석사학위논문

랜덤하게 입력되는 진동의 세기를 활용한 잠금 패턴 보안 강화구현

2019년

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

안 규 황

석 사 학 위 논 문
지도교수 서화정

랜덤하게 입력되는 진동의 세기를 활용한 잠금 패턴 보안 강화구현

Improved security of lock pattern
using random vibration intensity

2019년 12월 일

한성대학교 일반대학원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

안 규 황

석 사 학 위 논 문
지도교수 서화정

랜덤하게 입력되는 진동의 세기를 활용한 잠금 패턴 보안 강화구현

Improved security of lock pattern
using random vibration intensity

위 논문을 공학 석사학위 논문으로 제출함

2019년 12월 일

한성대학교 일반대학원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

안 규 황

안규황의 공학 석사학위 논문을 인준함

2019년 12월 일

심사위원장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

랜덤하게 입력되는 진동의 세기를 활용한 잠금 패턴 보안 강화구현

한 성 대 학 교 일 반 대 학 원
I T 융 합 공 학 과
I T 융 합 공 학 전 공
안 규 황

스마트폰에 적용되어 있는 패턴 잠금은 비밀번호 PIN 입력 방식을 대체하는 간편한 암호 입력 방식으로 대부분의 스마트폰에 보급되어 있다. 그러나 시중에 나와 있는 스마트폰에 패턴 인식을 입력하는 방식으로 비밀번호를 드래그 할 때 생기는 손자국으로 인해 Smudge Attack(추정 공격)에 취약하며, 고정되어 있는 포인트로 Shoulder Surfing Attack(어깨너머 공격 혹은 엿보기 공격)에 취약하다. 따라서 본 논문에서는 패턴을 인식하는 새로운 방식을 제안한다. 드래그를 입력할 때 단순히 패턴의 모양만 인식하는 것이 아닌 각 포인트 별로 랜덤하게 발생하는 진동의 세기(강, 중, 약) 중 사용자가 비밀번호로 입력한 포인트의 진동의 세기를 발생시킴에 따라 동일한 패턴을 그려도 한 포인트 당 3^n만큼의 보안 강도가 향상된다. 따라서 기존의 잠금 패턴의 경우 총 389,112가지 패턴을 그릴 수 있으나, 본 논문에서 제안하는 잠금 패턴은 약 9,953배 많아진 3,872,929,464가지 패턴을 그릴 수 있다. 이는 같은 'Z' 패턴을 그리더라

도 각 포인트 별 진동의 강도를 입력해야하기 때문에 ‘Z' 패턴의 Smudge Attack의 경우 $2*3^5$ 만큼 그리고 Shoulder Sulfing Attack의 경우 3^5 만큼 보안 강도가 향상된다. 따라서 같은 'Z' 패턴을 드래그 하더라도 완벽히 다른 패턴이 되는 새로운 패턴 인식 방법을 제안한다.

【주요어】 드래그, 스마트 폰, 어깨 너머 공격, 추정 공격, 패턴 잠금

목 차

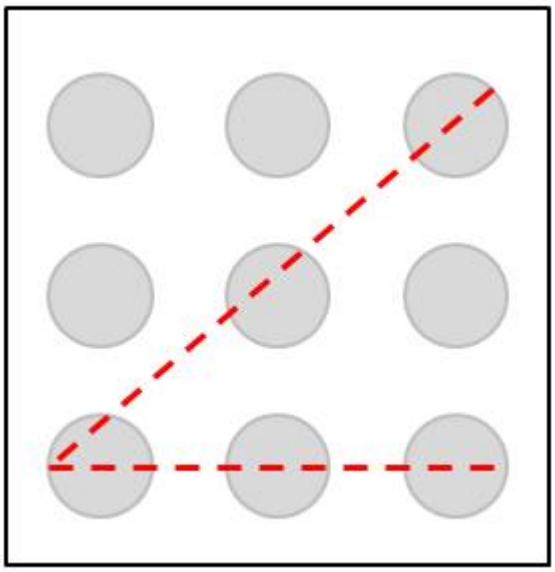
표 목 차

그 림 목 차

제 1 장 서론

제 1 절 패턴 잠금

패턴 잠금이란 안드로이드 핸드폰에서 제공하는 비밀번호 입력 방식 중 하나로 [그림 1-1]과 같이 가로 3개, 세로 3개 총 9개의 점을 이용하여 사용자가 드래그한 패턴을 이용하여 잠금을 해제하는 잠금 기법이다. [그림 1-1]은 1행 3열(이하 [1, 3])에서부터 [2, 2] → [3, 1] → [3, 2] → [3, 3]으로 드래그한 패턴 혹은 [3, 3] → [3, 2] → [3, 1] → [2, 2] → [1, 3]으로 드래그한 패턴의 예시이다.

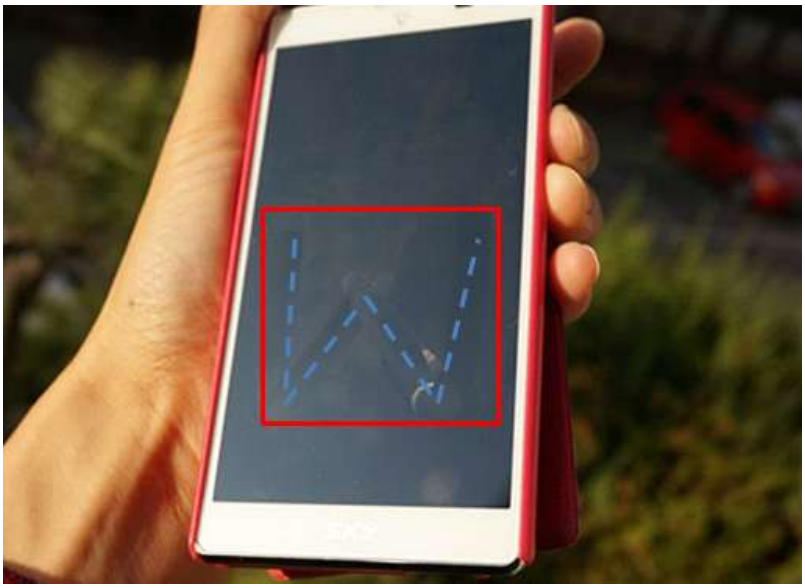


[그림 1-1] 패턴 잠금 예시

3x3 패턴은 단순해 보일 수 있으나, 9개의 점을 이용하여 만들 수 있는 최대 경우의 수는 389,112가지이다.

제 2 절 연구 내용

패턴 잠금 입력 방식은 현대 사회에 많은 분야에서 사용되고 있다. 가장 가까운 곳에서는 핸드폰 잠금 화면 그리고 인터넷 뱅킹 서비스를 이용할 때도 사용된다. 그러나 우리가 사용하고 있는 가장 기초적인 패턴 잠금 입력 방식은 초록에서 언급했듯이 Smudge Attack, Shoulder Surfing Attack 등 많은 공격에 취약하며 심지어 Smudge Attack의 경우 단순하게는 [그림 1-2]와 같이 핸드폰 위에 드래그 한 모양을 보고 유추할 수 있다.



[그림 1-2] 드래그하고 남은 자국을 이용한 패턴 잠금 유추

이러한 공격들로부터 보호하기 위해 핸드폰 UI에 드래그하는 잠금 패턴이 그려지지 않게끔 하는 새로운 패턴 잠금 기법이 제안되었다. 해당 패턴 잠금 기법의 경우 핸드폰 UI에 아무런 패턴이 그려지지 않는다는 점이 있지만, 추정 공격하는 Smudge Attack은 [그림 1-2]와 같이 남은 자국을 이용한 패턴 잠금 공격은 기존 패턴 잠금과 같이 동일하게 공격할 수 있다. 또한 어깨의 움직임을 보고 공격하는 Shoulder Surfing Attack과 같

이 여전히 취약하다.

본 논문에서는 위에 언급된 공격들로부터 아주 안전한 패턴 잠금을 제안하기 위해 각 포인트별로 서로 다른 3가지 진동 세기를 랜덤(각 포인트별로 강, 약, 중간을 랜덤하게 제공하여 만약 [1, 1]에 사용자가 등록해놓은 강도가 '강'일 때 드래그를 할 때 마다 강, 약, 중간이 랜덤하게 제공되며 '강'의 강도가 나왔을 때 다른 패턴으로 이동하면 된다.)하게 사용자에게 제공하여 동일한 패턴 내에서도 다른 패턴을 내포하는 새로운 패턴 잠금을 제안한다.

제 3 절 논문 구성

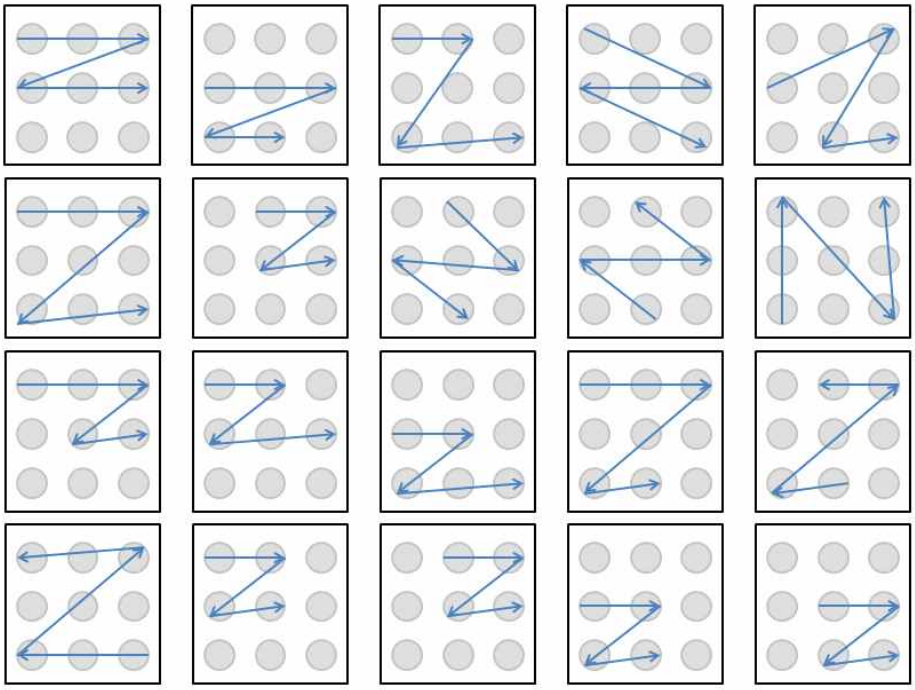
본 논문의 구성은 다음과 같다. 2장에서는 기존에 사용되는 잠금 패턴 방식의 취약성에 대해 알아보며, 이를 해결하기 위해 제안된 강화된 잠금 패턴에 대해 소개한다. 3장에서는 본 논문에서 제안하는 새로운 보안 패턴에 대해 설명한다. 4장에서는 본 논문에서 제안한 기법과 기존 잠금 패턴의 성능 비교를 진행하며, 5장에서 결론과 향후과제를 제시한다.

제 2 장 관련 연구

제 1 절 Cracking Android Pattern Lock in Five Attempts

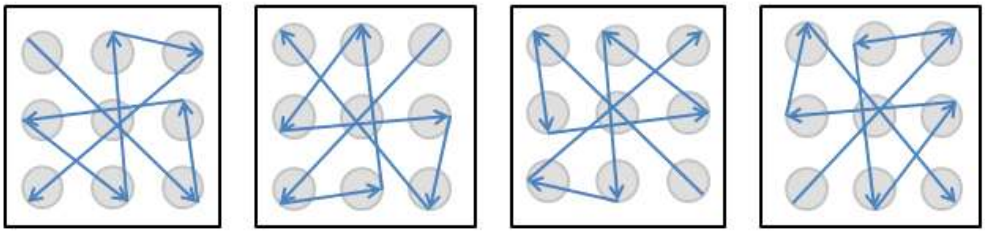
안드로이드 기기를 사용하는 대부분의 사용자는 본인의 잠금 패턴 보안을 강화하기 위해 패턴에 사용되는 포인트 수를 추가하여 복잡한 패턴을 기획한다. 그러나 Guixin Ye의 연구진에 의하면 현재 안드로이드 기기에서 제공하는 잠금 패턴의 경우 포인트 수를 추가하여 복잡한 패턴을 만들수록 되려, 보안에 취약해진다고 주장한다.

Shoulder Surfing Attack을 이용하여 사용자가 'Z' 모양으로 드래그함을 공격자가 획득했을 경우 [그림 2-1]과 같이 다양한 패턴을 시도해보아야 사용자가 저장한 패턴을 획득할 수 있다.



[그림 2-1] 단순한 패턴의 손가락 움직임 궤적에 대한 가능한 매핑 예시

그러나 만약 사용자가 복잡한 패턴을 [그림 2-2]와 같이 사용할 경우 공격자는 'Z' 패턴보다 낮은 경우의 수로 사용자 패턴을 획득할 수 있다.



[그림 2-2] 복잡한 패턴의 손가락 움직임 궤적에 대한 가능한 매핑 예시

따라서 만약 공격자가 Smudge Attack 혹은 Shoulder Surfing Attack을 이용하여 사용자의 패턴 모양을 획득할 수 있다면, 복잡한 패턴이 단순한 패턴보다 낮은 안전성을 갖게 됨을 보여준다.

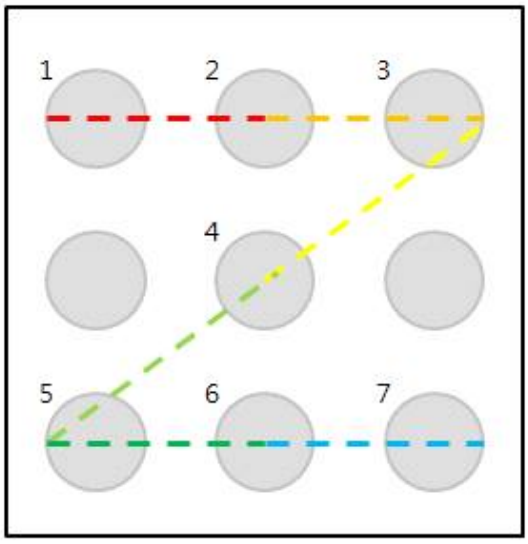
제 2 절 T-Lock: 스마트폰용 삼중 요소 기반 패턴 락 인증 기법

잠금 패턴에서는 패턴을 그리는데 있어 1가지(패턴)만을 활용한다. 그러나 T-Lock의 경우 3가지 요소를 이용하여 패턴을 인식하게 된다. 첫 번째로는 패턴, 두 번째로는 압력 정보 그리고 마지막으로 지문이다.

첫 번째로 비교하는 패턴의 경우 기존 패턴과 마찬가지로 사용자가 그리는 방법에 대한 지식 기반 인증 요소이며, 두 번째로 비교하는 압력 정보는 소유 기반의 인증 요소이다. 마지막으로 지문 정보는 생체 인식 기반의 인증 요소를 나타낸다.

압력 정보의 경우 각 포인트에서 포인트로 이동할 때 발생하는 손가락 압력을 이용한다. [그림 2-3]과 같이 'Z' 모양으로 드래그할 때 1번 포인트에서 2번 포인트(이하 1P → 2P)로 이동할 때 눌리는 손가락의 압력을 저장하고, 2P → 3P로 이동할 때 눌리는 손가락의 압력을 저장한다. 이와 같이 마지막 지점인 6P → 7P까지 이동하는 손가락의 압력을 저장하여

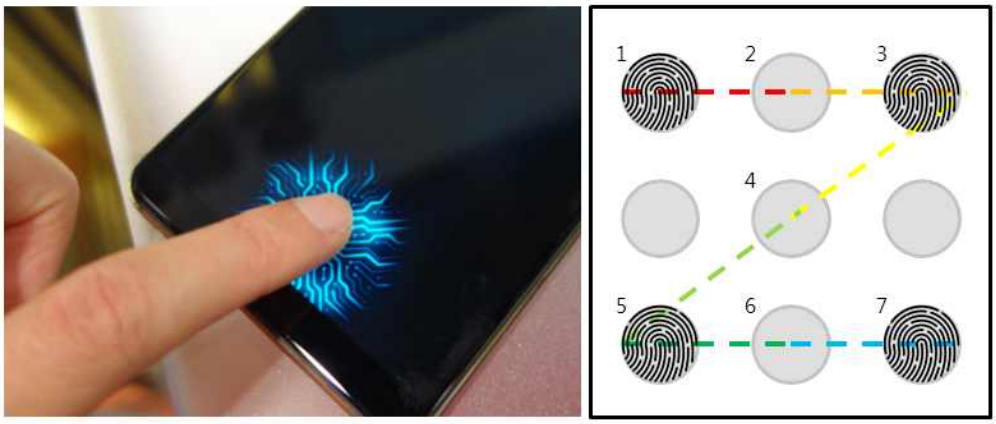
사용자가 최초에 저장한 잠금 패턴 및 압력과 비교하는 방식이다.



[그림 2-3] 패턴 입력과 압력 변화의 예시

세 번째로 제안한 지문을 활용한 잠금 패턴의 경우 드래그하는 영역에 대해서 시작 지점과 끝 지점 그리고 꺾이는 부분에서 드래그하는 사용자의 지문을 저장 된 지문과 비교하는 방식이다. 따라서 [그림 2-3]의 경우 [그림 2-4]와 같이 시작: 1P, 꺾임: 3P, 5P, 끝: 7P에서 지문을 비교하게 된다.

그러나 지문을 활용한 잠금 패턴은 현재 대중화되어 있는 대부분의 스마트폰에는 적용할 수 없다. 액정에서 지문을 인식하는 대부분의 스마트폰은 [그림 2-4]의 왼쪽과 같이 특정 영역에서만 지문을 인식할 수 있다. 이는 모든 액정이 지문을 인식할 수 있는 액정이어야 [그림 2-4]와 같이 꺾이는 지점인 1P, 3P, 5P, 7P에서 저장된 지문과 비교하는 작업을 수행하기 때문이다.



[그림 2-4] 패턴 입력과 지문 인증 예시

제 3 장 포인트와 진동을 활용한 패턴 잠금

대중적으로 사람들이 사용하는 잠금 패턴 기능을 사용해보면, 시작 포인트를 터치하는 순간부터 종료 포인트까지 손의 움직임에 따라 선이 이어지게 된다. 이때 스마트폰에 생긴 선을 공격자가 기억할 경우 단 2가지 경우(시작 포인트 → 종료 포인트 혹은 종료 포인트 → 시작 포인트)로 완벽하게 잠금 패턴을 해제할 수 있다.

사용자의 잠금 패턴 모양을 볼 수 있는 가시거리는 [그림 3-1]과 같이 일상생활에서 일반적으로 마주할 수 있는 거리이다. 공격자로부터 잠금 패턴이 그려지는 모습을 숨기기 위해 보안 필름 등을 붙이는 1차원적 행위를 시도하지만, 보안 필름을 붙인다고 모든 공격으로부터 안전해 지는 것은 아니다. 가령 Shoulder Surfing Attack, Smudge Attack으로부터의 위험성은 여전히 존재한다.

[그림 3-1] 잠금 해제 프로세스를 촬영하기 위해 휴대폰 카메라를 사용하는 시나리오의 예시

본 논문에서는 보안 필름을 붙이는 등의 추가적인 장비를 사용하지 않고, 엿보기 공격, Shoulder Surfing Attack, Smudge Attack으로부터 안전해질 수 있는 3가지 방법을 제안한다.

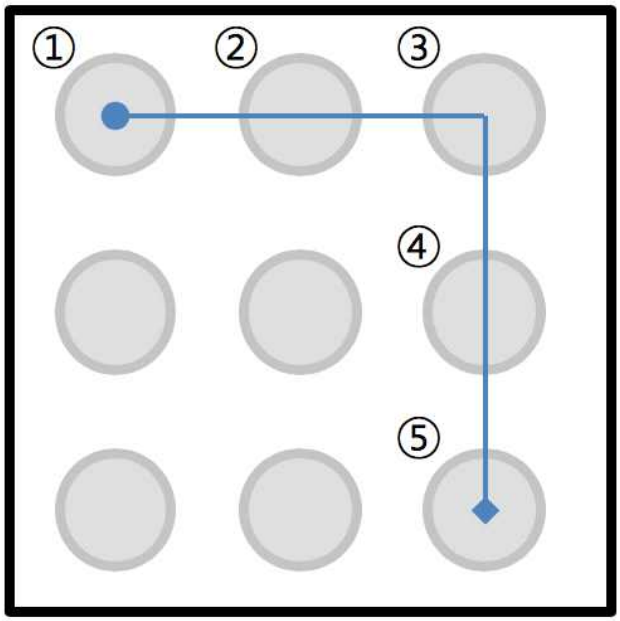
제 1 절 패턴 가시화

현재 일상적으로 사용하는 잠금 패턴을 보면 단순히 점과 점을 이어 하나의 패턴을 그려놓고 해당 패턴이 기존 사용자가 저장해놓은 패턴과 일치할 경우 잠금이 해제되며 그렇지 않을 경우 여전히 잠기게 된다. 그러나 본 논문에서 제안하는 방법은 단순히 패턴을 그리는 것이 아닌 패턴을 그림과 동시에 각 포인트 별 진동 세기를 입력하는 것이다.

각각의 포인트에는 '강, 중, 약'의 진동 세기를 출력한다. 이때 사용자는 기존의 패턴을 그리는 행위와 더불어 각 포인트 별 진동을 추가적으로 입력해주는데, 별도의 추가적인 행위를 요하는 것은 아니다. 구체적 사용 방법은 다음과 같다. 첫 번째로 사용자는 패턴을 그릴 시작 포인트를 지정하고, 해당 포인트에 터치한다. 두 번째로는 해당 포인트에서 사용자에게 '강, 중, 약'의 진동 세기 중 하나를 불규칙하게 선택하여 출력한다. 마지막으로 사용자는 해당 포인트에 입력 될 진동 세기(강, 중, 약 중 선택) 중 하나를 선택하고 '중'일 경우 스마트 폰이 '중'의 세기로 진동이 울린 직후 다른 지점의 포인트로 이동하면 된다.

기존 잠금 패턴과 같이 단순히 포인트에서 포인트로 이동하는 것이 아닌 각 포인트 별 '강, 중, 약'의 세기를 입력하고 원하는 세기가 출력 됐을 때 다른 포인트로 이동할 경우 해당 지점에서는 3가지 경우의 수가 추가된다. 즉 총 9개의 점으로 이루어진 잠금 패턴은 기존 잠금 패턴보다 최대 3^9만큼 보안 강도가 향상된다.

따라서 만약 [그림 3-2]와 같이 'ㄱ' 패턴을 그릴 경우, 총 5개의 포인트를 사용하기 때문에 3^5만큼 보안이 향상된다.



[그림 3-2] 포인트별 보안 강도 향상 예시

각 포인트 별 '강 → 중 → 약' 순으로 진동을 출력하지 않고 랜덤하게 진동을 출력하는 이유는 Shoulder Surfing Attack으로부터 보호하기 위함이다. Shoulder Surfing Attack은 사용자의 어깨의 움직임을 보고 잠금 패턴을 유추하는 공격 방법이다. 이때 만약 '강 → 중 → 약' 순으로 진동이 출력 될 경우 포인트에서 포인트로 움직인 직후 어깨가 움직이지 않는 시간을 측정한다면 어떤 세기의 진동을 입력하였는지 유추가 가능하다. 따라서 각 포인트별로 진동의 출력은 항상 불규칙하게 선정되어 출력된다.

다음으로 랜덤하게 출력되는 3가지 진동 세기(강, 중, 약)에 대한 방법에 대해 설명하겠다. 기존의 잠금 패턴의 경우 포인트에서 포인트로 이동할 때 단순히 서만 연결된다. 그러나 제안하는 기법의 경우 각 포인트를 사용자가 터치했을 때 진동(강, 중, 약)이 발생하게 되고, 사용자가 비밀번호로 입력한 진동의 세기가 출력 됐을 때 해당 진동을 입력하고 다음 포인트로 이동하는 방식이다.

이때 각 포인트에서 발생하는 작업에 대한 수도코드는 [그림 3-3]과 같다. Vibration 배열의 크기는 3이며, 해당 배열 안에는 1, 2, 3이라는 정수 값

이 정의되어 있다. 해당 정수의 의미로는 1은 진동의 세기의 약을 의미하며, 2는 징동 세기의 중 그리고 3은 진동 세기의 강을 의미한다. 포인트를 사용자가 터치했을 때 Vibration 배열의 3가지 값 중 한 개의 값이 랜덤하게 출력되게 되고, 0.5초 후 출력 된 값을 제외한 2가지 값 중 한 개의 값이 랜덤하게 출력되게 된다. 다시 0.5초 후 마지막으로 남은 값이 출력되는 형식이다. 예를 들어 1, 2, 3 값중 최초로 랜덤하게 2가 선택되어 출력 될 경우 다음으로는 1과 3중에 선택되게 되고, 3이 선택될 경우 마지막으로 1이 출력되는 형식이다.

Pseudocode for Random Vibration

```
01: void shuffle(int *arr, int size) {
02:     int tmp;
03:     int randNum;
04:     for (int i=0; i < size-1; i++) {
05:         randNum = rand() % (size - i) + i;
06:         tmp = arr[i];
07:         arr[i] = arr[randNum];
08:         arr[randNum] = temp;
09:     }
10: }
11: int main(void){
12:     int vibration[3] = {1, 2, 3};
13:     suffle(vibration, 3);
14: }
```

[그림 3-3] 랜덤하게 출력되는 진동에 대한 수도코드

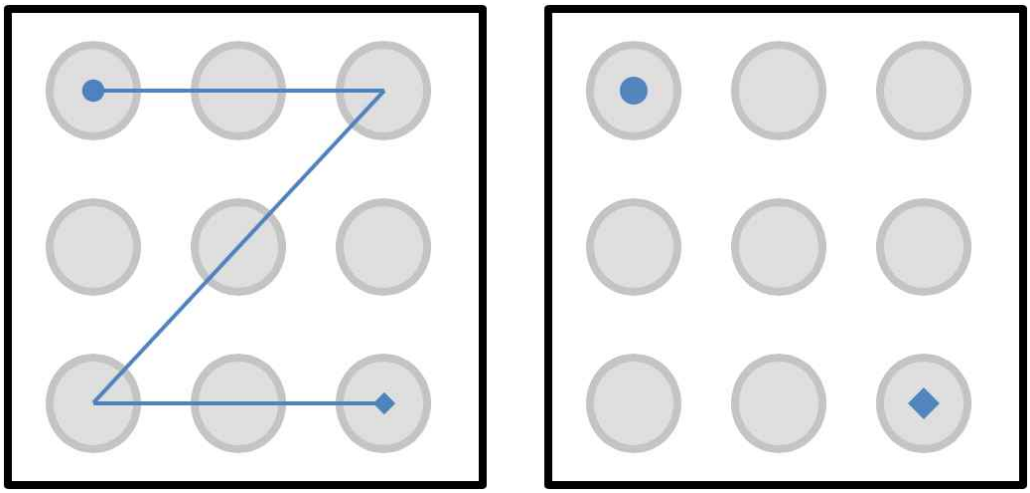
사용자가 잠금 패턴 값으로 입력한 진동 세기가 단번에 나오고, 다음 포인트로 이동할 경우 대기하고 있는 나머지 2개 값이 출력되는 일은 발생하지 않으며, 만약 사용자가 비밀번호를 입력할 때 첫 번째, 두 번째 진동 값도 아

닌 마지막 세 번째로 사용자 잠금 패턴 값이 출력된다면 3번의 기다림이 요구된다. 동작 모습은 1)링크에서 확인할 수 있다.

제 2 절 패턴 비가시화

공격자가 사용자의 잠금 패턴을 획득하는데 있어 가장 강력한 방법은 사용자가 그린 완성 된 잠금 패턴을 바로 옆에서 지켜보는 행위이다. 해당 방법은 사용자가 입력하는 시작 포인트부터 그리는 패턴을 볼 필요 없이, 종료 포인트 직전에 엿볼 수 있다면 사용자가 입력한 전체 잠금 패턴을 완벽하게 탈취할 수 있다.

이를 방지하기 위해 3장 1절에서 제안하는 진동 입력 방식은 동일하게 적용되지만, 사용자의 손가락 움직임에 따라 그려지는 선을 보이지 않게 하려고 한다. 기존의 잠금 패턴은 'Z' 모양을 그리면 [그림 3-4]의 좌측과 같이 선이 나타나게 된다. 그러나 비가시화 할 경우 [그림 3-4]의 우측과 같이 사용자 인터페이스에는 시작 포인트와 종료 포인트만 표시할 뿐 어떻게 드래그 했는지는 보이지 않는다. 하지만 백단에서는 동일한 패턴인지 아닌지 판단이 이루어진다.



[그림 3-4] 좌) 기존 잠금 패턴에서 'Z' 모양으로 드래그할 때
우) 비가시화 잠금 패턴에서 'Z' 모양으로 드래그할 때

1) https://j.gifs.com/VA2NMX.gif // 제안하는 잠금 패턴의 실제 구동 영상

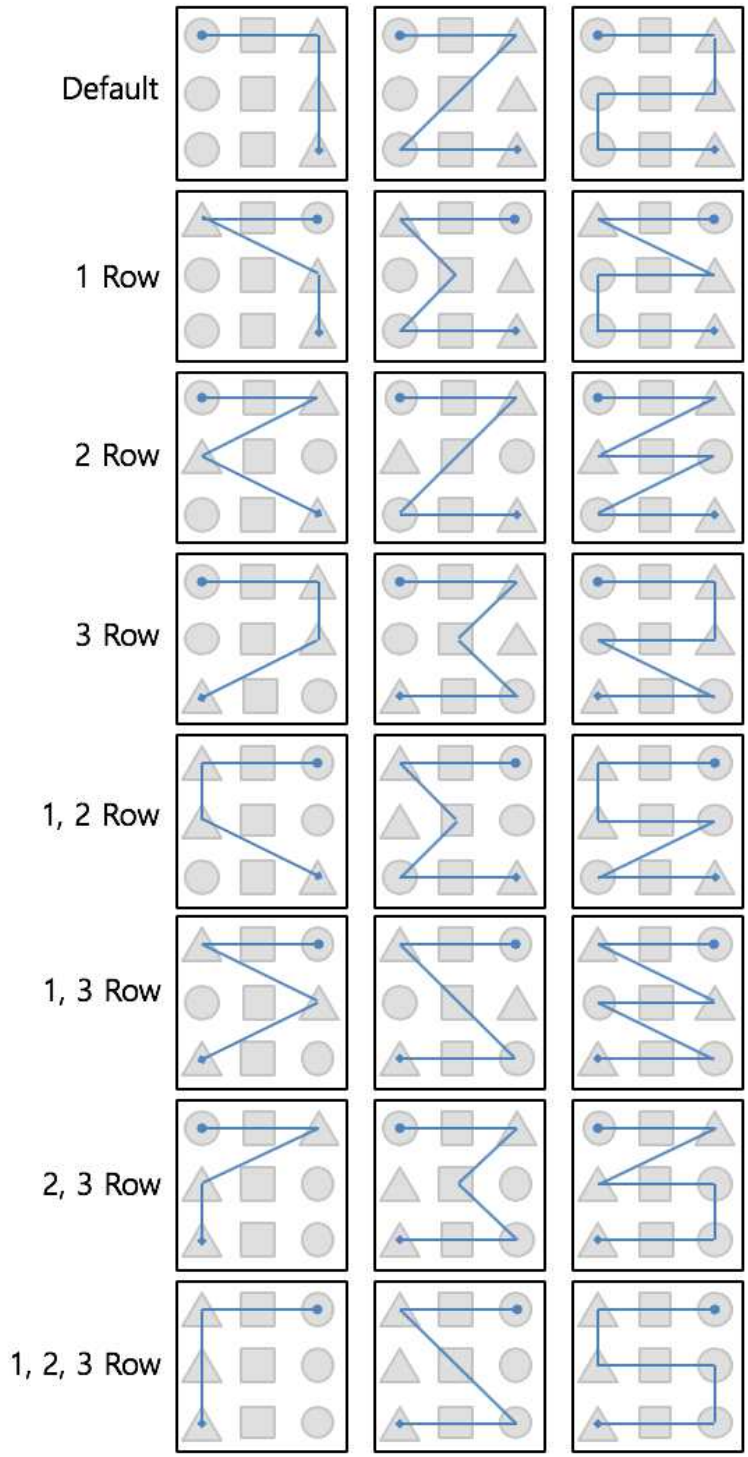
제 3 절 패턴 난독화

공격자가 사용자의 잠금 패턴을 탈취하려고 할 때, 비가시화도 좋은 방법일 수 있지만, 그려진 패턴 자체를 해독할 수 없게 한다면 공격자에게 혼란을 줄 수 있기에 더 좋은 방어 방법이다.

1) 기준 행을 기준으로 1, 3열 섞기

난독화를 시킴에 있어 기준 행을 어떻게 두느냐에 따라 각기 다른 UI가 생성된다. 만약 기준 행을 1행으로 두면 [1, 1]에 있는 포인트와 [1, 3]에 있는 포인트를 UI적으로 화면상에서 바꿔준다. 따라서 사용자가 [1, 1]을 터치할 경우 프론트 단에서는 [1, 3]을 터치한 것과 같이 화면상에 그려지며, 백단에서는 사용자가 기존에 터치한 [1, 1]로 인식되게 된다.

기준 행은 1행, 2행, 3행, 1-2행, 1-3행, 2-3행, 1-2-3행으로 총 6가지가 될 수 있다. 따라서 'ㄱ'을 그리더라도 기준 행을 어떻게 잡느냐에 따라 [그림 3-5]와 같이 총 6가지로 표현되게 된다.

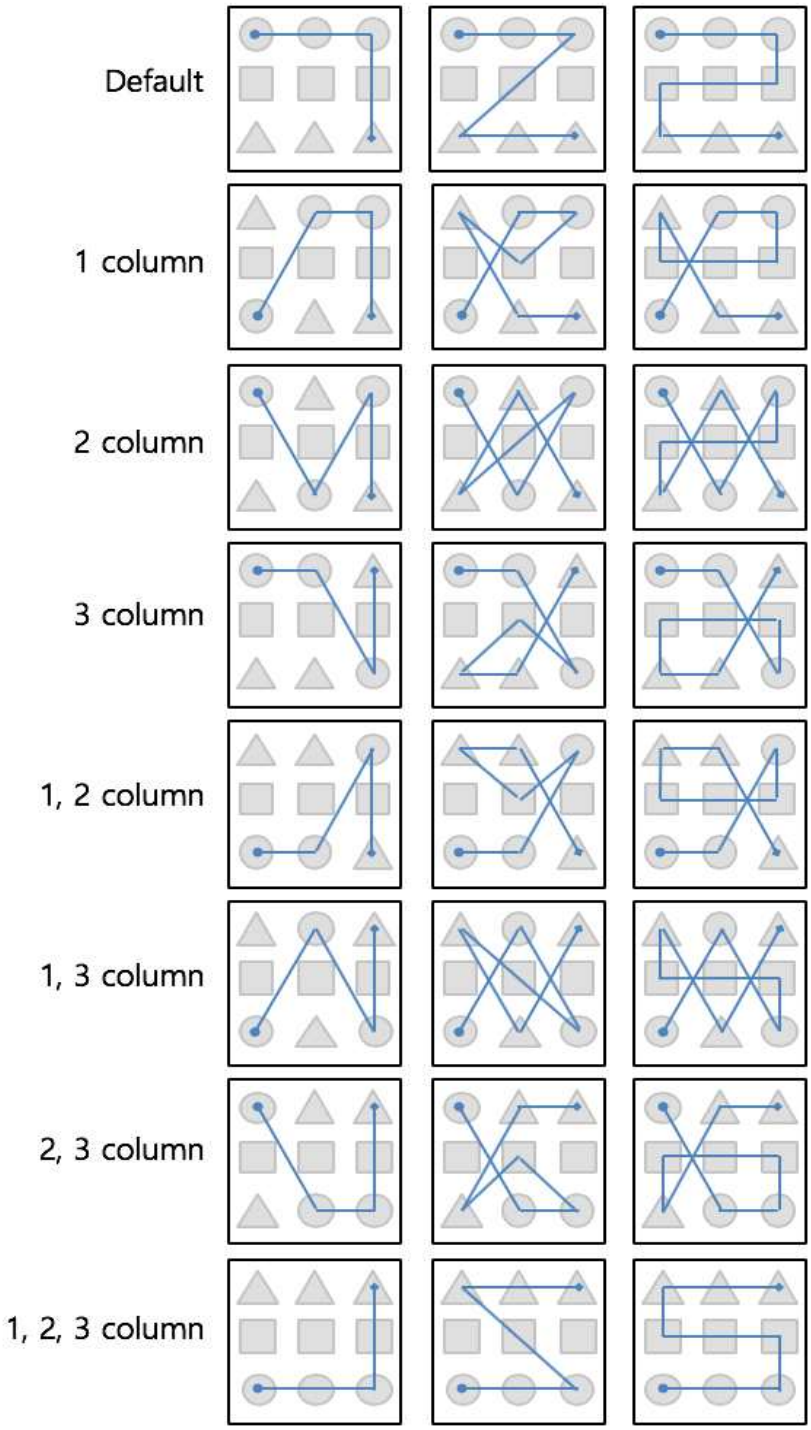


[그림 3-5] ‘ㄱ’, ‘Z’, ‘ㄹ’을 기준 행에 따라 어떻게 나타나는지 보여주는 도식화

2) 기준 열을 기준으로 1, 3행 섞기

난독화를 시킴에 있어 기준 행과 기준 열은 매우 상이한 UI 모습을 보여준다. 마약 기준 열을 1열로 두면 [1, 1]에 있는 포인트와 [3, 1]에 있는 포인트를 화면상에서 바꿔준다. 그러나 백 단에서는 사용자가 기존에 터치한 [1, 1]로 인식된다.

기준 열을 1열, 2열, 3열, 1-2열, 1-3열, 2-3열, 1-2-3열로 총 6가지가 되며, 3장 3절 1)과 다르게 기준을 열로 잡았기 때문에 같은 'ㄱ'을 그리더라도 [그림 3-6]과 같이 상이하게 나타난다.

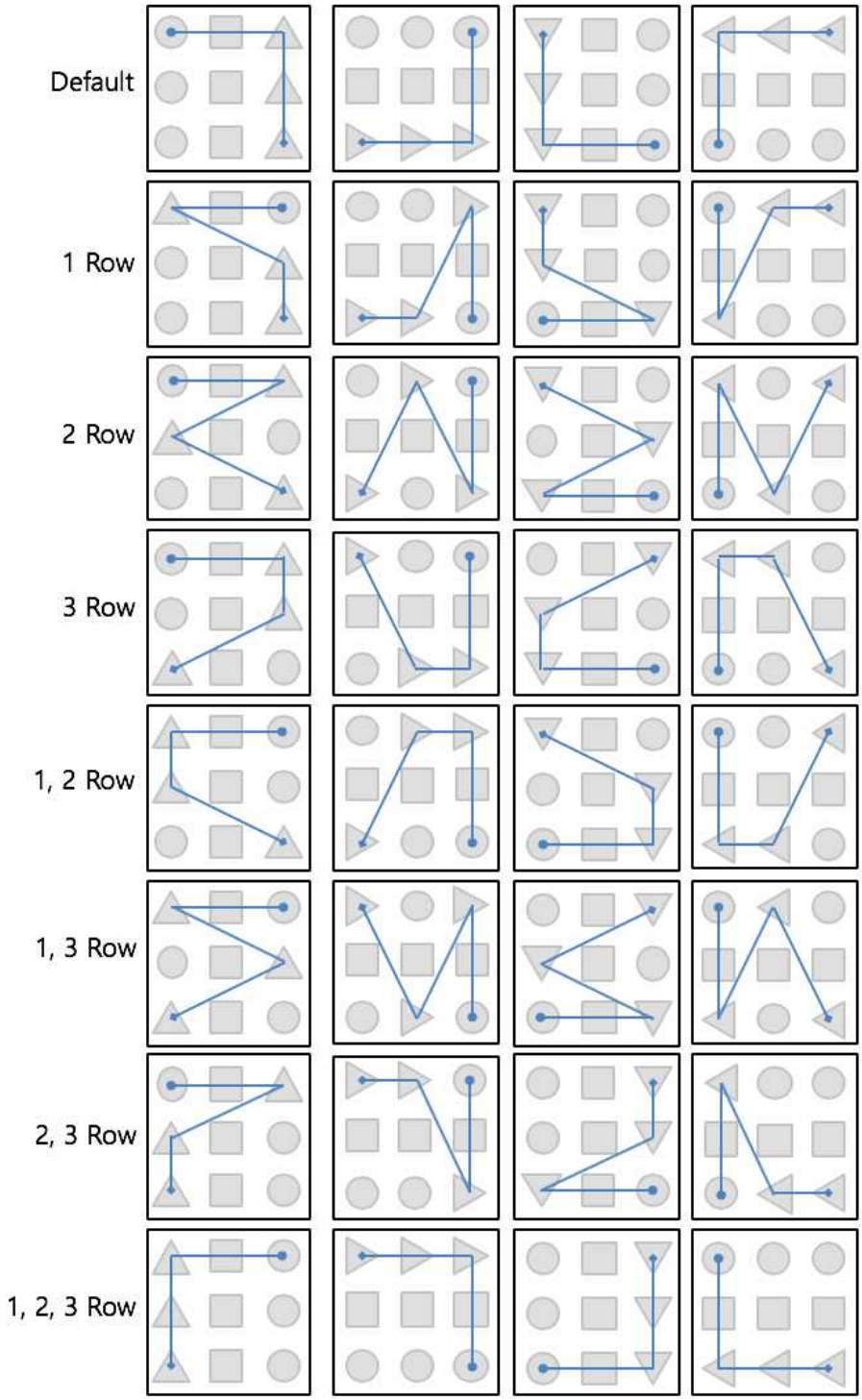


[그림 3-6] ‘ㄱ’, ‘Z’, ‘ㄹ’을 기준 열에 따라 어떻게 나타나는지 보여주는 도식화

3) 3-3-1 혹은 3-3-2를 적용한 후 90°,180°,270°회전

난독화를 시키는데 있어 다양한 방법들이 중첩됨에 따라 더욱 복잡한 결과가 발생하게 된다. 사용자가 드래그 하는 패턴을 기준 행, 열로 섞는 것뿐만 아니라 그렇게 발생한 결과를 90°,180°,270°로 한번 더 섞어준다면, [그림 3-7]과 같이 다양한 패턴을 확인할 수 있다.

결론적으로 사용자가 지정한 패턴을 그릴 때 기준 행, 열로 섞는 행위와 더불어 각도를 바꿔주었을 때 겹치는 패턴이 없다면 하나의 패턴에 최대 32가지 경우의 수가 발생하게 된다. 사용자가 패턴을 그릴 때 마다 백단에서는 사용자 의도대로 인식되며, 프론트 단에서는 32가지 패턴이 불규칙하게 출력된다면 기존의 잠금 패턴보다 Shoulder Surfing Attack 등에 안전하게 된다.



[그림 3-7] 좌측부터 'ㄱ'의 기준 행 그리고 90°,180°,270°회전 하였을 때 어떻게 나타나는지 보여주는 도식화

제 4 장 성능 평가

제 1 절 공격 기법

본 장에서는 잠금 패턴을 탈취하는데 사용되는 공격인 Brute-Force Attack, Shoulder Surfing Attack, Recording Attack 그리고 Smudge Attack을 이용하여, 대부분의 안드로이드 핸드폰에서 사용하는 잠금 패턴과 본 논문에서 제안하는 랜덤하게 입력되는 시간차를 활용한 잠금 패턴에 대한 성능 비교를 진행한다.

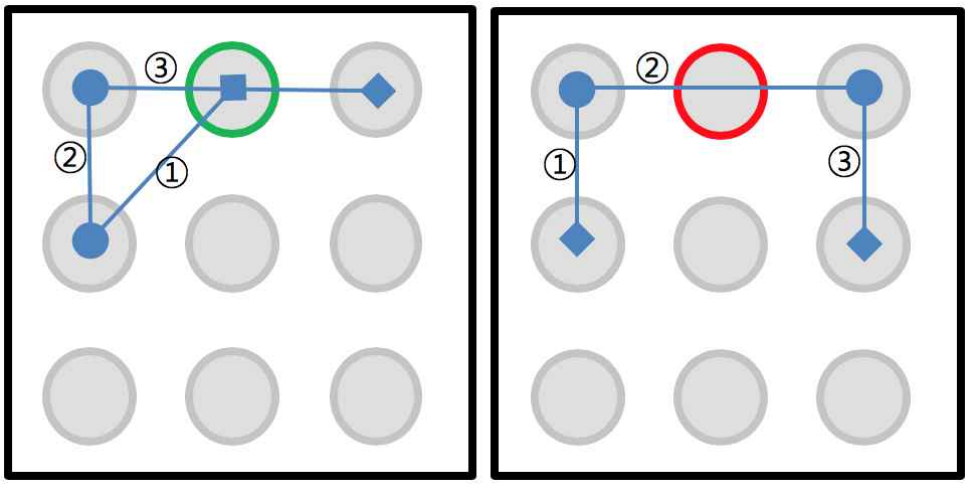
1) Brute-Force Attack

Brute-Force Attack이란 가능한 모든 경우의 수를 시도하는 공격 방법으로 만약 비밀번호가 숫자 4자리로 구성되어 있다면, '0000~9999'까지 최대 10,000번의 시도를 해야 하는 무차별 대입 공격이다. Brute-Force Attack은 운이 정말 좋으면 단 한번 만에 비밀번호를 획득할 수 있는 강력한 공격 방법이나, 운이 정말 안 좋을 경우 모든 경우의 수를 시도해봐야 하기 때문에 엄청난 시간이 걸릴 수 있다. 그러나 오랜 시간이 걸리더라도 확실하게 비밀번호를 획득할 수 있는 고전적이지만 아주 강력한 공격 기법이다.

Brute-Force Attack을 이용하여 기존 잠금 패턴과 본 논문에서 제안하는 기법의 안전성을 비교하면 [표 4-1]과 같다. Brute-Force Attack의 취약점은 바로 경우의 수가 늘어날수록 시간이 오래 걸린다는 점이다. 따라서 Brute-Force Attack에 대해 안전성을 높이기 위해서는 잠금 패턴의 복잡도를 높여야한다.

기존 잠금 패턴을 이룰 수 있는 기본 조건은 다음과 같다. 첫 번째로 점은 4개 이상 연결되어 있어야 한다. 두 번째로 최대 9개의 점에 연결할 수

있다. 세 번째로 한번 연결한 점은 다시 연결할 수 없다. 네 번째로 같은 모양의 패턴이라도 시작점이 다르면 다른 패턴이다. 마지막으로 직선상에 위치한 점들은 건너뛸 수 없다. 그러나 한번 연결된 점인 경우에는 [그림 4-1]의 왼쪽과 같이 건너뛰어 연결할 수 있다.



[그림 4-1] 좌) 중간점을 건너 뛸 수 있는 패턴의 예시
우) 중간점을 건너 뛸 수 없는 패턴의 예시

기존 잠금 패턴이 그릴 수 있는 경우의 수는 다음과 같다.

$\sum_{r=4}^{9}$ 으로 9가지의 포인트 중 4개의 포인트로 패턴을 그린 경우부터 9개의 포인트로 패턴을 그린 경우의 수까지 모두 더해준다. 4개의 포인트로 그렸을 때는 총 3,024개, 5개의 포인트로 그렸을 때는 총 15,120개, 6개의 포인트로 그렸을 때는 총 60,480개, 7개의 포인트로 그렸을 때는 총 181,440개, 8개의 포인트로 그렸을 때는 362,880개 마지막으로 9개의 포인트로 그렸을 때는 362,880개로 모두 더하면 985,824개가 나온다. 그러나 985,824개 안에는 [그림 4-1]의 오른쪽과 같이 점과 점 사이에 중간점이 있을 경우는 제외하지 않고 계산한 값이기 때문에 (1, 3),

(3, 1), (4, 6), (6, 4), (7, 9), (9, 7), (1, 7), (7, 1), (2, 8), (8, 2), (3, 9), (9, 3), (1, 9), (9, 1), (3, 7), (7, 3)과 같이 중간점을 건너뛰는 경우는 제외해야한다. 중간점을 건너뛰는 경우를 제외하면 4개의 포인트로 그렸을 때는 총 1,624개, 5개의 포인트로 그렸을 때는 총 7,152개, 6개의 포인트로 그렸을 때는 총 26,016개, 7개의 포인트로 그렸을 때는 총 72,912개, 8개의 포인트로 그렸을 때는 140,704개 마지막으로 9개의 포인트로 그렸을 때는 140,704개로 모두 더하면 389,112개가 나온다.

그러나 본 논문에서 제안하는 잠금 패턴은 각 포인트 별로 '강, 중, 약'이 패턴을 그리는 것과 독립적으로 입력해야하기 때문에 포인트 별 3가지 경우의 수가 추가된다. 따라서 4개의 포인트로 그렸을 때는 $1,624*^{4}$으로 131,544개, 5개의 포인트로 그렸을 때는 $7,152*^{4}$으로 1,737,936개, 6개의 포인트로 그렸을 때는 $26,016*^{4}$으로 18,965,664개, 7개의 포인트로 그렸을 때는 $72,912*^{4}$으로 159,458,544개, 8개의 포인트로 그렸을 때는 $140,704*^{4}$으로 923,158,944개 마지막으로 9개의 포인트로 그렸을 때는 $140,704*^{4}$으로 2,769,476,832개로 모두 더하면 3,872,929,464개이다. 따라서 본 논문에서 제안하는 잠금 패턴은 기존의 389,112개 보다 약 9,953배 많아진 3,872,929,464가지 경우의 수를 가진다.

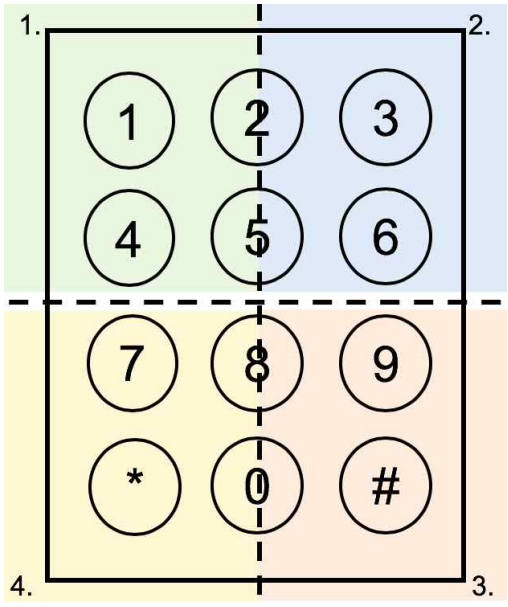
[표 4-1] Brute-Force Attack에 의한 잠금 패턴 안전도 비교

	Default	Visible	Invisible	Obfuscation
Brute-Force Attack	Low	High	High	High

2) Shoulder Surfing Attack

Shoulder Surfing Attack은 공공장소에서 직접적으로 손쉽게 당할 수 있는 공격 기법 중 하나로 주변 사람들에 대한 위험성을 강하게 인지하지 못 하는 상황에서 발생한다. 한 예로 사람이 붐비는 지하철 내에서 핸드폰 잠금을 풀 때 사용자는 거리낌 없이 비밀번호를 입력한다. 이때 주변에 있는 사람은 별다른 노력을 하지 않고 단순히 엿보는 행위만으로 비밀번호를 획득할 수 있다.

Shoulder Surfing Attack은 사용자가 비밀번호(잠금 패턴)을 누를 때 발생하는 손의 위치 혹은 어깨의 움직임을 이용하여 비밀번호를 유추하는 방법으로 [그림 4-2]를 보면 가로축, 세로축 중심으로 좌상단, 좌하단, 우상단 그리고 우하단으로 나누었다. 비밀번호를 누르기 위해 손가락을 패턴 위에 올렸을 때 좌상단에 있는 1번을 누르는 손의 위치와 우하단에 있는 9번을 누르는 손의 위치는 확실히 다르기 때문에 공격자는 멀리서도 손가락의 위치만으로 손 쉽게 비밀번호를 유추할 수 있다.



[그림 4-2] 손의 위치에 따라 구분할 수 있는 영역

Shoulder Surfing Attack에 대한 기본 잠금 패턴과 본 논문에서 제안하는 랜덤하게 입력되는 시간차를 활용한 잠금 패턴을 비교하면 [표 4-2]와 같다. 현재 안드로이드 핸드폰에서 사용하는 기본 잠금 패턴의 경우 Shoulder Surfing Attack에 매우 취약하다. 그 이유는 공격자가 사용자의 핸드폰을 훔쳐보고 있을 때 사용자가 그리는 비밀번호 패턴이 사용자 핸드폰 UI에 그대로 노출되기 때문이다.

그러나 본 논문에서 제안하는 잠금 패턴은 가시화 패턴의 경우 중간 정도의 안전성을 가지며 비가시화, 난독화 패턴의 경우 매우 높은 안전성을 가진다. 가시화 패턴은 기본 잠금 패턴과 마찬가지로 사용자가 그리는 비밀번호 패턴이 사용자 핸드폰 UI에 그대로 노출된다. 그러나 각 포인트별 출력되는 진동의 세기를 선택하여 입력해야하기 때문에 공격자 입장에서는 단순히 Shoulder Surfing Attack만으로는 사용자가 랜덤으로 입력하는 진동의 세기를 유추하는 것은 불가능해진다. 또한 비가시화와 난독화는 사용자가 그리는 비밀번호 패턴이 사용자 핸드폰 UI에 표시가 되지 않거나, 사용자가 그리는 패턴과 상이하게 출력되기 때문에 공격자에게 혼란을 줄 수 있다. 가시화와 마찬가지로 공격자가 진동의 세기를 유추하는 것은 불가능하기 때문에 다른 기법들에 비해 안전한 보안강도를 갖게 된다.

[표 4-2] Shoulder Surfing Attack에 의한 잠금 패턴 안전도 비교

	Default	Visible	Invisible	Obfuscation
Shoulder Surfing Attack	Low	Middle	High	High

3) Recording Attack

Recording Attack은 Shoulder Surfing Attack에 전자기기를 추가한 공격 방법으로 사용자가 입력하는 비밀번호를 스마트폰, 카메라 등을 이용하여 동영상을 촬영 및 기록하여 해당 영상을 통해 비밀번호를 유추하는 공격으로 기억력에 의존하는 Shoulder Surfing Attack보다 공격 성공률이 높다.

구글 글래스, 초소형 카메라, 드론 등과 같이 일상생활에서 사용자가 인지하지 못하게 동영상을 촬영할 수 있는 다양한 장비들이 개발되고 있다. 사용자와 공격자가 근거리 내에 있을 때만 유용할 수 있는 Shoulder Surfing Attack과 달리 Recording Attack은 원거리에서도 활용될 수 있어 Recording Attack은 보다 광범위한 공격 방법이다.

Cracking Android Pattern Lock in Five Attempts을 연구한 Guixin 연구진에 의하면 Shoulder Surfing Attack의 공격 성공률은 알아보기 위해 다양한 1~9m까지 다양한 거리에 의해 실험을 진행했다. 촬영 지점이 가까울수록 잠금 패턴을 알아낼 확률이 높았다. 1m이내인 경우엔 100%, 2m 내외인 경우엔 98.7%, 3.5m내외인 경우엔 68%의 성공률을 보였다. Recording Attack은 카메라 성능에 따라 다른 결과가 나오지만 평균적으로 68%보다 높은 공격 성공률을 보인다고 한다. 그러나 아직까지 Recording Attack을 대응할만한 기법은 그리 많지 않다. 하지만 본 논문에서 제안하는 기법을 이용하면 [표 4-3]과 같이 Recording Attack에 안전할 수 있다.

Recording Attack은 Shoulder Surfing Attack과 같이 사용자가 입력하는 잠금 패턴 모습을 보고 유추하는 공격 방법이다. 따라서 기존의 잠금 패턴은 사용자 기억에 의존했던 Shoulder Surfing Attack보다 완벽하게 취약하다. 그러나 본 논문에서 제안하는 기법의 경우 Shoulder Surfing Attack에서도 언급했듯 기본 잠금 패턴과 같이 사용자 입장에 시각적 기능에만 의존하는 것이 아닌 사용자의 촉각을 이용한 비밀번호 입력도 포함되어 있기 때문에 시각적 이미지를 이용한 공격인 Recording Attack에

대응할 수 있다.

Shoulder Surfing Attack과 달리 비가시화와 난독화에 높은 안전성이 아닌 중간-높은 안전성을 부여한 이유는 시각, 촉각을 이용한 잠금 패턴 입력에 시각적 안전성은 완벽히 대응할 수 없기 때문이다.

[표 4-3] Recording Attack에 의한 잠금 패턴 안전도 비교

	Default	Visible	Invisible	Obfuscation
Recording Attack	Super-Low	Middle	Middle High	Middle High

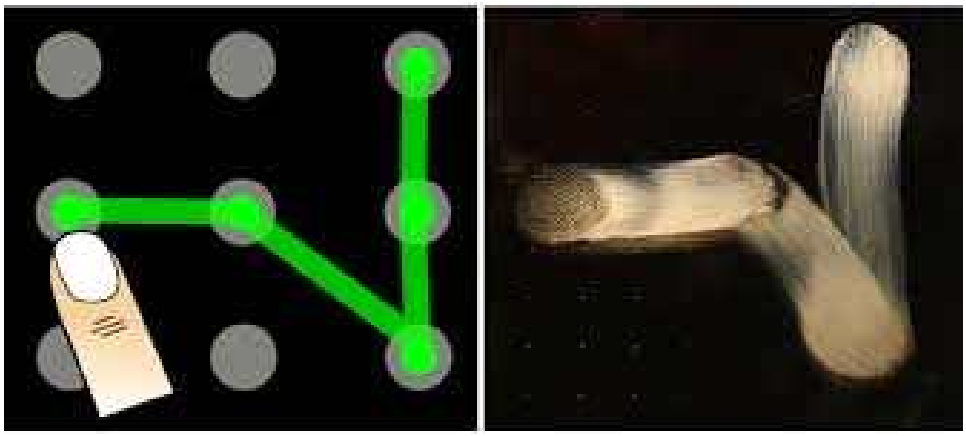
4) Smudge Attack

Smudge Attack이란 비밀번호를 입력하기 위해 번호를 터치 혹은 드래그할 때 핸드폰 화면에 남는 지문의 흔적을 이용하여 비밀번호(잠금 패턴)을 탈취하는 방법이다.

[2][그림 4-3]은 Smudge Attack에 활용되는 모습의 예시이다. 드래그 된 패턴의 시작과 끝 지점의 지문 모양을 보면 어디서부터 시작했는지 유추할 수 있다. 오른쪽 그림을 보면 시작과 끝 지점은 [1, 3] 혹은 [2, 1]이다. [1, 3]은 지문의 형상이 있는 것이 아닌 드래그로 인하여 뭉개진 모습을 볼 수 있으며, [2, 1]은 거의 완벽한 지문의 형상을 확인할 수 있다.

잠금 패턴을 그릴 때 한번 패턴을 그리기 시작하면 끝 지점에 도달할 때 까지 손가락을 분리할 수 없다. 따라서 시작 지점에서의 지문은 다음 지점으로 이동하기 위해 뭉개질 수밖에 없으며, 끝 지점에서는 다음 지점으로 이동할 필요 없이 손가락을 화면에서 분리하면 되기 때문에 거의 완벽한 지문의 형태가 보존된다. 그렇기에 [1, 3] → [2, 3] → [3, 3] → [2, 2] → [2, 1]로 드래그 됐음을 알 수 있다.

2) https://www.researchgate.net/figure/a-Shoulder-surfing-attack-b-smudge-attack_fig1_330934832 //'19.11.30. Hamaad Rafique 연구진의 공격 예시 이미지 참조



[그림 4-3] Smudge Attack 예시

Smudge Attack은 사용자가 잠금 패턴을 그리고 남은 흔적을 이용한 공격이다. 잠금 패턴을 그리고 나면 물리적인 흔적이 남게 되며, SW적으로는 물리적인 흔적을 지울 수 없기 때문에 완벽하게 대응할 수 없다. 물리적으로 흔적을 지우는 행위 외에 SW적으로 비교적 안정화시키는 방법에 대해 [표 4-4]와 같이 비교해보겠다.

물리적으로 남은 잠금 패턴을 물리적으로 지우는 것이 가장 안전하다는 기준으로 볼 때 기존의 잠금 패턴의 경우 [그림 4-2]의 오른쪽과 같이 양 끝 포인트의 형성 된 지문 모양을 이용하여 어떤 지점에서 시작했는지까지 구분이 되기 때문에 단 한 번의 시도만으로 공격자는 완벽하게 비밀번호를 탈취할 수 있다.

그러나 본 논문에서 제안하는 잠금 패턴의 경우 물리적으로 남아 있는 패턴 모양을 이용해 사용자가 잠금 패턴을 유추할 수 있으나 각 포인트별로 입력되는 진동의 세기를 유추하는 것은 완벽하게 불가능하다. 따라서 Smudge Attack만을 이용한 공격을 할 경우엔 각 포인트당 3^n 만큼의 보안 강도를 올릴 수 있기 때문에 가능한 많은 포인트를 사용하는 것이 최대 경우의 수를 늘릴 수 있어 보다 안전해진다.

[표 4-4] Smudge Attack에 의한 잠금 패턴 안전도 비교

	Physical Cleaning	Default	Visible	Invisible	Obfuscation
Smudge Attack	Super High	Super Low	Middle High	Middle High	Middle High

제 2 절 사용자 편의성 비교

본 논문에서 제안하는 기법에 대해 실제 구현하여 20대 10명, 30대 10명, 40대 10명 그리고 50대 10명 총 40명의 사용자를 기반으로 잠금 패턴 등록, 등록한 잠금 패턴을 이용한 로그인, 기억하기 편한 잠금 패턴 그리고 잠금 패턴 입력 속도에 대하여 테스트 진행하였으며, 그린 패턴은 [그림 3-2]의 좌측 그림과 같이 'Z' 모양의 패턴을 이용하였다. 구현 환경은 [표 4-5]와 같다.

[표 4-5] 잠금 패턴 구현 및 테스트 환경

Device	Nexus 5X
OS	Android
Version	API 28
Display	420dpi

1) Password Registration

기본 잠금 패턴과 제안하는 잠금 패턴 별 잠금 패턴 등록 편리성에 대해 [표 4-6]과 같이 비교해보겠다.

잠금 패턴을 사용함에 있어 우선 핸드폰에 사용자 중심 잠금 패턴을 등록 시켜야한다. 대중화 되어 있는 잠금 패턴의 등록하는 방법은 사용자들 모두 알고 있기 때문에 별도의 안내 없이 잠금 패턴을 등록하는 순간부터 완료할 때까지에 대한 편리성을 측정했으며 본 논문에서 제안하는 잠금 패턴은 사전

지식 없이는 어떻게 사용하는지 모르기 때문에 가시화, 비가시화, 난독화 잠금 패턴에 대해 설명한 이후 각각 3번씩 연습 테스트를 진행한 후 편리성을 측정하였다.

[표 4-6]을 보면 20대와 30대, 40대의 경우 본 논문에서 제안하는 잠금 패턴 3가지 모두 사용하는데 특별히 어려움은 없다. 그러나 50대의 경우 비가시화와 난독화에 있어 사용법에 대하여 설명했음에도 불구하고 본인이 잘못 패턴을 그렸거나 핸드폰에서 발생한 에러로 착각하는 경우가 다수 있어 사용에 어려움을 보였다.

[표 4-6] Password Registration에 따른 세대별 사용 편리성 평가

	Default	Visible	Invisible	Obfuscation
20's	Easy	Easy-Middle	Middle	Middle
30's	Easy	Easy-Middle	Middle	Middle
40's	Easy	Middle	Middle-Hard	Middle-Hard
50's	Easy	Middle-Hard	Hard	Hard

2) Login Process

기본 잠금 패턴과 제안하는 잠금 패턴 별 로그인 과정의 잠금 패턴 입력 속도에 대해 [표 4-7]과 같이 비교해보겠다.

4-2-1에서 등록한 잠금 패턴을 이용하여 로그인하는 과정을 실험하겠다. 20대와 30대, 40대의 경우 등록했던 잠금 패턴을 이용하여 로그인 하는데 별다른 어려움 없이 로그인 할 수 있었지만, 본 논문에서 제안하는 잠금 패턴에서 잠금 패턴을 그리는 것과 독립적으로 포인트 별 진동의 세기를 추가적으로 입력해야하기 때문에 기존의 잠금 패턴보다 시간이 더 걸림을 확인할 수 있다.

그러나 50대의 경우 4-2-1과 마찬가지로 새로운 잠금 패턴을 사용함에 있어 어려움을 느끼는 사용자가 있어 별도의 교육이 필요함을 알 수 있었다.

[표 4-7] Login Process에 따른 세대별 평균 성능 평가

	Default	Visible	Invisible	Obfuscation
20‘s	2.7s	7.3s	9.6s	9.5s
30‘s	2.7s	7.8s	8.7s	10.1s
40‘s	2.9s	8.3s	9.7s	10.4s
50‘s	3.4s	10.4s	12.3s	12.7s

3) Easy to Remember

기본 잠금 패턴과 제안하는 잠금 패턴 별 사용자 입장에서 기억하기 쉬운 것에 대해 [표 4-8]과 같이 비교해보겠다.

기존의 잠금 패턴의 경우 단순히 패턴을 어떻게 그리는지만 기억하면 되기 때문에 사용자 입장에서 기억하기 단순하다는 장점이 있다. 그러나 제안하는 잠금 패턴의 경우 그리는 패턴뿐만 아니라 각 포인트 별 진동의 세기를 기억해야하기 때문에 포인트 수가 늘어날수록 사용자 입장에서는 기억해야할 것이 늘어나게 된다.

[표 4-8] Easy to Remember에 따른 세대별 성능 평가

	Default	Visible	Invisible	Obfuscation
20‘s	Easy	Middle	Middle	Middle
30‘s	Easy	Middle	Middle	Middle
40‘s	Easy	Middle-Hard	Middle-Hard	Middle-Hard
50‘s	Easy	Hard	Hard	Hard

4) Typing Speed

기본 잠금 패턴과 제안하는 잠금 패턴 별 패턴 입력 속도에 대해 [표 4-9]와 같이 비교해보겠다.

20대와 30대의 경우 잠금 패턴을 이해하고 사용하는데 [표 4-9]와 같이 비슷한 시간이 걸렸으며, 40대와 50대보다 4가지(기본, 가시화, 비가시화 그리고 난독화 잠금 패턴) 비교 항목보다 비교적 편리한 잠금 패턴 등록 시간

이 걸렸다. 등록하는 시간이 각기 다른 이유는 각 포인트별 사용자가 등록한 진동의 세기가 언제 나오느냐에 따른 차이로 보인다.

[표 4-9] Typing Speed에 따른 세대별 평균 성능 평가

	Default	Visible	Invisible	Obfuscation
20‘s	2.3s	6.7s	8.6s	8.5s
30‘s	2.9s	7.3s	8.7s	11.1s
40‘s	2.9s	8.7s	8.7s	12.4s
50‘s	3.7s	11.2s	13.1s	13.7s

제 5 장 결론 및 향후 개선 방안

본 논문에서는 대중적으로 사용하는 잠금 패턴의 취약성을 파악하고 보안 필름과 같이 별도의 장비를 사용하지 않으면서 Bruter-Force Attack, Shoulder Surfing Attack, Recording Attack 그리고 Smudge Attack에 대응할 수 있는 새로운 잠금 패턴을 제안하였다.

특히 대중적으로 사용하고 있는 안드로이드 핸드폰의 잠금 패턴은 Brute-Force Attack을 이용할 때 총 389,112번의 경우의 수로 사용자가 등록한 잠금 패턴을 탈취할 수 있다. 그러나 본 논문에서 제안하는 랜덤하게 입력되는 시간차를 활용한 잠금 패턴의 경우 총 3,872,929,464번의 경우의 수를 시도해야 잠금 패턴을 탈취할 수 있다. 기존보다 무려 약 9000배 향상 된 보안 강도를 자랑한다.

또한 기존의 잠금 패턴은 사용자가 핸드폰에 패턴을 드래그하면 핸드폰 UI에 드래그한 모양이 그려지게 되어 공격자가 핸드폰, 카메라 등을 이용하여 사용자가 잠금 패턴을 드래그하는 모습을 영상 기록물로 남기게 되면 100% 잠금 패턴을 탈취 당하게 된다. 그러나 본 논문에서 제안하는 잠금 패턴은 잠금 패턴의 시각적 효과 외에 핸드폰을 들고 있는 사용자만이 느낄 수 있는 진동의 세기를 이용하여 Shoulder Surfing Attack, Recording Attack 그리고 Smudge Attack에 완벽히 대응할 수 있다.

Bruter-Force Attack, Shoulder Surfing Attack, Recording Attack 그리고 Smudge Attack에 대한 기존 잠금 패턴과 제안하는 잠금 패턴의 보안 강도를 비교해보면 [표 5-1]과 같다. 가장 안전한 잠금 패턴 기법은 비가시화와 난독화 잠금 패턴이며, 다음으로는 가시화 잠금 패턴 마지막으로 가장 안전하지 않은 잠금 패턴은 기본 잠금 패턴이다.

[표 5-1] 각기 다른 공격 방법에 의한 잠금 패턴 안전성 비교

	Default	Visible	Invisible	Obfuscation
Brute-Force Attack	Low	Middle High	Middle High	Middle High
Shoulder Surfing Attack	Low	Middle	High	High
Recording Attack	Super Low	Middle	Middle High	Middle High
Smudge Attack	Super Low	Middle High	Middle High	Middle High

사용자 편의성 관점에서 각 잠금 패턴에 대해 비교했을 때는 [표 5-2]와 같다. [표 5-2]는 4장 2절에서 비교한 세대별 사용자 편리성을 세대별이 아닌 전체 평균 내어 나타낸 값으로 Password Registration의 경우 기본 잠금 패턴이 가장 사용하기 편리하며, 비가시화와 난독화가 사용자 입장에서 가장 사용하기 어렵다. Login Process의 경우 기본 잠금 패턴이 2.9초로 가장 빠르게 잠금을 해제할 수 있었으며 난독화가 10.6초로 가장 늦게 잠금을 해제할 수 있었다. Easy to Remember의 경우 기본 잠금 패턴이 가장 기억하기 쉬웠으며 가시화, 비가시화, 난독화 모두 중간-어려움의 강도를 보였다. 마지막으로 Typing Speed는 단순히 선만 그으면 되는 기본 잠금 패턴이 2.9초로 가장 빨랐으며, 난독화가 11.4초로 가장 느리게 잠금 패턴을 해제하였다.

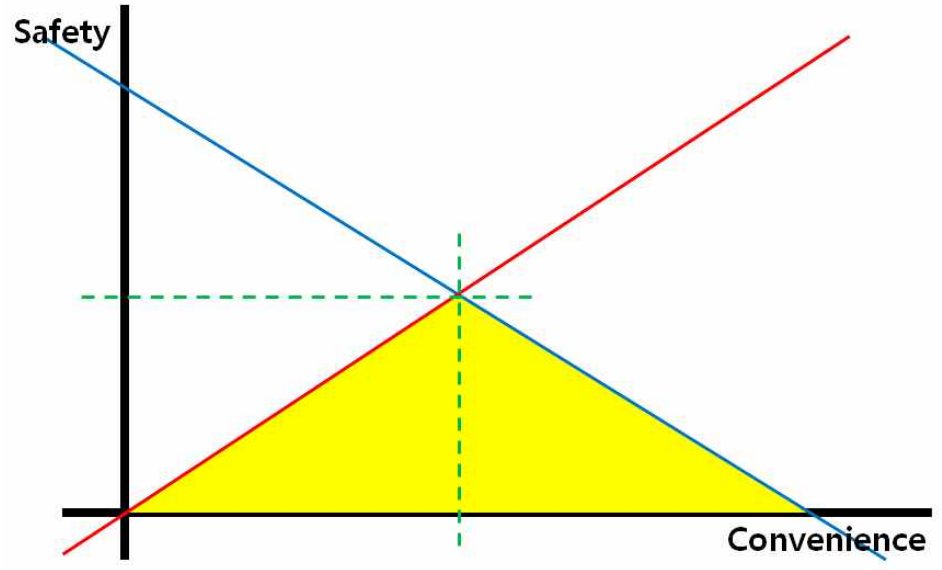
[표 5-2] 각기 다른 공격 방법에 의한 사용자 편리성 비교

	Default	Visible	Invisible	Obfuscation
Password Registration	Easy	Middle	Hard	Hard
Login Process	2.9s	8.4s	10.0s	10.6s
Easy to Remember	Easy	Middle-Hard	Middle-Hard	Middle-Hard
Typing Speed	2.9s	8.4s	9.7s	11.4s

보안 강도로만 본다면 비가시화와 난독화 잠금 패턴이 가장 우수하였지만 사용자 편의성 관점에서만 본다면 기존 잠금 패턴이 가장 우수하다. 그러나 기존 잠금 패턴은 4장 1절에서 보여준 공격 기법에 대해 전혀 대응을 하지 못한다.

이는 보안 강도를 높여 안전하다고 하여 사용자 입장에서 무조건 적으로 좋은 기술은 아니다. [그림 5-1]을 보면 안전성과 편리함은 한쪽이 증가하면 같이 증가하는 것이 아닌 어느 시점을 지나서면 떨어지게 된다. 즉 편리함이 증가하면 안전성은 줄어들게 되며, 안전성이 증가하면 편리함이 줄어들게 된다. 따라서 어느 한쪽에도 치우치지 않게 양쪽에서 모두 이득을 취할 수 있는 지점을 찾는 것이 중요하다.

따라서 잠금 패턴을 사용하는 환경과 사용자를 고려하여 적절한 기법을 선정하는 것이 사용자 관점에선 중요하다고 볼 수 있다. 가시화 잠금 패턴이 나머지 2개보다 사용자 편의성이 우수하기 때문에 다. 가시화 잠금 패턴으로 먼저 본 논문에서 제안하는 기법을 학습한 이후 비가시화, 난독화 순으로 사용해 보는 것을 제안한다.



[그림 5-1] 안전성과 편리함의 거래

참 고 문 헌

1. 국외문헌

B. Hoanca, K. Mock, (2006). *Secure graphical password system for high traffic public areas*. Proc. ACM symposium on eye tracking research & applications, 35.

F. Mohsen, M. Shehab, (2013). *Android keylogging threat*. Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on. IEEE, 545-552.

F. Tari, A. Ozok, SH. Holden, (2006). *A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords*. ACM symposium on usable privacy and security (SOUPS), 56-66.

J. Angulo, E. Wastlund, (2012). *Exploring touch-screen biometrics for user identification on smart phones*. Privacy and identity management for lifeIFIP advances in information and communication technology, vol. 375, 130-143.

J. Bonneau, (2012). *The science of guessing: analyzing an anonymized corpus of 70 million passwords*. Security and Privacy (SP), IEEE, Symposium, 538-552.

J. Thorpe, PC, Oorschot van, (2007). *Human-seeded attacks and exploiting hot-spots in graphical passwords*. USENIX security symposium, 103-118.

K. Bicakci, NB. Atalay, M. Yuceel, H. Gurbaslar, B. Erdeniz, (2009). *Towards usable solutions to graphical password hotspot problem*. Proc. IEEE annual international computer software and applications conference, vol. 2, 318-323.

K. Taekyoung, N. Sarang, (2014). *TinyLock: Affordable defense against smudge attacks on smartphone pattern lock systems*. Computers & Security, 42, 137-150.

P. Andriotis, T. Tryfonas, G. Oiknomou, (2014). *Complexity metrics*

and user strength perceptions of the pattern-lock graphical authentication method. International Conference on Human Aspects of Information Security, Privacy, and Trust, 115-126.

P. Andriotis, T. Tryfonas, G. Oikonomou, C. Yildiz, (2013). *A pilot study on the security of pattern screen-lock methods and soft side channel attacks*. In Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks, 1-6.

P. Golle, D. wagner, (2007). *Cryptanalysis of a cognitive authentication scheme*. IEEE symposium on security and privacy, 66-70.

S. Wiedenbeck, J. Waters, JC. Birget, A. Brodskiy, N. Memon, (2005). *Authentication using graphical passwords: effects of tolerance and image choice*. ACM SOUPS (Symposium on usable privacy and security), 1-12.

S. Wiedenbeck, J. Waters, L. Sobrado, JC. Birget, (2006). *Design and evaluation of a shoulder-surfing resistant graphical password scheme*. ACM international working conference on advanced visual interfaces, 177-184.

AJ. Aviv, K. Gibson, E. Mossop, M. Blaze, JM. Smith, (2010). *Smudge attacks on smartphone touch screens*. 4th USENIX conference on Oensive technologies, 1-7.

ABSTRACT

Improved security of lock pattern using random vibration intensity

An, Kyu-Hwang

Major in IT Convergence Engineering

Dept. of IT Convergence Engineering

The Graduate School

Hansung University

The pattern lock applied to smartphones is a simple password input method that replaces the password PIN input method, and is widely used in most smartphones. However, it is vulnerable to Smudge Attack due to handprints generated by dragging a password by entering pattern recognition on a commercially available smartphone, and is a fixed point of the Shoulder Surfing Attack. Therefore, this paper proposes a new method of pattern recognition. The same pattern is generated by generating the strength of the vibration of the point input by the password among the strength of vibration (strong, medium, weak) randomly generated for each point, instead of simply recognizing the shape of the pattern when the drag is input. Even if you draw it, the security strength per point is improved. Therefore, in the case of the existing lock pattern, a total of 389,112 patterns can be drawn, but the

lock pattern proposed in this paper can draw 3,872,929,464 patterns, which is about 9,953 times larger. This means that even if the same 'Z' pattern is drawn, the strength of each point's vibration must be entered, so the security strength is improved as much as in the Smudge Attack of the 'Z' pattern and the Shoulder Surfing Attack. Therefore, we propose a new pattern recognition method that makes a completely different pattern even when dragging the same 'Z' pattern.

KEYWORD: Drag, Smartphone, Shoulder Surfing Attack, Smudge Attack, Pattern Lock