성하였고, 중간 평가를 해주셨던 교수님들의 피드백을 수용하고 해결하기 위해 팀원과 노력했습 니다. 그러다보니 SADS가 훨씬 더 넓은 범주에서 사용될 수 있다는 것을 알게 되었습니다. 추가 개발없이 저희 프로젝트는 자연스레 확장되었고, 이후 더욱 넓은 범주에서 스푸핑 공격을 방어할 수 있는 시스템으로 만들 수 있는 기회를 얻었습니다.

SADS는 타원 곡선 암호학, 칼만 필터, 홀트 선형 추세 기법 등 복잡한 수식과 메커니즘을 많이 사용합니다. 때문에 개발 초기엔 시스템의 완성을 우선시하여 여러 라이브러리를 사용했었습니다. 하지만 라이브러리의 사용은 수식에 사용되는 여러 상수를 마음대로 조정할 수 없거나, 연산 시 시간이 오래 걸린다는 단점이 있었습니다. 따라서 저와 팀원은 역할을 나누어 각각의 수식과 메 커니즘을 공부하고 직접 개발하여 사용했습니다. 이 과정에서 개발의 역량이 많이 향상되었고, 라 이브러리나 API를 사용하는 것은 시간을 많이 절약하면서 편할 수 있지만 그것의 본질을 아는 것 이 얼마나 중요한 것인지 깨닫게 되었습니다.

저희 팀은 스푸핑 공격으로부터 BLE Beacon을 보호해줄 수 있는 장치나 프로그램을 제작한 것 이 아닙니다. BLE Beacon을 스푸핑 공격으로부터 방어할 수 있는 수단과 방법을 제안한 것이죠. 따라서 SADS가 독자적인 방어 체계로 사용되는 것은 매우 어렵다고 생각합니다. 하지만 저와 팀 원의 연구를 바탕으로 앞으로 출시될 여러 BLE 및 Beacon에 SADS와 비슷한 공격 방어 시스템이

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

구축된다면, 보안에 큰 약점을 가진 BLE의 문제점이 조금은 해결될 수 있을 것이라 생각합니다.

## 문성찬 (20171620)

앱이나 웹 개발이 아닌 연구라는 목적으로 진행한 이번 프로젝트에서 저와 팀원은 목표한 바 프로젝트를 완성할 수 있었습니다. 물론 중간중간 SADS 시스템에 대한 모순된 가설을 세우기도 했으며 실험 때 생각하지 못했던 변수들이 발생하면서, 시스템에 도입할 기술을 새로 찾아보거나 아예 처음부터 시스템을 새로 개발하기도 했습니다. 하지만 그 속에서도 저와 팀원은 좌절하지 않고 해결방안을 찾아냈으며, 결론적으로 목표한 바 시스템을 구현하고 연구할 수 있었습니다. 그러한 과정 속에서, 시스템에 대한 저희만의 가설을 세우고 해당 가설을 검증하기 위해 실험을 진행하며 가설이 타당한지 아닌지를 판단하는 과정을 거치며 연구라는 다소 생소한 분야에 대해서 경험을 쌓을 수 있었습니다.

또한 시스템에 대한 가설을 실제로 적용하기 위해 SADS에서는 홀트 선형 추세 기법, 칼만 필터, Elgamal 알고리즘 등과 같은 복잡한 메커니즘을 사용합니다. 따라서 저와 팀원은 해당 개념들을 제대로 이해하고 사용하기 위해 각종 논문과 자료들을 찾아보며 이해하려고 노력했습니다. 끝내 해당 기술들을 지원해주는 라이브러리가 있음에도 불구하고 저와 팀원은 각각의 기술들을 직접 구현하여 시스템에 적용함으로써 연산 능력의 저하나 파라미터 수정 불가와 같은 불편함 없이 해당 기술들을 온전히 사용할 수 있었으며, 이 과정에서 기술 개발 및 논문 탐색에 대한 역량을 향상시킬 수 있었습니다.

연구한 내용을 정리하고 기록하는 것은 중요하다라는 지도 교수님의 조언을 통해 저와 팀원은 매주 한주동안 연구한 내용을 보고서 형식으로 정리해 기록해왔습니다. 보고서를 작성하면서 현재까지 진행한 연구 내용에 대해서 다시 한번 공부할 수 있었으며 그 속에서 저와 팀원이 놓친 부분 또한 발견할 수도 있어서 많은 도움이 되었습니다. 추후 이러한 보고서들은 발표 자료와 결과 보고서 등을 작성할 때도 유용했으며 결론적으로 SADS를 설계하는데 기반이 되는 자료로서 사용되어, 보고서 정리에 대한 중요성을 느낄 수 있었습니다.

마지막으로 저희가 최종적으로 개발한 SADS는 단순히 저희가 세운 가설들을 입증하기 위한 모델이지 실제로 현업에서 적용가능한 모델은 아닙니다. 하지만 저희는 SADS를 통해 실내 위치 측위에서의 스푸핑 공격 탐지 및 예방에 대한 기술적인 방법을 제안했으며 성능 평가를 통해 의미 있는 결과를 만들 수 있었습니다. 더불어 중간 평가 때의 교수님들의 피드백을 수용한 저와 팀원은 프로젝트 주제의 확장을 탐색하고 공부했습니다. 결론적으로 SADS가 오로지 실내 위치 측위에만 적용되는 것이 아닌 BLE Beacon에서의 스푸핑 공격 탐지에도 적용될 수 있다는 사실을 확인함으로써 추후 연구를 더 진행해 BLE Beacon에서의 스푸핑 공격 탐지를 입증한다면, BLE 통신에서의 유의미한 보안 정책을 세울 수 있을 것 같습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

## 10 Reference

### 10.1 참고문헌

- [1] Röbesaat, J., Zhang, P., Abdelaal, M., & Theel, O. (2017). An improved BLE indoor localization with Kalman-based fusion: An experimental study. *Sensors (Switzerland)*, 17(5).  
<https://doi.org/10.3390/s17050951>
- [2] Shen, Y., Hwang, B., & Jeong, J. (2020). Particle Filtering-Based Indoor Positioning System for Beacon Tag Tracking. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2020.3045610>
- [3] Bluez. Accessed: Dec. 13, 2021. [Online]. Available: <http://www.bluez.org/>
- [4] Supplement to the Bluetooth Core Specification Bluetooth® Specification. (2011). [www.bluetooth.com](http://www.bluetooth.com).
- [5] Ahrens, L., Ahrens, J., & Schotten, H. D. (2019). A machine-learning phase classification scheme for anomaly detection in signals with periodic characteristics. *Eurasip Journal on Advances in Signal Processing*, 2019(1). <https://doi.org/10.1186/s13634-019-0619-3>
- [6] Lahmadi, A., Duque, A., Heraief, N., & Francq, J. (2020). MitM Attack Detection in BLE Networks using Reconstruction and Classification Machine Learning Techniques. <https://hal.inria.fr/hal-02948407>
- [7] Wu, J., Nan, Y., Kumar, V., Payer, M., & Xu, D. (n.d.). BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy Networks.
- [8] Tänzer, L., & Janwar, I. (n.d.). Guide to Elliptic Curve Cryptography.
- [9] Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., & Aboutanios, E. (2017). Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. In *IEEE Communications Surveys and Tutorials* (Vol. 19, Issue 2, pp. 1327–1346). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/COMST.2016.2632427>
- [10] Faheem, Z., Devetsikiotis, M., & Hacker, T. (n.d.). An iBeacon based Proximity and Indoor Localization System.
- [11] Kunhoth, J., Karkar, A. G., Al-Maadeed, S., & Al-Ali, A. (2020). Indoor positioning and wayfinding systems: a survey. In *Human-centric Computing and Information Sciences* (Vol. 10, Issue 1). Springer.  
<https://doi.org/10.1186/s13673-020-00222-0>
- [12] Sadowski, S., & Spachos, P. (2018). RSSI-Based Indoor Localization with the Internet of Things. *IEEE Access*, 6, 30149–30161. <https://doi.org/10.1109/ACCESS.2018.2843325>
- [13] Subhan, F., Khan, A., Saleem, S., Ahmed, S., Imran, M., Asghar, Z., & Bangash, J. I. (2019). Experimental analysis of received signals strength in Bluetooth Low Energy (BLE) and its effect on distance and position estimation. *Transactions on Emerging Telecommunications Technologies*. <https://doi.org/10.1002/ett.3793>
- [14] Wu, J., Nan, Y., Kumar, V., Tian, D., Bianchi, A., Payer, M., & Xu, D. (n.d.). *BLESA: Spoofing Attacks against Reconnections in Bluetooth Low Energy*.
- [15] Oliff, W., Filippoupolitis, A., & Loukas, G. (2017). Evaluating the impact of malicious spoofing attacks on Bluetooth low energy based occupancy detection systems. *Proceedings - 2017 15th IEEE/ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2017*, 379–385. <https://doi.org/10.1109/SERA.2017.7965755>
- [16] Chan, A. C.-F., & Chung, R. M. H. (n.d.). *Security and Privacy of Wireless Beacon Systems*.
- [17] LESTER, S. The Emergence of Bluetooth Low Energy. <http://www.contextis.com/resources/blog/emerge>

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

nce-bluetoothlow-energy/, May 2015.

- [18] Faheem, Z., Gkelias, A., & Leung, K. K. (n.d.). A Survey of Indoor Localization Systems and Technologies.
- [19] Chatfield, B., & Haddad, R. J. (2017). RSSI-Based Spoofing Detection in Smart Grid IEEE 802.11 Home Area Networks.
- [20] Chen, Y., Yang, J., Trappe, W., & Martin, R. P. (2010). Detecting and localizing identity-based attacks in wireless and sensor networks. *IEEE Transactions on Vehicular Technology*, 59(5), 2418–2434. <https://doi.org/10.1109/TVT.2010.2044904>
- [21] Tal Melamed. BLE Application Hacking. [https://owasp.org/www-pdf-archive//OWASP2017\\_HackingBLEApplications\\_TalMelamed.pdf](https://owasp.org/www-pdf-archive//OWASP2017_HackingBLEApplications_TalMelamed.pdf), 2017. Accessed: May 24, 2022
- [22] Amer Daeri, Amer R. Zerek, & Mohamed A. Abuinjam. (n.d.). ElGamal public-key encryption.
- [23] Sravana Kumar, D. (2012). Encryption Of Data Using Elliptic Curve Over Finite Fields. *International Journal of Distributed and Parallel Systems*, 3(1), 301–308. <https://doi.org/10.5121/ijdps.2012.3125>
- [24] Hazra, A. (2017). Using the confidence interval confidently. *Journal of Thoracic Disease*, 9(10), 4125–4130. <https://doi.org/10.21037/jtd.2017.09.14>
- [25] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
- [26] Simon, D. (2006). Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons.
- [27] Vinutha, H. P., Poornima, B., & Sagar, B. M. (2018). Detection of outliers using interquartile range technique from intrusion dataset. In *Information and Decision Sciences* (pp. 511–518). Springer, Singapore.
- [28] Bluetooth SIG. Bluetooth Market Update. <https://www.bluetooth.com/bluetoothresources/2019-bluetooth-market-update/>, 2021. Accessed: May 24, 2022
- [29] H.Givehchian and N.Bhaskar, "Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices." US San Diego, 2022

## 11 부록

### 11.1 실험 환경

#### **Server**

- Laptop
- Python 3

#### **Anchor Point (AP)**

- Raspberry Pi 4B
- Python 3

#### **Transmitter (Tag / Attacker)**



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

- Smart Phone (Android)
- Flutter

## 11.2 사전 설정

### Server

- 패키지 설치 (Python 3)
  - numpy
  - firebase-admin

```
$ pip install firebase-admin
```

- 실험 환경 설정
  - SADS의 Server 코드 다운로드
  - 사용하는 네트워크의 IP 주소 확인
  - main.py의 HOST 변수 값 IP 주소에 맞게 변경

### Anchor Point (AP)

- 패키지 설치 (Python 3)
  - numpy
  - scipy
  - sympy

```
$ pip3 install sympy
```

- bluez

```
$ sudo apt-get update
$ sudo apt-get install bluetooth bluez libbluetooth-dev
$ sudo python3 -m pip install pybluez
```

- 실험 환경 설정
  - SADS의 Anchor Point 코드 다운로드
  - Server에서 설정한 HOST변수 값 확인
  - udp-client.py의 initSetting 함수에서 serverHost 변수 값 설정

### Transmitter (Tag / Attacker)

- 실험 환경 설정
  - SADS의 Tag 코드 다운로드
  - Flutter Build 수행



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>다학제간캡스톤디자인I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	SADS: Spoofing Attack Detection System at Indoor Positioning using BLE	
	<b>팀 명</b>	04	
	Confidential Restricted	Version 2.0	2022-05-23

```
$ flutter run
```

## 11.3 실험 매뉴얼

### 1. Server 실행

```
$ python ./main.py
```

### 2. Anchor Point (AP) 실행

```
$ sudo python3 ./udp-client.py
```

### 3. Tag 실행

- 공개키 / 개인키 생성
- 광고 패킷 생성
- 광고 패킷 방송

### 4. Attacker 실행

- Tag와 동일한 개인키를 입력해 Tag와 동일한 식별자를 가지는 광고 패킷 생성
- 광고 패킷 방송