[사물인터넷 환경 상에서 안전한 PIN 입력 방법]

제 출 일 자	2016. 9. 23	
기관명/학교명	싱가포르 국가기술청 / 부산대학교	
소속부서/학과	Institute of Infocomm Research / 컴퓨터공학부	
성 명	서화정 / 박태환, 서규원	
연 락 처	휴대전화	010-3563-5804 (박태환)
	e-mail	hwajeong84@gmail.com

요약

정보통신 기술의 발전은 사용자가 언제어디서나 금융 서비스에 접근하는 것이 가능하게 하였다. 하지만 최근 도래하고 있는 사물인터넷 환경에서는 사용자의 많은 정보가 사물들에 설치된 수많은 센서를 통해 쉽게 관찰되어 노출될 수 있는 문제점을 가지고 있다. 이는 기존의 금융 보안 솔루션 상에서는 안전하다고 생각되었던 정보보호 기법들이 사물인터넷 환경 상에서는 더 이상 효력이 미치지 않음을 의미한다. 특히 IT 디바이스를 통해 비밀정보를 입력하는 사용자의 특정 입력 동작과 화면상에 표기되는 시각적 정보는 공격자의 소형 카메라 및 해킹 툴을 통해 쉽게 녹화 및 분석되어 비밀정보가 노출될 수 있는 문제점을 가진다. 블랙햇 USA'14에서는 구글 글라스와 같은 웨어러블 디바이스를 통해 사용자의 입력 장면을 원거리에서 녹화하고 이를 머신러닝 기법을 통해 분석하는 방식으로 사용자의 비밀번호를 확인하는 방법이 발표되었다. 이처럼 사물인터넷 디바이스는 언제 어디서나 아무런 제약 없이 비밀정보를 수집 및 분석하는 게 가능하게 함으로써 이전 보안 기법들에 대해 제고가 필요한 시점이다. 이러한 공격 기법을 방어하기 위해 제안된 보안키보드로는 무작위로 키보드의 레이아웃을 구성하거나 레이아웃이 실시간으로 지속적으로 변화하도록 하여 노출된 입력화면과 비밀정보의 연관성을 없애는 방법이다. 하지만 무작위로 키보드가 구성되는 경우에는 사용자가 해당 키보드를 입력하기 위해 매순간 레이아웃을 학습하는 시간이 필요할 뿐 아니라 지속적으로 변화하는 키보드는 사용 편의성을 많이 저하시키는 문제점을 가지고 있다. 이러한 문제점을 해결하기 위해 제안된 행 기반 보안키보드는 기존의 보안키보드와는 달리 행과 행의 시작 위치만을 무작위로 설정함으로써 사용자의 편의성을 높였을 뿐 아니라 공격자가 현재의 입력 값을 유추하는 것이 불가능하도록 하였다. 하지만 해당 기법의 경우에도 대・소문자 입력 시에 비밀 정보가 여전히 유출되는 문제점이 있을 뿐 아니라 모든 행과 행의 시작 위치가 무작위로 배치되어 사용자의 학습시간이 많이 요구되는 문제점이 있었다. 이를 방지하기 위해 본 논문에서는 무작위로 대・소문자 레이아웃을 생성하여 현재 레이아웃의 상태를 공격자가 알 수 없도록 하였다. 또한 행을 재배열하는 대신 전체 레이아웃을 이동시켜 배치하는 방안을 제시함으로써 사용 편의성을 높이도록 하였다. 두 번째로 해당 공격보다 보다 진화된 공격 시나리오로는 공격자가 사용자의 모든 시각적 정보를 확인할 수 있도록 공간 안에 몰래카메라를 설치하거나 화면을 해킹한 경우이다. 이 경우에는 사용자의 화면이 공격자에 의해 녹화되어 공개될 뿐 아니라 사용자의 입력 동작도 모두 공격자에 의해 관찰된다. 이러한 녹화 공격을 방어하기 위해서는 사용자만이 접근이 가능한 보안 채널을 따로 제공함으로써 공격자가 모든 시각적 정보를 이용하는 경우에도 안전한 입력이 가능하도록 하는 방안이 제시되고 있다. 대표적인 녹화 공격 방어 기법으로는 사용자만이 들을 수 있는 보안 라디오 채널을 통해 현재 입력되고 있는 레이아웃의 비밀 값 혹은 힌트를 알려주고 사용자가 해당 비밀 값 혹은 힌트에 따라 비밀번호를 입력하는 것이다. 하지만

지금까지 제안된 대부분의 녹화 공격 방어 키보드의 경우 경우의 수가 적어 단순한 숫자 키보드 상에서의 방어 기법이 논의되었다. 현재 대부분의 비밀번호가 보안성 강화를 위해 대·소문자, 숫자, 특수문자를 포함하도록 권장하고 있는 시점을 생각해 볼 때 최고의 보안이 제공되어야 하는 금융 보안에서는 QWERTY 키보드 상에서의 녹화 공격 방지 방안이 제시되어야 한다. 본 논문에서는 개선된 행 기반 보안 키보드의 기본적인 개념과 보안 라디오 채널을 통해 전달되는 비밀 값을 종합하여 자신이 원하는 입력이 가능하도록 하는 보안 키보드를 제안한다. 화면에 보이지 않는 실제 키보드는 행 기반 보안 키보드의 원리로 무작위로 생성되며 라디오 채널을 통해 현재 누르는 값이 확인되도록 하였다. 특히 대·소문자를 효율적으로 구분하기 위해 대문자는 남성의 목소리로 소문자는 여성의 목소리로 발음되도록 설계하였다. 사용자의 사용 편의성을 높이기 위해 기본적인 키보드 레이아웃이 화면에 표기되도록 하며 음성 채널 상으로는 문자 정보와 더불어 행 번호까지 같이 사용자에게 전달되도록 하였다. 본 논문에서 제시한 두 보안 키보드는 실제로 구현 및 테스트를 통해 이전 보안 키보드보다 보안성과 실용성이 월등히 향상되었음을 확인하였다.

논문 목차

I. 서론

인터넷을 통해 은행 업무를 이용하는 것이 가능하게 하는 온라인 금융서비스는 사용자의 컴퓨터 혹은 스마트 폰으로 시간과 장소의 제약 없이 가능하도록 한 획기적인 서비스이다. 이러한 금융서비스는 사용자의 금전 정보를 다루는 만큼 보안에 각별히 신경을 써야한다. 현재 온라인 금융 서비스를 안전하게 이용하기 위해 사용자의 비밀 번호 혹은 비밀 정보를 통해 적법한 사용자임을 인증을 거치도록 하며 인증에 사용되는 대표적인 수단으로는 크게 PIN (Personal Identification Number) 번호 그리고 생체 정보 (지문, 홍채) 로 나누어 볼 수 있다. 그 중에서도 PIN 번호는 추가적인 하드웨어가 필요하지 않고 가장 보편화된 인증 기술로써 지금도 대부분의 금융서비스 인증에 사용되고 있다.

하지만 사물인터넷 시대가 옴에 따라 사용자의 거의 모든 정보가 사물에 부착된 센서들을 통해 쉽게 수집 및 분석되어 비밀정보가 쉽게 노출될 수 있는 문제점이 제기되고 있다. 2014년 블랫햇 USA에서 제안된 구글 글라스 기반의 키보드 위치 유추 공격은 사용자로부터 3m 거리에 위치한 공격자가 사용자가 터치스크린을 통해 비밀번호를 입력하는 모습을 촬영한 다음 사용자의 손동작의 각도와 움직임을 정밀히 분석하여 90%의 정확도로 비밀번호를 유추해 낼 수 있음을 증명하였다 [1]. 이처럼 사물인터넷 디바이스가 고도화됨에 따라 기존의 안전하다고 여겨졌던 사용자 인증 기법의 취약점이 하나둘씩 나타나고 있다. 이를 방지하기 위해 제안된 방안은 화면상에 나타나는 키보드 레이아웃의 위치를 무작위로 변경하여 공격자가 사용자의 손끝이 향하는 위치점을 유추하더라도 정확한 키보드의 레이아웃이 나타내는 값을 확인하는 것이 불가능하도록 하는 것이다. 해당 기법은 구글 글라스를 통한 유추공격을 효과적으로 방어할 수는 있지만 사용자의 입력 편이성이 저하되는 문제점을 가지고 있다. 이를 개선하기 위해서 [2]에서 제시된 방안은 QWERTY 키보드의 행을 기준으로 입력키를 묶은 다음 무작위로 재 정렬함으로써 동일한 입력 값에 대해서도 서로 다른 입력 위치가 선택되도록 하는 기법이다. 이는 모든 문자를 무작위로 섞지 않기 때문에 사용자가 원하는 값을 보다 쉽게 찾을 수 있다는 장점을 가진다. 하지만 여전히 공격자가 사용자의 비밀번호 입력 장면을 분석하여 사용자의 대・소문자를 확인할 수 있을 뿐 아니라 공격자가 사용자의 화면을 온전히 녹화하는 경우에는 모든 정보가 쉽게 노출될 수 있는 문제점을 가진다.

본 논문에서는 [2]에서 제안된 보안 키보드의 취약점을 개선하고 사용 용이성을 향상시킨 두 개의 보안 키보드를 제안한다. 첫 번째 키보드는 대・소문자의 배치를 무작위로 설정하여 공격자가 현재 입력되는 키보드의 대・소문자 배치를 확인하는 것이 불가능하도록 하는 기법이다. 두 번째 보안키보드는 사용자의 화면이 공격자에 의해 녹화되는 경우에도 비밀 정보가 노출되지 않도록 사용자만이 비밀 정보를 안전하게 받을 수 있는 음성 채널을 통해 적절한 피드백을 주는 보안키보드 기법을 제시한다. 해당 보안 키보드는 실제로 안드로이드 스마트폰 상에서 구현되었으며 여러 명의 실 사용자

를 통해 테스트됨으로써 그 보안성과 실용성이 확인되었다.

본 논문의 구성은 다음과 같다. 2장에서는 키보드를 통한 인증에 대한 공격과 이를 방지하는 보안 키보드에 대해 살펴본다. 3장에서는 제안하는 키패드의 특징에 대하여 설명한다. 4장에서는 제안된 키보드의 성능에 대해 평가한다. 마지막으로 5장에서는 본 논문의 결론을 내린다.

<그림 1> 구글 글라스를 통한 어깨너머 공격 [1]

<그림 2> 터칭 프레임에 대한 정의 [1]

<그림 3> 입력값에 대한 위치 공격에 취약한 사용자 인증 기법 [3]

Ⅱ. 관련 연구

1. 어깨너머 공격을 통한 키보드 해킹 기술

어깨너머 공격은 사용자의 입력과정을 공격자가 관찰하여 사용자의 비밀번호를 유추해내는 기법을 의미한다. 해당 관찰의 범위는 사물인터넷 디바이스의 발전으로 점차 광범위해 지고 있다. 블랫햇 USA'14에서 제시된 구글 글라스 기반 공격은 원거리에서 구글글라스로 사용자의 입력화면을 녹화한 이후에 Deformable Part-based Model을 기반으로 사용자가 입력하고 있는 물체를 확인한 후에 k-means clustering과 같은 머신러닝기법을 적용하여 사용자가 입력하는 값을 정확히 판단한다. 해당 기법은 공격자가 원거리에 위치하는 경우에도 사용자의 비밀번호를 정확하게 알아낼 수 있다는 것을 증명함으로써 기존의 보안키보드에 대해 다시 한 번 고찰하게 만드는 계기를 마련하였다. <그림 1>은 원거리에서 구글 글라스를 통해 어깨너머 공격을 시도하는 모습을 나

타내고 있다. <그림 2>는 구글 글라스를 통해 사용자를 관찰하는 화면으로써 정확히 사용자의 터칭 프레임을 정의하고 그 공간 안에서 사용자가 입력하는 값의 좌표를 확인하게 된다. 이러한 입력 값에 대한 위치 공격은 특히 <그림 3>에서와 같이 고정된 레이아웃을 사용하는 스마트폰, ATM 기기 그리고 자동 도어락 상에서 PIN을 이용한 사용자 인증 시 비밀번호를 노출하게 되는 문제점을 가지고 있다 [3]. 따라서 사용자는 사물인터넷 시대에는 보안성이 강화된 보안키보드를 통한 PIN 입력이 요구되어 지고 있다.

<그림 4> 색상 기반 보안 키보드 [4]

<그림 5> 진동기반 보안 키보드 [5]

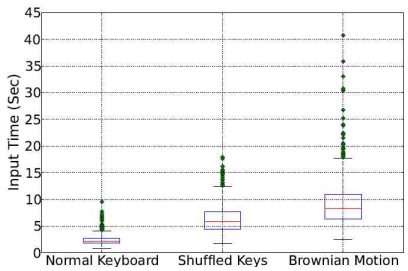
<그림 6> 입력메시지 암호화 기반 보안 키보드 [6]

2. 이전 보안 키보드 기술

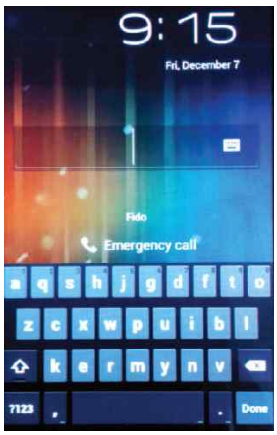
초기의 QWERTY 키보드 상에서의 안전한 보안키보드의 경우 공격자가 어깨너머 공격을 통해 사용자의 상단 입력창을 확인함으로써 비밀번호가 노출되는 공격을 방지하는 방안이 제시되었다. <그림 4>에서와 같이 2013년도 금융보안공모전 우수상 논문에 소개된 색상기반 보안 키패드의 동작방식은 기존에 숫자 혹은 글자를 그대로 입력창에 보여주던 피드백 방식 대신 각각의 키에 매칭되는 색상을 입력창에 표기함으로써 공격자가 입력되고 있는 문자를 확인하는 것이 불가능하게 하였다 [4]. 또한 매번 자판의 색상을 무작위로 배치하여 색상정보를 통해 기존의 입력정보를 유추하는 것이 불가능하도록 하였다. <그림 5>에서와 같이 2014년도 지급결제제도 발전 논문 현상공모 아이디어상 논문에서 제시된 보안키보드에서는 기존의 문자 피드백 대신 스마트 폰 상의 진동정보를 통해 사용자에게 적합한 피드백을 줌으로써 공격자가 시각정보를 통해 사용자의 비밀정보를 확인하는 것이 불가능하게 하였다 [5]. 이는 서로 인접한 키패드 간에 서로 다른 패턴의 진동을 발생시켜 사용자가 눈이 아닌 손의 진동으로 현재 입력되는 값의 피드백을 확인하는 기법이었다. 그 다음으로 제시된 <그림 6>의 방식은 기존의 피드백 방식이 한 문자 각각에 대한 피드백을 주는 방식이었다면 해당 기법은 제대

로 된 입력이 들어온 경우에만 마지막에 제대로 입력되었다는 피드백을 주는 방식이 제시되었다 [6]. 이는 사용자가 자신의 입력이 적합함을 확인함과 동시에 공격자의 어깨너머 공격을 효과적으로 방어할 수 있는 방안이다. 하지만 블랙햇 USA'14에서 제시된 구글 글라스 기반 어깨너머 공격은 제안하는 보안 키보드 기법을 통해서는 방어가 불가능하다. 그 이유는 해당 보안 키보드의 레이아웃 정보는 변화가 없기 때문이다. 따라서 이를 방어하기 위한 보안키보드에 대한 연구가 활발히 진행되었다.

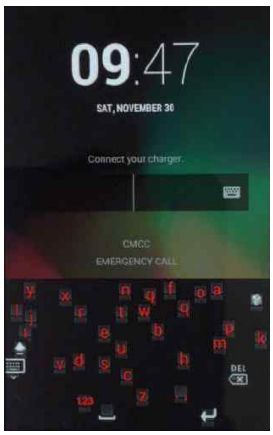
<그림 7> 입력 속도 비교 [1]



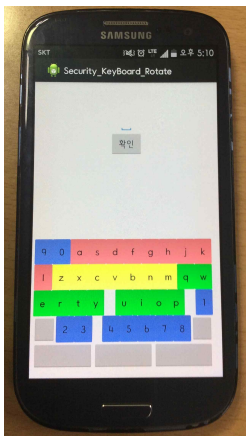
<그림 8> 무작위 레이아웃 [1]



<그림 9> 레이아웃 이동 [1]



<그림 10> 행기반 레이아웃 [2]



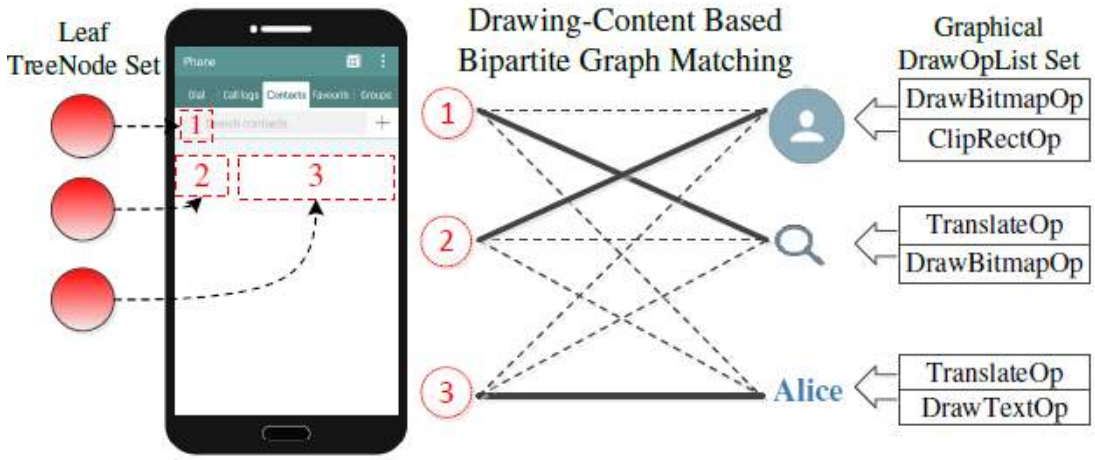
구글 글라스를 통한 어깨너머 공격을 효율적으로 방어하기 위해서는 기존에 알려진 키보드 레이아웃 정보를 변경하여 공격자가 기존의 레이아웃 정보를 통해 현재 입력되는 값을 유추하지 못하도록 해야 한다. 가장 기본적인 방법으로는 레이아웃을 무작위로 배치하는 기법이다. <그림 8>에서와 같이 기존의 QWERTY 키보드의 레이아웃 형식은 그대로 유지하지만 실제로 입력되는 값은 무작위로 배치됨으로써 공격자가 기존에 알려진 키보드 레이아웃 정보로는 현재 입력되고 있는 값을 확인하는 것이 불가능하게 하였다. <그림 9>에서는 레이아웃이 실시간으로 변경되도록 하여 동일한 위치 점을 누르는 경우에도 서로 다른 입력 값이 선택되도록 하는 기법이다. 하지만 해당 기법들은 사전에 알려진 레이아웃을 변경하여 새로운 형태의 키보드 레이아웃을 구성하였기 때문에 해당 키보드를 사용하는 매 순간마다 사용자는 레이아웃을 새로 학습해야 하는 문제점을 가진다. 그 결과 <그림 7>에서와 같이 사용자의 입력 시간이 기존 키보드에 비해 많이 걸린 뿐 아니라 사용자마다 해당 키보드에 대한 적응도 차이 (표준편차)가 큰 문제점을 가진다.

<그림 10>에서는 무작위로 변경되는 키보드 레이아웃의 불편함을 해소하기 위해 기존의 레이아웃 정보를 최대한 활용하면서 보안성을 확보하는 방안이 제안되었다 [2]. 기존의 키보드 레이아웃은 총 4개의 행으로 이루어지며 1행에는 10개, 2행에는 10개, 3행에는 9개 그리고 4행에는 7개의 입력 값이 위치한다 (1행은 "1~0", 2행은 "Q~P",

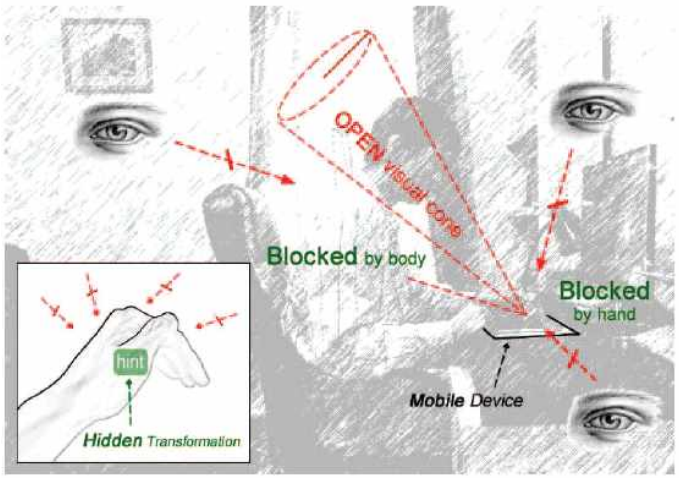
3행은 “A~L”, 4행은 “Z~M” 으로 구성된다). 본 기법에서는 각행의 값들을 묶은 다음에 4개의 행들을 무작위로 변경하도록 하였다. 만약 (3, 4, 2, 1)로 행이 배열되는 경우에는 “A~L”, “Z~M”, “Q~P”, “1~0” 으로 행이 구성된다. 행이 무작위로 변경된 이후에는 다시 행의 시작점을 무작위로 이동시키게 된다. 따라서 만약 선택된 무작위 값이 2가 되고 열의 배치가 (3, 4, 2, 1)가 되는 경우의 예시는 <그림 10>와 같다. 특히 행의 시작위치가 2가 되는 경우에는 가장 하위의 값 2개가 올라와서 키보드 레이아웃의 첫 번째 줄을 채우게 된다. 따라서 키보드 레이아웃 상에는 회전되는 방식으로 전체 값을 나타내게 된다.

위에서 제시된 보안키보드 기법들은 공격자가 사용자의 입력화면을 볼 수 없고 멀리서 사용자의 입력 장면을 관찰하여 비밀번호를 알아내는 방법을 방어하는데 효과적인 보안키보드 기법이다. 하지만 사물인터넷 환경 상에서는 초소형 카메라가 탑재된 사물이 사용자의 근처에서 사용자의 입력화면을 녹화할 수 있는 위험성이 제기되고 있다. 이와 더불어 스마트 폰 상에서의 입력화면이 공격자의 해킹에 의해 쉽게 노출될 수 있음이 최근 CCS’15 논문에서 증명되었다 [7]. 해당 논문에서는 사용자의 스마트폰 상에서 이전에 실행했던 어플리케이션의 GUI정보를 통해 실제 보이는 화면을 재구성하는 것이 가능함을 증명하였다. 즉 어플리케이션을 실행한 이후에도 메모리에는 GUI 정보가 남아 있게 되는데 해당 정보를 <그림 11>에서와 같이 그래픽 매칭을 통해 실제화면으로 재구성하는 것이다. 해당 기법은 사용자가 PIN 정보를 입력하는 화면 또한 공격자의 해킹에 의해 쉽게 노출될 수 있음을 의미한다. 따라서 보안 키보드의 입력화면이 공격자에 의해 노출되는 경우에도 안전하게 입력이 가능한 기법에 대한 연구가 필요하다.

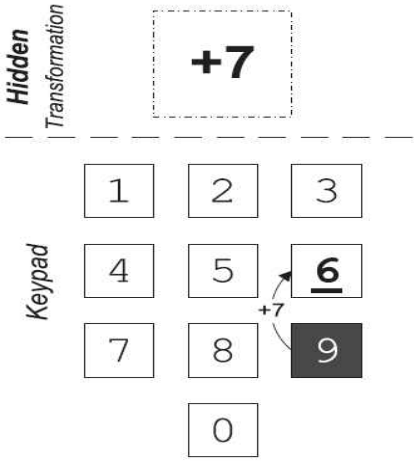
<그림 11> 그래프 매칭을 통한 GUI 재생성 기법 [7]



<그림 12>
숨겨진 정보를 통한 비밀번호 입력방법론 [8]



<그림 13> 숨겨진 번호를 통한 숫자기반 보안 키보드 [8]



<그림 14> 비밀번호가 “1” 인 경우 [9]

공격자에 의해 사용자의 입력 장면이 노출되는 경우를 방지하기 위해 [8]에서는 안전한 채널을 통해 비밀정보에 대한 힌트를 제공함으로써 공개된 레이아웃 정보 상에서 안전하게 비밀번호를 입력하는 방안이 제시되었다. <그림 12>에서와 같이 사용자는 자신만이 알 수 있는 경로를 통해 비밀정보에 대한 힌트를 제공받게 된다. <그림 13>에서는 숨겨진 비밀정보 힌트와 현재 제시된 레이아웃을 종합하여 안전하게 비밀번호를 입력하는 방안을 제시하고 있다. 만약 숨겨진 비밀정보 힌트가 “+7” 이고 자신의 비밀번호가 “9” 인 경우 연산 (7+9 mod 10)을 하여 나머지 “6” 을 결과 값으로 입력하게 된다. 이는 공격자가 숨겨진 비밀정보 힌트를 확인할 수 없는 경우 난수 값과 섞인 실제 비밀번호를 확인하는 것이 불가능하다. 이처럼 녹화 공격를 방어하기 위해서는 사용자만이 알 수 있는 비밀정보에 대한 힌트를 안전하게 전달하는 방안이 강구되어야

하며 현재 많은 논문에서 연구되고 있다. [9]에서는 자신이 입력하고자하는 값이 가지는 색깔을 연속적으로 선택하는 방법을 통해 어깨너머 공격에 강인한 방안이 제시되었다. <그림 14>에서는 비밀번호 “1” 을 보안 키보드를 통해 입력하는 경우가 나타나 있다. 총 4번의 입력을 수행해야 하며 차례로 “1” 이 가지는 색깔인 검은색, 검은색, 흰색, 흰색을 입력하게 된다. 하지만 공격자가 사용자의 비밀번호를 여러 번 관찰하는 경우 해당 색깔의 교집합을 추려냄으로써 비밀번호가 노출되는 문제점이 있다.

[10]에서는 사용자만이 접근 가능한 시각 채널을 통해 보안키보드를 설계함으로써 화면 녹화공격에 안전한 방안이 제시되었다. <그림 15>에서 왼쪽 그림과 같이 3차원으로 보여 지는 일반적인 화면이 있고 오른쪽 그림과 같이 초점이 맞추어진 화면이 스마트폰 상에 나타나도록 한다. 사용자는 화면에서 하나의 숫자가 자신만이 볼 수 있는 특수한 시각화면을 통해 초점이 맞추어지는 것을 확인하고 해당 초점이 맞추어진 숫자 위에 자신이 입력하고자 하는 값을 위치시키는 방식으로 비밀번호를 입력하게 된다. 공격자는 사용자의 특수한 시각을 확인할 수 없으므로 입력되는 비밀번호를 확인할 수 없다.

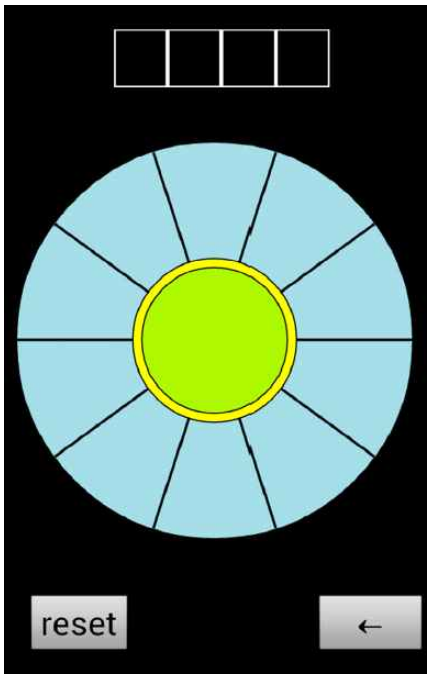
<그림 15> 3DPIN 기반 보안 키보드 [10]

시각적인 채널이 아닌 사용자만이 접근 가능한 음성 채널을 통해 비밀정보에 대한 힌트를 보내고 이를 통해 비밀번호를 도출해 내는 방안이 최근에 제시되고 있다. 음성을 통한 접근 방법은 <그림 16>에 제시된 Apple의 Airpod와 같이 사용자가 언제 어디서나 자신만의 음성 채널을 가질 수 환경이 조성됨에 따라 적합한 방법론으로 생각해 볼 수 있다. 먼저 음성 채널을 전화기의 원형 입력 방식에 적용한 기법은 [11]에서 소개되었다. <그림 17>에서와 같이 사용자는 화면에서 원형 입력 판을 확인할 수 있고 입력 판을 누르면 무작위로 하나의 번호가 선택되어 음성 채널로 전달되도록 한다. 사용자가 원형 입력 판을 이동하면 이에 따라 번호가 순차적으로 증가 혹은 감소하는 방식으로 음성 채널로 번호 값이 발음된다. 만약 사용자가 입력을 원하는 값이 음성으로 전달될 경우 해당 값을 선택하기 위해 사용자는 원형의 중앙으로 드래그를 수행하여 해당 값을 선택하게 된다. [12, 13]에서는 <그림 18>에서와 같이 드래그 버튼을 특정한 위치로 움직이는 동안에 사용자만이 알 수 있는 효과음을 주거나 햅틱 패턴을 주어 사

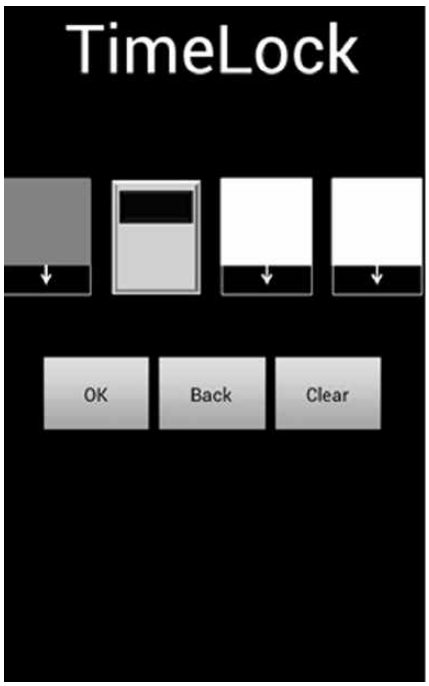
용자가 현재 입력되는 비밀번호의 값을 확인하는 것이 가능하도록 하였다. 최근에 제시된 보안키보드 방안에서는 최소한의 비밀 정보에 대한 힌트만을 음성으로 전달함으로써 사용 편의성을 높임과 동시에 보안성도 확보하였다 [13]. 자세한 동작화면은 <그림 19>에 나타나 있다. 보안키보드 시작 초기에 "A" 라는 비밀정보에 대한 힌트가 전달된다. 만약 사용자의 비밀번호가 "8" 인 경우 사용자는 왼쪽 혹은 오른쪽 화살표를 조작하여 "A" 라는 비밀정보 위에 자신이 입력하고자 하는 비밀 번호 "8" 을 위치한 후 입력을 수행하는 기법을 제시하였다. 이는 공격자가 사용자의 비밀 정보에 대한 힌트를 확인할 수 없기 때문에 공격자의 녹화 공격에도 안전하다.

<그림 16> Apple Airpod

<그림 17> Phone Lock [11]



<그림 18> mTimeLock [12, 13]

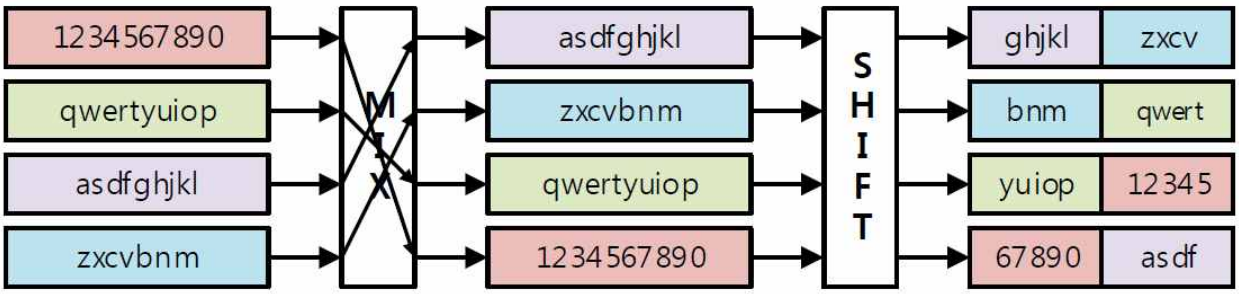


<그림 19> LinA [13]

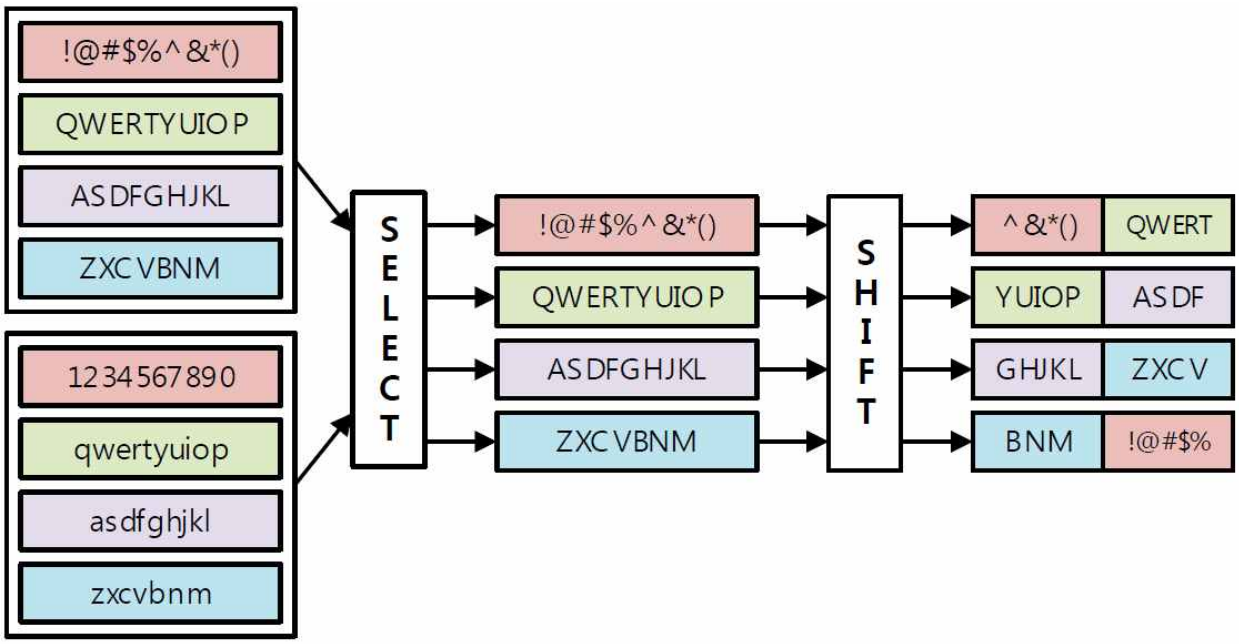
위에서 살펴본 바와 같이 공격자의 녹화공격을 효율적으로 방어하는 다양한 보안키보드가 제시되었다. 하지만 해당 보안키보드는 10개의 숫자를 가지는 숫자키보드에만 적합한 기술 일뿐 일반적인 QWERTY 키보드와 같이 입력이 많은 경우에는 적용이 어렵다. 특히 숫자 키보드만을 사용할 경우 보안성 강화를 위해 대·소문자 그리고 특수

문자까지 요구되는 현 비밀번호 체계상에서는 안전한 입력이 불가능하다는 문제점을 가진다. 따라서 본 논문에서는 QWERTY 키보드와 같이 많은 입력 값이 필요로 되는 환경에서도 안전한 보안 키보드를 제시한다.

<그림 20> 행 기반 보안키보드 생성 방법 [2]



<그림 21> 개선된 행 기반 보안키보드 생성 방법



Ⅲ. 제안하는 보안 키보드

1. 개선된 행 기반 보안키보드

원거리 상에서 사용자의 입력 장면을 녹화하여 사용자의 비밀번호를 알아내는 방법은 [2]에서와 같이 행 기반으로 키보드의 레이아웃을 재배열하는 방법을 통해 효율적으로 방어가 가능하다. 하지만 해당 보안 키보드는 여전히 대·소문자 입력을 안전하게 수행할 수 없을 뿐 아니라 사용자의 입력 편의성 향상이 가능한 부분이 아직 존재한다. 따라서 본 논문에서는 이전의 행 기반 보안 키보드를 보다 안전하고 편리하게 개선한 행 기반 보안 키보드를 제시한다. 제안하는 보안 키보드에 앞서 이전 행 기반 보안 키보드의 동작 방식을 자세히 살펴보면 <그림 20>과 같다. 프로그램이 실행되면

QWERTY 키보드의 레이아웃은 4 개의 행정보로 나누어 표기한다. 4개의 행들은 무작위로 순서가 변경된다. 마지막으로 행의 시작 위치를 무작위로 선택하여 행들이 이동된 형태로 배치되도록 한다. 해당 레이아웃은 하나의 입력이 이루어 질 때마다 위의 과정이 다시 수행되도록 하여 사용자의 비밀 정보 노출이 없도록 하였다. 하지만 해당 기법은 대・소문자에 대한 입력 시 사용자가 SHIFT 버튼과 같은 변환 버튼을 눌러야 하기 때문에 사용자의 비밀 정보의 일부가 노출되는 문제점을 가지고 있다. 현재 비밀번호의 보안성을 높이기 위해 비밀번호 생성 시 대・소문자 그리고 특수 문자를 비밀번호에 포함하도록 권고하고 있기 때문에 해당 입력 값들이 안전하게 입력될 수 있는 방안이 검토 되어야 한다. 본 논문에서 제안하는 보안 키보드는 대・소문자 및 특수문자 입력 시에도 안전하게 입력이 가능할 뿐 아니라 기존의 레이아웃 변경을 보다 간략화하여 사용자의 입력 속도 또한 향상시켰다. 개선된 행 기반 보안키보드의 생성 방법은 <그림 21>와 같다. 기존 행 기반 보안 키보드와 다르게 초기 생성 시에 키보드의 대・소문자가 무작위로 선택되게 된다. 그림에 나타난 예시에서는 대문자가 생성되었다고 가정한다. 두 번째로 키보드는 무작위로 이동되어 나타나게 된다. 해당 예시에서는 5만큼 왼쪽으로 이동된 형식으로 하였다. 해당 이동 범위는 0에서부터 모든 문자의 개수 (36)와 동일하도록 하여 모든 입력 위치에 모든 입력 값이 무작위로 위치할 수 있도록 설계되었다. 이는 기존의 기법과 비교하여 행간의 연관성을 보존한 방안이라고 할 수 있다. 이는 이전 기법의 첫 번째 단계에서 행간의 연관성을 없애기 위해 행들을 무작위로 배치함으로써 사용 편의성을 없앤 것과 비교해 볼 때 사용자가 자신이 원하는 입력값을 보다 쉽게 찾아갈 수 있도록 행간의 정보가 보존된 방안이다.

<표 1> Algorithm 1. 개선된 행 기반 보안 키보드 생성 방법

Input: Random seed Output: Randomized keyboard	
1	Random number is generated from random seed
2	If (Random number % 2) is 0
3	Layout is set to lower character
4	else
5	Layout is set to higher character
6	Random number is generated from random seed
7	Offset is (random number % 36)
8	Layout is shifted by offset
9	Return Layout

<표 1>에서는 개선된 행 기반 보안 키보드 생성 방법을 알고리즘 형식으로 나타내고 있다. 해당 알고리즘은 입력으로 난수 생성 값을 받으며 출력으로 보안성이 강화된 보안 키보드를 생성하게 된다. 처음에 프로그램이 시작되면 난수 생성 값을 통해 난수 값을 생성하게 된다. 해당 난수 값은 2로 나눈 이후에 나머지 값이 짝수인 경우에는

소문자로 레이아웃이 결정되게 되며 해당 값이 홀수인 경우에는 대문자로 레이아웃이 결정되게 된다. 그 다음에는 새로운 난수 값을 난수 생성 값을 통해 다시 생성하게 된다. 해당 난수 값은 36으로 나눈 이후에 나머지 값을 레이아웃 이동범위 값으로 설정하게 된다. 여기서 36이라는 값은 QWERTY 키보드에서 입력이 가능한 문자의 수를 의미하며 만약 더 많은 문자에 대한 입력이 필요한 경우에는 해당 값을 증가시켜서 사용하면 동일한 효과를 얻을 수 있다. 해당 이동범위 값은 주어진 레이아웃 전체를 이동시킬 횟수를 의미한다. 최종 보안 키보드는 해당 이동범위 값만큼 레이아웃이 오른쪽으로 이동되어서 나타나게 된다. 만약 하단의 가장 오른쪽에 위치한 값들 중 더 이상 오른쪽으로 이동할 수 없는 경우에는 왼쪽 상단으로 그 값이 이동되어 나타나게 된다.

<표 2> Algorithm 2. 개선된 행 기반 보안 키보드의 전체 동작 방법

Input: Touch event Output: Password	
1	Random seed is generated from entropy source
2	Password is initialized
3	Run Algorithm 1
4	While (!confirmation)
5	If(Touch event is character)
6	Character is added to the password
7	Run Algorithm 1
8	else if (Touch event is delete button)
9	Last password character is removed
10	else if (Touch event is convert button)
11	Character type is toggled
12	Return Password

<표 2>에서는 개선된 행 기반 보안 키보드의 전체 동작 방법이 제시되어 있다. 보안 키보드를 실행하면 초기에 난수 초기 값을 엔트로피 정보로부터 추출하여 세팅하게 되며 비밀번호를 저장하는 공간 또한 초기화되어 나타낸다. 세 번째로는 현재 키보드 레이아웃을 알고리즘1을 이용하여 새롭게 재정의하게 된다. 재정의를 위해서는 난수 생성 값이 알고리즘1에 사용되게 된다. 보안 키보드 레이아웃이 생성되고 난 이후에는 사용자의 비밀번호를 입력하게 된다. 해당 입력과정은 사용자가 비밀번호 입력을 하고 난 이후에 비밀번호 입력을 마무리한다는 확인 명령어를 누르기 전까지 계속된다. 만약 사용자가 특정한 문자 입력을 시도했다면 해당 문자는 비밀번호에 추가되게 되고 현재의 키보드 레이아웃은 알고리즘1을 통해 재정의되어 나타나게 된다. 이는 보안 키보드 레이아웃도 동일한 레이아웃을 반복적으로 사용하게 될 시에 보안 정보가 노출됨을 방지하기 위해서이다. 만약 사용자가 현재 입력된 비밀번호를 수정하고 싶은 경우에는 삭제 버튼을 이용하여 가장 마지막에 입력된 문자부터 차례대로 삭제 후 재입력하는 과정을 거치게 된다. 만약 사용자가 대・소문자를 변환하기 위해 변환 버튼을 누

르게 될 경우에는 현재의 레이아웃이 대문자인 경우에는 소문자로 소문자인 경우에는 대문자로 변환되어 나타나게 된다. 이때에는 레이아웃의 위치정보는 그대로 유지되도록 하여 사용자가 레이아웃을 재학습해야하는 불편함이 없도록 하였다. 마지막으로 사용자가 현재 입력된 비밀번호 값을 결정하고 싶은 경우에는 확인 버튼을 눌러서 비밀번호 입력을 마무리하게 된다.

<그림 22>에서는 개선된 행 기반 보안키보드의 실제 동작화면이 나타나 있다. 해당 실험에서는 대・소문자, 숫자, 특수문자를 모두 포함하는 임의의 문자열 "!1Qq"를 입력하는 방안을 나타내고 있다. 해당 보안 키보드 레이아웃의 구성은 일반적인 QWERTY 키보드 레이아웃과 더불어 삭제 (DEL) 버튼, 전환 (CONVERT) 버튼으로 구성된다. 해당 레이아웃의 배치는 사용자의 편의에 따라 변경이 가능하다. 보안 키보드의 초기화면은 무작위로 설정된 대문자 레이아웃으로 설정되어 있으며 특수문자 '!'를 입력하기 위해서는 세 번째 줄에서 왼쪽 두 번째 버튼을 눌러 입력하는 것이 가능하다. 해당 입력은 공격자로 하여금 현재 입력되고 있는 문자가 대문자인지 소문자인지 확인할 수 없게 하는 방안이다. 두 번째 입력화면에서는 대문자로 키보드 레이아웃이 설정되어 있다. 하지만 숫자를 입력하기 위해서는 키보드 레이아웃이 소문자인 경우에만 가능하므로 대문자에서 소문자로 레이아웃을 변경하기 위해 전환(CONVERT) 버튼을 입력하여 대・소문자를 전환해주게 된다. 소문자로 전환된 세 번째 입력화면에서는 첫 번째 줄에서 다섯 번째에 위치한 '1'을 선택하여 입력을 하게 된다. 네 번째 입력화면에서는 키보드 레이아웃이 초기에 대문자로 세팅되어 있다. 이 경우 대문자 'Q'는 레이아웃 변환과정이 필요없이 바로 네 번째 줄의 두 번째 버튼을 선택함으로써 입력이 가능하다. 다섯 번째 입력화면에서는 키보드가 소문자로 세팅되어 있다. 따라서 입력하고자 하는 소문자 'q'는 네 번째 줄의 네 번째 버튼을 누름으로써 바로 입력이 가능하다. 마지막으로 여섯 번째 입력화면에서는 "!1Qq" 문자열이 모두 입력되고 난 후의 최종화면이 보이게 된다. 매 문자 입력 시에는 입력이 마지막으로 이루어진 문자가 사용자에게 보이도록 하여 사용자가 자신이 입력한 값에 대한 피드백을 원활히 받을 수 있도록 하였다. 특히 매 입력화면에서 확인할 수 있듯이 대・소문자 레이아웃에 대한 설정이 무작위로 결정되도록 하여 공격자가 사용자가 현재 입력하는 값이 대문자인지 소문자인지 여부를 확인하는 것이 불가능하도록 되어 있다. 입력 시 사용자는 원하는 대・소문자 레이아웃 세팅이 아닐 경우 하단의 전환(CONVERT) 버튼을 이용하여 자신이 원하는 대・소문자 형식으로 변환한 다음 사용하면 된다. 사용자가 전환 버튼을 누르는 경우와 누르지 않는 경우 모두 초기에 대・소문자가 무작위로 선택되기 때문에 공격자는 키보드 레이아웃의 상태를 확인하는 것이 불가능하다. 또한 사용 편의성을 높이기 위해 대・소문자 전환 시에는 레이아웃의 이동정보를 이전 정보 그대로 유지하도록 하여 사용자가 원하는 문자를 선택하기 위해 레이아웃을 재학습해야 하는 문제점이 발생하지 않도록 설계되었다. 매번 입력 값의 위치가 이동범위 값에 따라 이동된 이후에 나타나도록 설계됨으로써 공격자가 사용자의 입력위치를 확인하고

도 현재 입력된 문자를 유추하는 것이 불가능하도록 했다. 또한 사용자는 기존의 QWERTY 키보드 레이아웃과 매우 유사한 키보드 레이아웃 상에서 원하는 값을 쉽게 찾을 수 있다.

<그림 22> 개선된 행 기반 보안키보드의 실행 화면 (“!1Qq” 입력 테스트), (a) ‘!’ 입력, (b) 대・소문자 전환, (c) ‘1’ 입력, (d) ‘Q’ 입력, (e) ‘q’ 입력, (f) 입력 완료, 빨간 네모로 표시된 부분은 현재 입력되는 부분

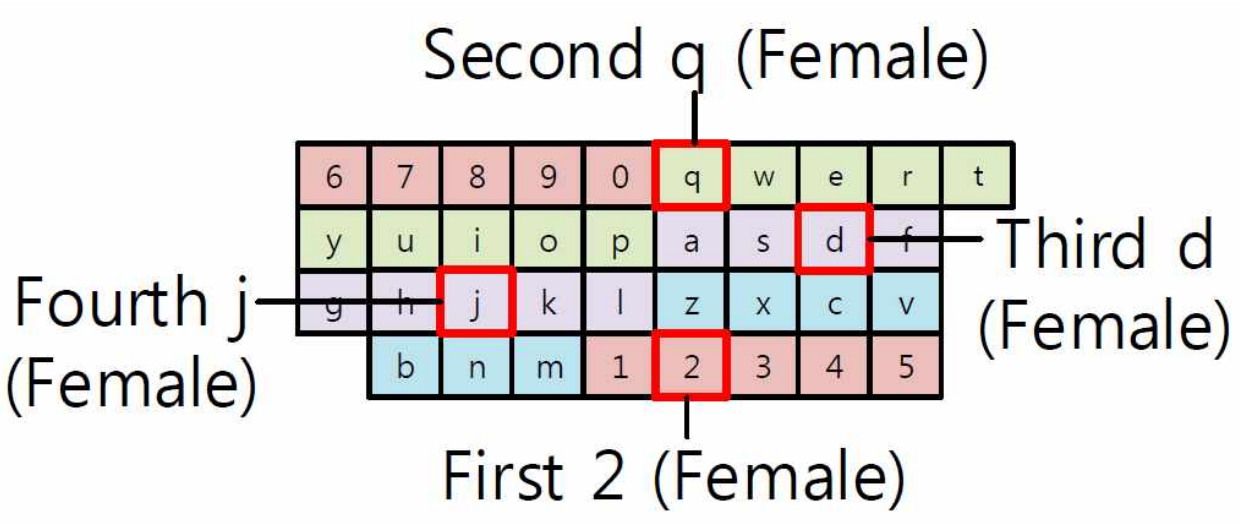
2. 녹화 공격에 안전한 보안키보드

이전 장에서는 공격자가 원거리에서 구글글라스 혹은 소형 카메라를 통해 사용자의 입력 동작을 확인하는 경우에도 PIN값을 안전하고 편리하게 입력하는 보안 키보드에 대해 살펴보았다. 하지만 고도로 발전하고 있는 사물인터넷 환경에서는 디바이스가 점차 소형화 및 스마트폰 해킹기술의 발전에 따라 사용자가 입력하고 있는 화면이 공격자에 의해 노출될 수 있는 위험이 있다. 이는 기존의 보안키보드 상에서 사용자의 입력화면은 안전하게 보호된다는 가정이 지켜지지 않는 경우로써 많은 보안키보드 상에서 취약점이 발생하게 된다. 따라서 공격자에 의해 PIN 입력화면이 노출되는 경우에도 안전한 보안 키보드의 개발이 사물인터넷 시대에는 요구되고 있다. 본 장에서는 공격자에 의해 사용자의 모든 행동이 관찰되고 비밀번호 입력 화면이 노출되는 경우에도 안전하게 값을 입력할 수 있는 보안키보드에 대해 확인해 보도록 한다.

녹화 공격에 안전한 보안키보드는 이전 장에서 제안한 개선된 행 기반 보안키보드를 기본 동작으로 사용하지만 보안정보 힌트를 안전하게 전달하기 위해 음성 채널을 추가로 사용하게 된다. 이전 보안키보드와는 달리 화면상에는 백지의 QWERTY 키보드 레이아웃이 나타나게 된다. 사용자가 하나의 버튼을 누르는 경우 해당 버튼에 상응하는 결과 값이 음성 채널을 통해 전달된다. 하지만 QWERTY 키보드는 입력 값 36개로써 숫자만을 사용하는 키보드에 비해 월등히 많을 뿐 아니라 문자정보가 표기되지 않은 백지 키보드를 사용하기 때문에 사용자 입장에서는 자신이 원하는 입력 값을 찾아가는 것이 매우 어렵다. 이를 효율적으로 보완하기 위해 두 가지 입력 효율성 제고 방안을 제시한다. 첫 번째로 해당 입력이 대・소문자인지 효율적으로 확인하는 방안이 필요하다. 이전 보안 키보드에서는 사용자가 시각을 통해 현재 세팅을 확인하는 것이 가능했다. 하지만 현재 보안 키보드는 백지의 레이아웃을 사용하며 시각적인 정보는 공격자에게 모두 노출된다는 문제점을 가지고 있다. 따라서 음성정보 전달시 사용자만이 알 수 있는 방안을 통해 대・소문자 세팅을 전달하는 방안을 제시한다. 가장 쉽게 생각할 수 있는 방안은 현재 입력되는 문자가 대문자인 경우 음성을 통해 대문자를 발음하는 것이다. 하지만 사용자는 대문자라는 음성을 듣는 동안 시간 소비해야하는 문제점이 있다. 이를 효과적으로 해결하기 위해 음성발음 시 대문자인 경우에는 남성의 음성으로 소문자인 경우에는 여성의 목소리로 발음이 나도록 구성하였다. 이는 사용자가 보다 직관적으로 현재의 입력 상태를 확인할 수 있을 뿐 아니라 입력 시간을 효과적으로 단축시킬 수 있다. 두 번째로는 해당 입력문자를 찾아가는 어려움이다. 36개의 문자가 무작위로 배치된 경우 최악의 경우 36번의 시도를 거쳐야 원하는 값을 도출할 수 있는 문제점을 가지고 있다. 이를 해결하기 위해 발음하는 입력 값 전에 해당 입력 값이 QWERTY 키보드에서 위치하는 행의 순서를 발음해 줌으로써 사용자는 보다 편하게 자신이 원하는 값을 찾아갈 수 있다. 즉 해당 기법은 QWERTY 키보드 상에서 자신이 입력하고자 하는 값의 행을 선택한 이 후에 소속된 문자를 찾아가는 방안을 제시함으로써 사용자는 기존의 1단계 대신 2단계를 통해 편하게 문자를 찾아가는 것이 가능하

다. <그림 23>에는 4가지 입력 값에 대한 음성 정보 예시를 나타내고 있다. 현재 레이아웃은 소문자로 세팅되어 있으며 총 4개의 열이 이동범위 값 5만큼 이동되어 나타나 있다. 가장 윗줄의 소문자 'q' 는 여성의 목소리로 "Second q" 가 발음되게 된다. 이는 해당 입력 값이 전형적인 QWERTY 키보드의 두 번째 행에 위치하기 때문이다. 마찬가지로 두 번째 줄의 소문자 'd' 는 여성의 목소리로 "Third q" 가 발음되며 세 번째 줄의 소문자 'j' 는 여성의 목소리로 "Fourth j" , 그리고 마지막으로 네 번째 줄의 소문자 '2' 는 여성의 목소리로 "First 2" 로 발음되게 된다. 만약 현재 레이아웃이 대문자인 경우에는 여성의 음성대신 남성의 목소리로 입력 값이 발음되게 된다.

<그림 23> 녹화 공격에 안전한 보안키보드의 동작 상세



<표 3> Algorithm 3. 음성 재생 방법

Input: Touched character, previous row, character setting Output: Vocalization	
1	If(Previous row is equal o touched character's row)
2	If(Character setting is upper character)
3	Touched character is vocalized in male voice
4	else
5	Touched character is vocalized in female voice
6	else
7	If(Character setting is upper character)
8	Row number and touched character is vocalized in male voice
9	else
10	Row number and touched character is vocalized in female voice
11	Previous row is set to touched character's row

<표 3>에는 녹화 공격에 안전한 보안 키보드를 위한 음성 재생 방법이 나타나 있다. 사용자의 터치 이벤트가 발생하면 현재 터치된 문자와 이전 문자행의 위치 그리고 현

재 문자의 대・소문자 세팅정보가 알고리즘의 입력으로 사용되게 된다. 처음에는 현재 문자의 행이 이전 문자행과 동일한지 확인하게 된다. 이는 문자의 행이 동일한 경우에는 문자의 행을 발음하지 않음으로써 음성이 재생되는 시간을 줄이는 목적으로 사용된다. 현재의 문자 행 세팅이 대문자인 경우에는 남성 목소리로 현재의 문자가 발음되게 된다. 반대로 소문자인 경우에는 여성 목소리로 현재 문자가 발음되게 된다. 만약 이전 행과 현재 행이 동일하지 않은 경우에는 행 번호와 문자 정보가 발음되게 된다. 해당 발음 알고리즘을 종료하기 전에는 이전 행 정보에 현재 행 정보를 저장함으로써 다음에 들어오는 문자의 행 정보와의 비교에 사용되게 된다.

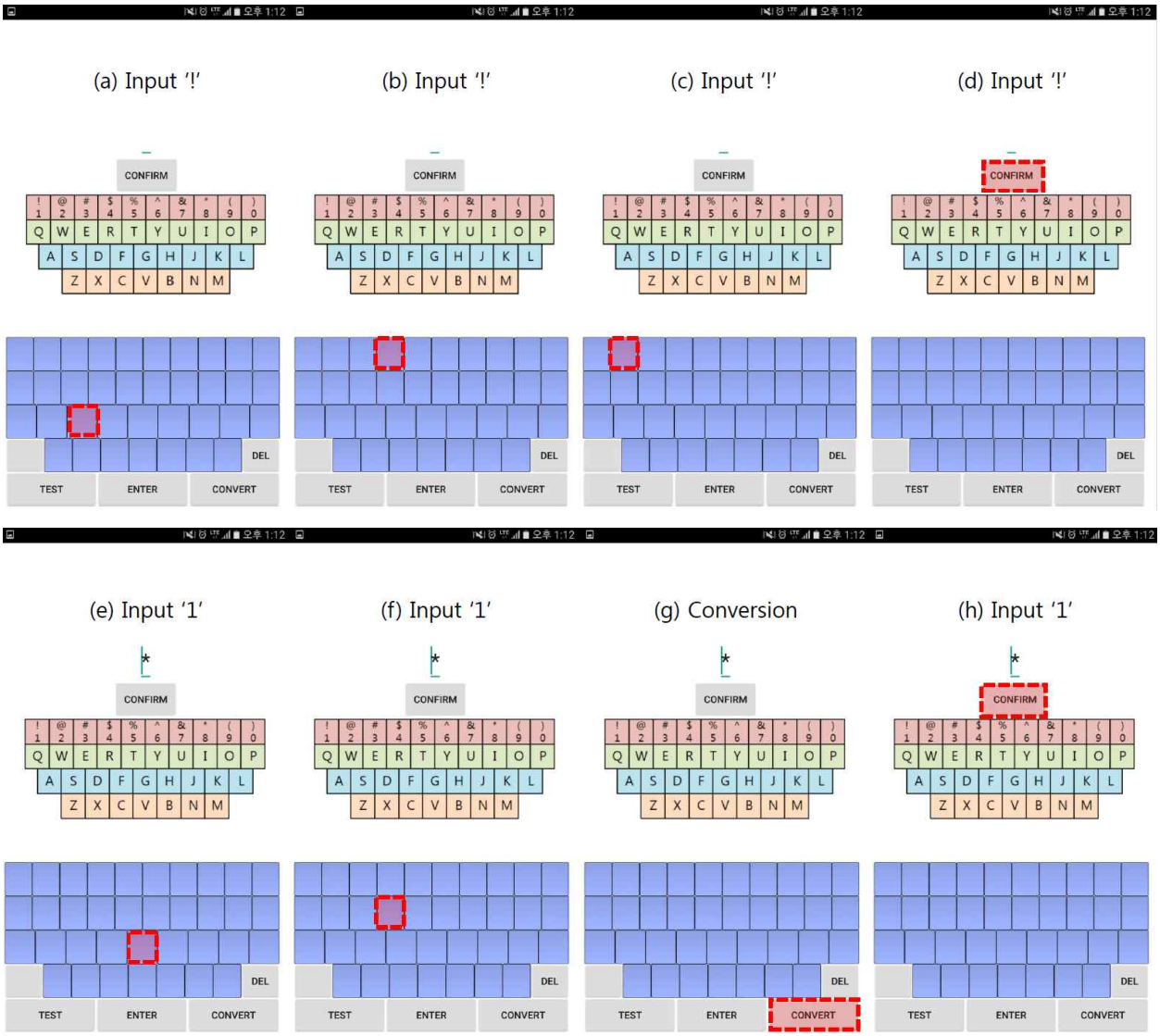
<표 4> Algorithm 4. 녹화 공격에 안전한 보안 키보드의 전체 동작 방법

Input: Touch event Output: Password	
1	Random seed is generated from entropy source
2	Password is initialized
3	Run Algorithm 1
4	While (!confirmation)
5	If(Touch event is character)
6	Run Algorithm 3
7	If(Character is selected)
8	Character is added to the password
9	Run Algorithm 1
10	else if (Touch event is delete button)
11	Last password character is removed
12	else if (Touch event is convert button)
13	Character type is toggled
14	Return Password

<표 4>에서는 녹화 공격에 안전한 보안 키보드의 전체 동작 방법이 제시되어 있다. 보안키보드를 실행하면 초기에 난수 초기 값을 엔트로피 정보로부터 추출하여 세팅되며 비밀번호를 저장하는 공간 또한 초기화되어 나타낸다. 그 이후에는 키보드 레이아웃 정보가 알고리즘1을 통해 설정되게 된다. 초기 설정이 마무리되고 난 이후에는 비밀번호 입력이 수행되게 된다. 해당 입력과정은 사용자가 비밀번호 입력을 마무리한다는 확인 명령어를 누르기 전까지 계속된다. 만약 사용자가 특정한 문자를 눌러 터치 이벤트를 발생시켰다면 알고리즘3을 실행시켜 현재 값에 대한 음성 정보를 확인하게 된다. 만약 해당 문자가 입력을 원하는 문자인 경우에는 해당 문자를 선택하게 된다. 선택된 문자는 비밀번호에 추가되게 되고 현재의 키보드 레이아웃은 알고리즘1을 통해 새롭게 정의되게 된다. 만약 사용자가 현재의 비밀번호를 수정하고 싶으면 삭제 버튼을 이용하여 가장 마지막에 입력된 문자부터 차례대로 삭제 후 수정이 가능하다. 만약 사용자가 변환 버튼을 누르게 될 경우에는 현재의 레이아웃이 대문자인 경우에는 소문

자로 소문자인 경우에는 대문자로 변환되어 나타나게 된다. 해당 변환 결과는 시각적으로는 확인이 불가능하지만 음성을 통해서는 확인이 가능하다. 마지막으로 현재 비밀번호가 정확하여 해당 비밀번호를 입력하고 싶은 경우 확인 버튼을 눌러서 현재 비밀번호를 입력하게 된다.

<그림 24> 녹화공격에 안전한 보안키보드의 실행 화면 (“!1” 입력 테스트), (a) ‘!’ 입력, (b) ‘!’ 입력, (c) ‘!’ 입력, (d) ‘!’ 입력, (e) ‘1’ 입력, (f) ‘1’ 입력, (g) 대・소문자 전환, (h) ‘1’ 입력



<그림 24>에서는 녹화공격에 안전한 보안키보드의 실행화면이 나타나 있다. 화면의 구성은 입력창, 참고용 QWERTY 키보드 레이아웃, 입력 키보드, 삭제 (DEL) 버튼, 선택된 문자 입력 (ENTER) 버튼 그리고 전환 (CONVERT) 버튼으로 구성된다. 해당 실험에서는 임의의 숫자와 특수문자로 구성된 문자열 “!1” 를 입력하는 방안에 대해 설명하고 있다. 사용자가 입력을 시작하면 백지의 보안키보드 상에서 어디에 ‘!’ 라는 입

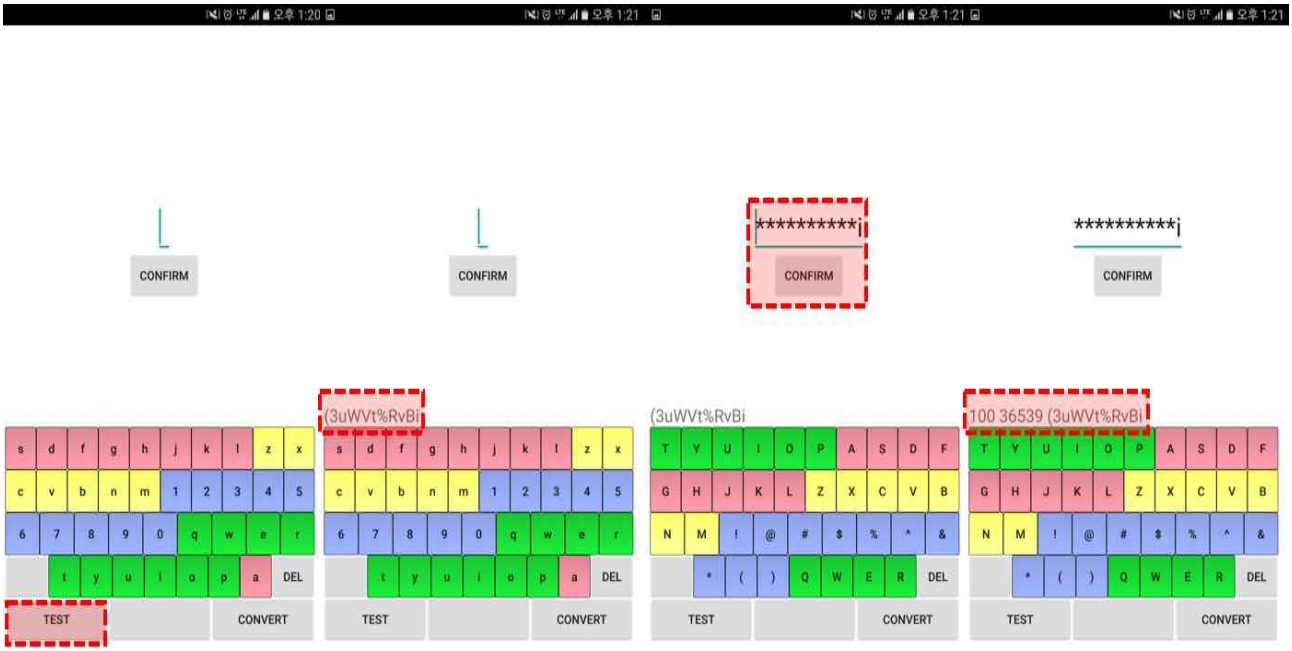
력이 위치하는 지 확인할 수 없으므로 아무 버튼이나 눌러서 현재 키보드 레이아웃의 정보를 확인하게 된다. <그림 24>의 (a)에서는 세 번째 줄의 세 번째 칸을 선택하게 되고 해당 입력에 대한 출력으로 남성의 목소리로 “Third S” 라는 음성을 듣게 된다. 이는 현재 입력이 대문자 ‘S’ 이며 해당 입력이 세 번째 행임을 알려주게 된다. 사용자가 입력하고자하는 값인 ‘!’ 의 경우 첫 번째 행에 위치함으로 사용자가 선택한 행에서부터 대략 2줄 위에 원하는 값이 위치하게 된다. <그림 24>의 (b)에서는 이전 입력으로부터 2줄 위인 1번째 행의 4번째 버튼을 선택하게 된다. 해당 버튼은 남성음성으로 “First 3” 이 발음되게 되며 현재 값은 ‘#’ 임을 확인할 수 있다. 따라서 <그림 24>의 (c)에서는 원하는 문자 ‘!’ 는 왼쪽 2칸 옆에 위치한다는 것을 확인할 수 있다. 음성정보를 통해 해당 버튼이 원하는 문자가 맞는 경우 <그림 24>의 (d)에서와 같이 확인(CONFIRM) 버튼을 눌러 해당 값을 비밀번호 입력에 추가하게 된다. 그 다음으로 입력값 ‘1’ 을 입력하기 위해서 <그림 24>의 (e)에서와 같이 다시 임의의 버튼을 눌러보게 된다. 그 이유는 첫 번째 입력이 끝나고 나면 키보드 레이아웃은 새롭게 설정되므로 이전의 레이아웃 정보는 사용할 수 없기 때문이다. 만약 해당 값이 남성의 목소리로 “Second W” 를 발음하게 되는 경우에는 현재 레이아웃이 대문자로 세팅되어 있으며 두 번째 행임을 확인할 수 있다. 따라서 원하는 입력 값 ‘1’ 을 선택하기 위해 <그림 24>의 (f)에서와 같이 왼쪽으로 11칸 이동된 버튼을 눌러서 입력하게 된다. 해당 버튼은 남성의 목소리로 “First 1” 이라고 발음하게 되며 이는 사용자가 원하는 값의 위치가 된다. 하지만 현재 세팅이 대문자로 되어 있으므로 이를 소문자로 변경하기 위해 <그림 24>의 (g)에서와 같이 왼쪽 하단의 변환 (CONVERT) 버튼을 선택하게 된다. 마지막으로 해당 값을 입력하기 위해서 <그림 24>의 (h)에서와 같이 확인 (CONFIRM) 버튼을 선택하게 된다.

IV. 실험 및 평가

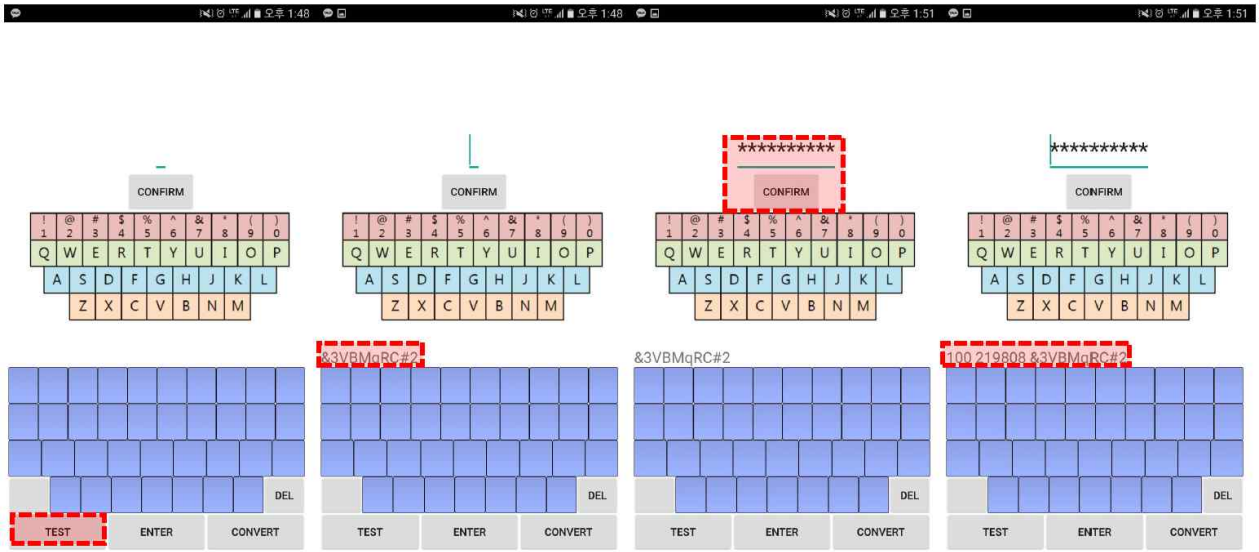
본 장에서는 본 논문에서 제안하는 두 보안 키보드 기법을 안드로이드 스마트 폰 상에서 구현한 결과물을 직접 테스트해 봄으로써 제안하는 기법의 보안성과 신속성 그리고 정확도에 대해 비교 분석해 보도록 한다. 개선된 행 기반 보안키보드 테스트를 위한 안드로이드 스마트폰의 환경은 <그림 25>과 같다. 스마트폰 어플리케이션을 실행하면 초기화면에는 입력칸이 공백으로 설정되어 있다. 왼쪽 하단의 “TEST” 라는 버튼을 누르게 되면 화면 중간에 테스트를 위한 10자리 문자가 무작위로 생성되게 된다. 해당 무작위 문자값은 전체 입력이 가능한 값 중에서 무작위로 10개의 문자를 뽑아서 하나의 문자열을 만드는 방식으로 테스트마다 매번 새롭게 생성되도록 하였다. 실험자는 해당 무작위 문자열을 보안키보드 동작 방식에 따라 동일한 값을 입력하게 되며 해당 값이 입력될 때마다 입력칸에는 이전 입력값들은 ‘*’ 형식으로 변환되어 나타나게 되고 마지막 입력값만 입력칸에 나타나 해당 입력값이 입력되었음을 확인할 수 있도록 한다. 모든 테스트가 끝나고 나면 중간의 확인 버튼을 눌러서 해당값을 비밀번호로 결

정하게 된다. 마지막으로는 해당 입력값에 대한 정확도와 속도 그리고 현재 입력된 비밀번호가 차례로 나타나게 된다. 해당 값들은 수집 및 분석되어 해당 장에서 결과도출에 사용되게 된다.

<그림 25> 개선된 행 기반 보안키보드 테스트 환경



<그림 26> 녹화공격에 안전한 보안키보드 테스트 환경

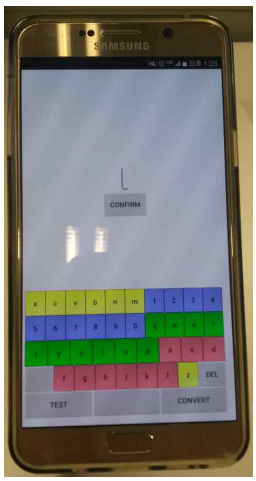


녹화공격에 안전한 보안키보드 테스트를 위한 안드로이드 스마트폰의 환경은 <그림 26>과 같다. 스마트폰 어플리케이션을 실행하면 초기화면에는 입력칸이 공백으로 설정되게 되며 입력 참고를 위한 기본적인 QWERTY 키보드가 화면 중앙에 위치하게 된다. 왼쪽 하단의 “TEST” 라는 버튼을 누르게 되면 화면 중간에 테스트를 위한 10자리 무

작위 값이 생성되게 된다. 해당 무작위 문자 값은 전체 입력이 가능한 값 중에서 무작위로 10개의 경우를 뽑아서 설정된다. 실험자는 해당 무작위 문자열을 보안키보드 동작 방식에 따라 값을 입력하게 되며 해당 값이 입력될 때마다 입력칸에는 '*' 가 하나씩 채워져서 해당 값이 입력되었음을 확인할 수 있도록 한다. 모든 테스트가 끝나고 나면 중간의 확인 버튼을 눌러서 해당 값을 비밀번호로 최종 입력하게 된다. 입력이 끝나고 나면 해당 입력 값에 대한 정확도와 속도 그리고 현재 입력된 비밀번호가 차례로 나타나게 된다.

본 실험에 사용된 스마트 폰은 <그림 27>와 같이 갤럭시 노트5 스마트폰으로 선정하였다. 해당 스마트폰은 최신 스마트폰으로써 많은 국민들이 사용하고 있어 객관적인 지표 도출에 적합하다고 생각되어 선택하게 되었다. 갤럭시 노트5에 대한 상세 성능은 <표 5>와 같다.

<그림 27> 타겟 장비 테스트 화면



<표 5> 타겟 장비에 대한 상세 설명

디바이스 이름	갤럭시 노트5
해상도	1440 x 2560
디스플레이	슈퍼 아몰레드 5.7 인치
버튼크기(cm)	0.7 x 0.8 (가로x세로)
CPU	Octa-core (4 x 2.1 GHz Cortex-A57 & 4 x 1.5 GHz Cortex-A53)
GPU	Mali-T760MP8
MEMORY	32 GB, 4 GB RAM
OS	Android OS

타겟 플랫폼 상에서의 테스트를 위해 프로그램은 안드로이드 어플리케이션 형식으로 개발되었다. 프로그램에서 사용된 난수값은 Java 라이브러리의 java.util.Random 패키지를 이용하여 매번 새로운 난수값을 생성하도록 하였다. 두 번째 보안 키보드 상에서의 음성 생성을 위해 온라인상에서 음성을 mp3 파일 형식으로 제공해 주는 사이트 상에서 모든 음성파일을 추출하였다 [14]. 남성 음성의 경우 미국억양을 가진 남성이 보통 속도로 발음해 주도록 하였다. 여성 음성의 경우 미국억양을 가진 여성이 보통 속도로 발음해 주도록 하였다. 해당 음성 파일은 안드로이드 프로젝트에 추가된 이후에 android.media.MediaPlayer와 android.media.AudioManager 패키지를 통해 정확한 발음이 스마트폰을 통해 재생되도록 하였다. 본 실험에 참가한 실험자들은 건장한 체격에 시력과 청력에 문제가 없으며 기존에 스마트폰을 사용하여 소프트 버튼을 능숙히 사용할 수 있다. 실험에는 총 5명의 실험자가 참가하였다. 테스트한 종목은 본 논문에서 제안

하는 2개의 보안키보드로 선정하였다. 기존의 보안키보드의 성능은 이전 논문의 결과 값을 참고하였다 [2].

<표 6> QWERTY 키보드 간 비교 분석, 입력속도는 10자를 입력할 때의 속도 측정, (ver 1)의 경우 대・소문자 지원하지 않음, (ver 2)의 경우 대・소문자 지원

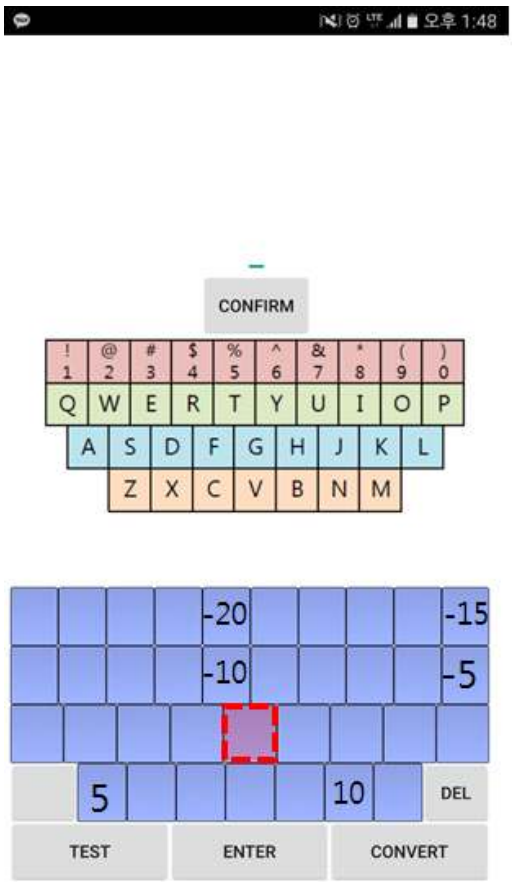
	Random 보안 키보드 [1]	행 기반 보안 키보드 [2]	개선된 행 기반 보안 키보드 (ver 1)	개선된 행 기반 보안 키보드 (ver 2)	녹화 공격에 안전한 보안 키보드 (ver 1)	녹화 공격에 안전한 보안 키보드 (ver 2)
입력속도 (sec)	28.0	23.6	19.9	25.9	75.6	97.4
정확도 (%)	98	99	99	98	97	96
보안성 (P_{GA})	$\frac{1}{36^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{72^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{72^{10}}$
보안성 (P_{CA})	$\frac{1}{36^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{72^{10}}$	$\frac{1}{36^{10}}$	$\frac{1}{72^{10}}$
보안성 (P_{RA})	1	1	1	1	$\frac{1}{36^{10}}$	$\frac{1}{72^{10}}$

1. 신속성

<표 6>에서는 실험을 통해 도출한 QWERTY 키보드 상에서의 입력속도, 정확도 그리고 보안성을 비교 분석한 결과를 나타내고 있다. 다른 보안 키보드인 [8-13]의 경우에는 숫자키보드만을 목적으로 하였기 때문에 제안하는 키보드와 같이 많은 입력을 가지는 QWERTY 키보드와의 비교가 적합하지 않기 때문에 포함하지 않았다. 신속성 측면에서 비교했을 때 제안하는 개선된 행 기반 보안키보드 (ver 1)이 가장 높은 성능을 나타내었다. 이는 키보드 레이아웃에 대한 변화가 이전 행 기반 보안 키보드에 비해 적어서 사용자가 자신이 원하는 입력 값을 찾아가는 과정에서 시간이 보다 더 적게 소모되기 때문이다. 반면에 녹화 공격에 안전한 보안 키보드 (ver 2)는 가장 느린 성능을 나타내었다. 해당 기법이 가장 느린 이유는 하나의 입력에 대한 음성 피드백이 짧게는 1초 길게는 3초정도가 소모되기 때문이다. 또한 레이아웃의 정보가 표기되지 않아 자신이 원하는 정보를 찾아가기 위해 최소 2번 이상은 입력 시도를 해야 적합한 입력이 가능하기 때문이다. 따라서 해당 보안 키보드의 성능을 향상시키기 위해서는 사용자가 자신이 원하는 입력 값을 빠르게 찾아가기 위한 피드백 정보를 주는 방법이 있다. <그림 28>에서와 같이 사용자가 하나의 버튼을 선택하게 되는 경우 해당 값을 중심으로 이동거리 값을 특정 거리마다 버튼 위에 표기해 줄 수 있다. 해당 기법은 사용자가 자신이 누른 버튼으로부터 대략적인 거리를 확인할 수 있게 해줌으로써 사용자가 원하는

목적 문자와의 거리 계산에 보다 효율적으로 사용이 가능하다. 다른 성능개선 방법으로는 입력 방법 효율화이다. 기존의 버튼을 눌러서 문자를 입력하는 방법 대신 <그림 29>에서와 같이 레이아웃 상에서 버튼을 드래그할 때마다 값을 확인하는 방식이다 [15]. 해당 방식은 사용자가 원하는 목적 값을 클릭이 아닌 드래그를 통해 확인할 수 있게 해줌으로써 사용자가 클릭에 소모하는 입력 시간을 효율적으로 단축시킬 수 있다.

<그림 28>
성능 개선방안 #1



<그림 29>
성능 개선방안 #2 [15]

2. 정확성

<표 6>의 두 번째 비교 항목으로는 보안키보드를 통해 입력된 값의 정확도를 확인하는 것이다. 대부분의 보안 키보드 상에서 96% 이상의 정확도를 나타냄을 확인할 수 있었다. 특히 녹화공격에 안전한 보안 키보드와 같이 레이아웃 정보가 화면상에는 보이지 않는 경우에도 음성 채널을 통해 현재 입력된 값을 보다 정확하게 확인할 수 있음이 확인되었다. 이처럼 제안하는 보안키보드들은 사용자가 원하는 값을 정확하게 입력하는 데에는 전혀 문제가 없음을 확인할 수 있다.

3. 보안성

보안 키보드의 가장 중요한 항목은 보안성이라고 할 수 있다. <표 6>에서는 총 3가지 측면에서 보안성을 확인해 보았다. 첫 번째 보안성 테스트인 P_{GA} (Probability of successful guessing attack)는 성공적으로 비밀번호를 추측할 수 있는 확률을 의미한다. 공격자는 여러 번의 테스트를 통해 정확한 PIN을 추측할 수 있는지 확인하게 된다. 두 번째 보안성 테스트인 P_{CA} (Probability of successful challenge only attack)는 공격자가 여러 번의 시도를 통해 확보한 정보를 통해서 PIN정보를 도출하는 공격이다. 마지막 보안성 테스트는 P_{RA} (Probability of successful recording attack)으로써 공격자가 사용자의 입력화면을 녹화한 경우에 비밀번호를 도출할 수 있는 확률을 의미한다. 대・소문자를 구분하지 않는 QWERTY 키보드의 경우 숫자와 문자를 포함한 36개의 경우가 하나의 입력에서 선택 가능하다. 따라서 공격자가 사용자의 10 자리 비밀번호를 추측하는 경우에는 $\frac{1}{36^{10}}$의 확률로 비밀번호를 확인하는 것이 가능하다. 본 논문에서 제안하는 보안 키보드의 경우 대・소문자를 포함하고 숫자의 변환 값인 특수문자까지 포함한 총 72 개의 문자가 입력 가능하다. 공격자가 10자리 비밀번호를 추측하는 경우에는 $\frac{1}{72^{10}}$의 확률로 비밀번호를 확인할 수 있다. 두 번째 보안성 테스트인 P_{CA}의 경우에는 여러 번의 비밀 번호 입력 시도를 통해 정보를 확인하는 경우에도 비밀번호의 특성이 노출되지 않기 때문에 대・소문자 구분이 없는 경우와 있는 경우 각각 $\frac{1}{36^{10}}$ 그리고 $\frac{1}{72^{10}}$의 확률로 비밀번호를 확인할 수 있다. 마지막으로 P_{RA}은 기존의 보안 QWERTY 키보드는 입력화면이 노출되는 순간 모든 비밀 정보가 공격자에 의해 확인이 가능한 취약점을 가진다. 하지만 녹화공격에 안전한 보안 키보드의 경우에는 안전한 음성 채널을 통해 비밀정보에 대한 힌트가 전달되기 때문에 공격자의 녹화 공격에도 보안성이 유지되는 장점을 가진다. 따라서 제안하는 보안 키보드 상에서 대・소문자 구분이 없는 경우와 있는 경우 각각 $\frac{1}{36^{10}}$ 그리고 $\frac{1}{72^{10}}$의 확률로 비밀번호를 확인할 수 있다.

V. 결론

본 논문에서는 사용자가 온라인 금융 서비스를 이용하기 위해 QWERTY 키보드를 통해 입력하는 PIN 정보가 공격자가 가진 소형화된 사물인터넷 장비를 통해 관찰될 경우 비밀번호가 노출되는 문제점이 있음을 확인하고 이를 방지하기 위한 두 가지 보안 키보드 기법을 제안하였다. 첫 번째 보안키보드는 이전 행 기반 보안 키보드의 문제점이었던 대・소문자 입력 시 노출되는 비밀정보를 제거하고 입력 속도를 향상시키기 위해 기존의 QWERTY 레이아웃 상에서 이동범위 값만 수정하는 방식으로 새로운 행 기반 보안 키보드를 제안하였다. 두 번째 보안키보드는 공격자가 사용자의 시각적인 정보를

모두 접근이 가능한 경우에도 공격으로부터 안전하도록 음성 채널을 통해 보안 정보에 대한 힌트를 전달하는 보안 키보드를 설계하였다. 해당 기법은 음성을 통해 사용자에게 비밀정보에 대한 힌트가 전달되도록 하여 기밀성이 제공한다. 또한 사용자가 입력을 편하게 할 수 있도록 첫 번째로 제안한 개선된 행 기반 보안 키보드의 레이아웃을 사용하였다. 특히 해당 보안 키보드는 이전에 제시된 많은 보안 키보드가 숫자 키보드와 같이 입력 값의 범위가 작은 경우에만 한정된 것과 달리 상에서 실제적으로 많이 사용되는 복잡한 QWERTY 키보드를 안전하게 보호했다는 측면에서 그 활용도가 높다. 본 제안 기법들은 안드로이드 스마트폰 상에서 실제로 구현 및 테스트 되었으며 실험 결과를 통해 기존 기법에 비해 보다 안전하며 사용 용이성이 높음을 확인할 수 있었다.

참고 문헌

[1] Yue, Qinggang, Zhen Ling, Xinwen Fu, Benyuan Liu, Wei Yu, and Wei Zhao. "My Google Glass Sees Your Passwords!." Black Hat USA2014, (2014).

[2] 서화정, 김호원. "입력 위치 유추 방지를 위한 보안 키패드의 설계." Journal of The Korea Institute of Information Security & Cryptology 26.1 (2016).

[3] Diksha Shukla, Rajesh Kumar, Abdul Serwadda, and Vir V. Phoha, "Beware, your hands reveal your secrets!." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, (2014).

[4] 김현진, 서화정, 이연철, 박태환, 김호원, "어께 넘어 훔쳐보기에 저항성을 가진 가상금융키패드의 구현," 정보보호학회지, 23(6), 21-29 (2013).

[5] 서화정, 이연철, 김현진, 박태환, 나엔하반, 석선희, 김경훈, 김호원, "촉각을 이용한 보안키패드의 설계 " 2014년도 지급결제제도 발전 논문 현상공모 아이디어상, (2014).

[6] 서화정, 김호원, "입력 메시지 암호화를 통한 보안 키패드의 설계와 구현," 한국정보통신학회논문지, 18(12), 2899-2910. (2014).

[7] Brendan Saltaformaggio, Rohit Bhatia, Zhongshu Gu, Xiangyu Zhang, and Dongyan Xu, "GUITAR: Piecing together android app GUIs from memory images," In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, (2015).

[8] Qiang Yan, Jin Han, Yingjiu Li, Jianying Zhou, Robert H. Deng, "Designing leakage-resilient password entry on touchscreen mobile devices." Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, (2013).

[9] Roth, Volker, Kai Richter, and Rene Freidinger. "A PIN-entry method resilient against shoulder surfing." Proceedings of the 11th ACM conference on Computer and communications security. ACM, (2004).

[10] Lee, Mun-Kyu, Jin Bok Kim, and Matthew K. Franklin. "Enhancing the Security of Personal Identification Numbers with Three-Dimensional Displays." Mobile Information Systems 2016 (2016).

[11] Andrea Bianchi, Ian Oakley, Dong Soo Kwon, "The phone lock: audio and haptic shoulder-surfing resistant PIN entry methods for mobile devices." Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction. ACM, (2011).

[12] Andrea Bianchi, Ian Oakley, Dong Soo Kwon, "Counting clicks and beeps: Exploring numerosity based haptic and audio PIN entry." Interacting with computers 24(5), 409-422, (2012).

[13] Lee, Mun-Kyu, Hyeonjin Nam, and Dong Kyue Kim. "Secure bimodal PIN-entry method using audio signals." Computers & Security 56, 140-150, (2016).

[14] Text2 Speech, available in http://www.text2speech.org/

[15] Swype, available in http://www.swype.com/