情報保護學會論文誌
第19卷 第5號, 2009. 10

행렬 상에서 문자 간 연산을 수행하는 패스워드 인증 기법*

강 전 일,[1†] 맹 영 재,[1] 양 대 헌,[1‡] 이 경 희,[2] 전 인 경[3]
[1]인하대학교, [2]수원대학교, [3]한국인터넷진흥원

Password Authentication Scheme based on Operation of Alpha-numeric Characters on Matrix*

Jeonil Kang,[1†] YoungJae Maeng,[1] DaeHun Nyang,[1‡]
KyungHee Lee,[2] Inkyung Jeun[3]
[1]INHA University, [2]University of Suwon, [3]Korea Internet & Security Agency

요 약

패스워드는 그 자체만으로도 낮은 복잡도를 가지고 있을 뿐만 아니라, 안전하지 않은 환경에서 그대로 키보드와 같은 입력 장치를 통하여 입력하는 행위는 훔쳐보기와 같은 공격으로 쉽게 노출될 수 있다. 이러한 문제를 극복하고자 사용자 비밀의 형태를 다른 것으로 바꾸거나 복잡한 입력과정을 통하여 입력을 수행하는 방법들이 제안되어 왔으나, 보안성과 사용자 편의성에 있어서 적합한 타협점을 찾지 못하고 있다. 이 논문에서는 행렬 상에 문자들 사이에서의 연산을 통하여 기존 형태의 패스워드를 비밀로 사용하는 인증 기법에 대해서 소개한다. 다양한 각도에서의 분석을 통하여 기법이 갖는 안전성을 보이고, 사용자 실험을 통하여 사용자들이 실제로 느끼는 기법에 대한 어려움 등을 확인할 것이다.

ABSTRACT

Besides the passwords have low complexity, they can easily be revealed by the shoulder-surfing attack when they are inputted through the input devices such like keyboard. To overcome these problems, many new authentication schemes, which change the user secret different form or let users input their secrets through the more complex manners, have been suggested, but it is still hard to find the balanced point between usability and security. In this paper, we introduce a new authentication scheme that use the traditional alpha-numeric password as user secret based on operation of them on matrix. We show the security strength of our proposal through the analyses in the various aspects and confirm the difficulty that users feel from our proposal through the user study.

Keywords: Password Authentication, Alternative Password, Shoulder-surfing Resistance

I. 서 론

패스워드를 사용한 사람의 인증은 패스워드는 그 사람만이 알고 있는 비밀이라는 가정 아래에서 지금까지 널리 사용되고 있다. 패스워드의 가장 큰 장점은 사람이 무언가를 휴대할 필요가 없이 손쉽게 인증 과정을 수행할 수 있다는 것이다. 반면, 사람은 패스워

접수일(2009년 7월 30일), 게재확정일(2009년 10월 9일)
* 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 대학중점연구소 지원 사업으로 수행된 연구임. (2009-0074476)
† 주저자, dreamx@isrl.kr
‡ 교신저자, nyang@inha.ac.kr

드를 기억하기 쉽도록 의미 있는 단어의 조합으로써 만들려는 경향이 강하고 이 때문에 패스워드의 복잡도가 낮아지는 문제가 있다. 이미지를 알파벳과 숫자 대신 사용하는 Déjà Vu[1]나 PassPoints[2], PassIcons[3]와 같은 많은 그래픽 패스워드(graphical password) 기법들은 복잡도가 높은 정보를 더 오랫동안 기억하기 쉽도록 하는 방법을 제안픽 패스한 편, 패스워드 인증은 기억하가 비밀이 되는 정보를 키패드 등에 직접 입력함으로써, 키패드나 시스템 외부에서 정보를 취할 수 있는 공격자에게 취약한 면을 가지고 있다. Matsumoto와 Imai의 연구[4]나 Hopper와 Blum의 연구[5]는 이러한 외부 공격자에게 안전하게 패스워드를 입력하는 방법에 대한 것이다. 최근에는 그래픽적인 요소를 이용하여, 키로거(Key Logger)나 훔쳐보기(Shoulder-surfing, 또는 Peeping)와 같은 외부 공격에도 안전하면서도 입력하기 쉬운 방법에 대한 연구가 진행되고 있다[6-8].

더하기나 곱하기, 범위 제한과 같은 간단한 연산을 사용자에게 수행하게 하는 기법들의 경우 숫자를 기본으로 사용한다. 때문에 패스워드는 숫자로만 이루어져야 하거나, 숫자로 변환이 가능해야한다. 연산의 어려움을 덜기 위하여 그래픽적인 요소를 사용하는 방법들의 경우 인증에 사용되는 비밀은 사용자가 선택한 이미지가 된다. 현재 인터넷에서 널리 사용되는 패스워드 인증 프로토콜이 알파벳과 숫자로 이루어진 문자(Alpha-numeric)인 것을 생각한다면, 기존의 패스워드를 유지한 채 새로운 기법을 적용하기가 쉽지 않음을 알 수 있다. H. Zhao 등의 S3PAS[9]와 X. Bai 등의 PAS[10], Z. Zheng 등이 제시한 기법[11] 등은 패스워드에 텍스트를 이용하여 이러한 지적에 있어서 어느 정도 자유로울 수 있었다.

그러나 여전히 재전송 공격(Replay Attack)과 인증 세션 간 교차 공격(Intersection Attack) 사이에서 발생하는 타협점은 풀기 힘든 것으로 알려져 있다. 사용자가 수행해야 하는 연산이 간단해야 하는 이러한 유형의 인증 기법들에서 하나의 성공한 인증 세션으로부터 공격자는 비밀을 알아내지 못하였다 하더라도 일정한 양의 정보를 알아낼 수 있다. 공격자가 얻어낸 일정한 양의 정보가 누적이 되면 최종적으로 여러 인증 세션들로부터 사용자의 비밀을 알아낼 수 있게 된다. 세션 사이에서 노출되는 정보의 양이 적을 경우, 공격자는 재전송 공격에 성공할 확률이 높아지게 된다.

이 논문에서는 행렬 위의 배치를 통하여 문자 간 연산을 지원하여 패스워드를 인증하는 기법에 대해서 소개한다. 제안하는 기법은 현재 사용되고 있는 영문자와 숫자 조합의 패스워드를 그대로 사용여 패스워으며, 작은 화면 표시 공간의 사용으로 다양한 기기에 사용될 패스워도록 고안되었다. 2장에서는 제안하는 기법과 유사한 연구들에 대한 소개와 이러한 기법들이 가진 특성을 분석하고 문제점과 이에 대한 해결책이 무엇이 있을 패스워드지 확인한다. 3장에서는 제안 기법에 대한 기본적인 절차와 다양한 선택 사항에 대해서 기술하고, 4장에서는 파라미터에 따른 보안성을 분석하여 적합한 파라미터를 선택여 패스워도록 배려하였다. 5장에서는 제안한 기법에 대해서 사용자 실험을 통하여 기법이 갖는 사용성과 편의성에 대해서 확인하였고, 6장에서는 보안성, 사용자 편의성들을 다른 기법과 비교하였다. 7장에서는 제안하는 기법에 대한 결론과 앞으로 그대로 내용을 포함하고 있다.

II. 관련 연구

2.1 유사 기법 소개

복잡도가 높은 비밀을 오랫동안 기억하게 하도록 하는 인증 기법들이나, 외부 공격에 안전하도록 하는 인증 기법들은 그 수가 매우 많으나, 이 논문에서는 비밀을 알파벳과 숫자로 이루어진 패스워드를 사용하는 기법들에 대해서만 소개하기로 한다.

H. Zhao 등이 제안한 S3PAS[9]는 S. Wiedenbeck 등이 제안한 PassIcons[3]을 문자 형태로 변형한 것이다. S3PAS에서 화면에는 10×10 행렬에 문자가 무작위로 출력되고, 사용자는 자신의 패스워드를 3자리씩 끊어서 화면에 각각의 패스워드 문자를 꼭짓점으로 하는 삼각형을 머릿속으로 떠올린다. 사용자는 해당 삼각형의 내부의 임의의 한 문자를 선택하여 키보드나 마우스를 이용하여 해당 문자를 입력한다. 이를 순차적으로 패스워드의 모든 부분에 대해서 수행한다. 예를 들면, 패스워드가 'abcde'라면, 'abc', 'bcd', 'cde', 'dea', 'eab'에 대해서 수행한다. 검증자(Verifier)는 각각의 입력을 확인하고 모두 유효한 입력이라면 사용자가 패스워드를 알고 있다고 가정한다. S3PAS와 PassIcons의 가장 큰 차이점은 사용자가 가지고 있는 비밀이 '순차적인 요소를 가지고 있는 중복 가능한 문자열'과 '순서에 관계없는 서로 다른 아이콘의 집합'이라는 것이다. 순차적인 요소는

인증 세션을 바라보고 있는 공격자에게 더 많은 정보를 알려주게 된다. ('abc', 'bcd'의 입력에서 'b', 'c'는 동일한 꼭짓점이 된다.) 이러한 정보는 공격자에게 비밀을 알아내는 데 도움을 준다. 또한 PassIcons와 같은 문제로써 공격자가 무작위로 입력을 선택하여도 인증을 성공할 확률이 다른 기법들에 비해 큰 문제가 있다.

X. Bai 등이 제안한 PAS[10]는 비밀로부터 검증자로부터 전달되는 무작위 수에 따라 술어(Predicate)를 생성하고 이를 이용하여 인증을 수행한다. 이 기법에서 사용자는 특별한 형태(표 안의 셀을 지정하기 위한 정수와 알파벳으로만 이루어진 문자열)를 가진 비밀을 최소 두 개를 기억해야 한다. 검증자는 무작위로 선택한 정수와 무작위로 선택한 문자가 들어 있는 표들, CAPTCHA로 만든 추가적인 표, 이 셋을 사용자에게 준다. 사용자는 자신의 비밀로부터 술어를 생성하고 술어를 이용하여 문자가 들어 있는 표들을 뒤져 술어와 일치하는 지 확인한다. 이렇게 확인된 정보를 다시 이용하여 CAPTCHA로 만든 표에 적합한 값을 찾아 입력한다. 예를 들어, 비밀이 '23 sente'와 '41 logig'였고, 검증자가 선택하여 준 정수가 '15'였을 때, 사용자는 '23e'와 '41g'를 술어로 만든다. 사용자가 받은 문자열이 적힌 표 두 개에서 각각 (2,3)셀과 (4,1)셀을 뒤져 'e', 'g'가 있는 지 확인하고 그 결과로써 'yes yes', 'yes no'와 같은 정보를 만들어낸다. 이 후, 이 정보를 이용하여 CAPTCHA가 들어 있는 표에 해당 'yes yes', 'yes no'에 속하는 셀에 적힌 글자를 읽어 이를 입력한다. 이 기법에서는 SAT 계산기(solver)에 기존의 기법들이 모두 약하다는 사실을 지적하고 자신들의 기법이 SAT 계산기에 더 안전할 수 있음을 주장하고 있다. 그러나 여전히 그 증가된 안전성이 미미할뿐더러 비밀을 특별한 형태로 두 가지 이상을 기억하고 있어야 하는 것은 이 논문에서 목표로 하고 있는 패스워드의 형태와 맞지 않다.

Z. Zheng 등이 제안한 기법[11]은 사용자가 패스워드에 해당하는 획(Stroke, 여기서는 표에서의 셀의 순서로 정의된다)을 검증자에 저장하고, 검증자로부터 전달한 무작위 표에서 획의 순서에 따라 해당 셀을 읽어서 입력하는 방식이다. 검증자의 입장에서 패스워드는 표 위에 지정된 획 정보가 되지만, 사용자 입장에서는 문자열로 된 패스워드로부터 지정했던 획을 기억하는 방식이다. 그러나 사실, 이 방식은 DAS(Draw-A-Secret)[12]를 키보드로 입력하게 하여 특정한 셀을 선택하는 것을 보이지 않게 한 방식으로 볼 수 있어, 역시 이 논문에서 원하는 패스워드의 형태와 다르다고 할 수 있다.

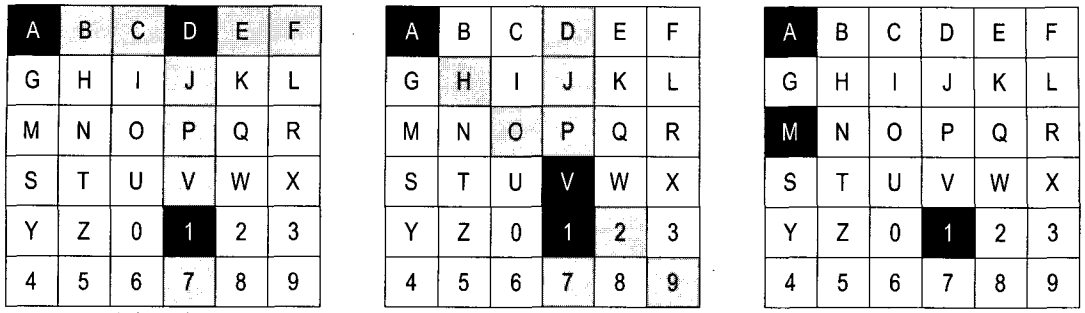
2.2 일반적인 접근 방법론에 대한 이해

앞서 소개한 몇몇 기법 이외에 더 많은 기법들이 안전한 인증 기법을 만들기 위하여 사용하는 접근 방법론은 크게 몇 가지로 생각해볼 수 있다.

• 검증자로부터 주어지는 무작위성: 재전송 공격을 막기 위하여 보통의 기법에서 검증자는 무작위로 선택된 값이 전송된다. 사용자는 이 값을 이용하여 자신의 비밀과 혼합하여 모종의 연산을 거친 뒤 그 결과를 검증자에게 입력으로써 돌려준다. 이 과정에서 공격자는 검증자로부터 전달되는 값(Challenge)과 그에 대한 사용자의 응답(Response)을 볼 수 있다. 당연하게도 이 과정에서 비밀에 관한 정보가 노출되게 된다. 공격자는 여러 인증 세션을 관찰하여 사용자의 비밀을 얻어낼 수 있다. 전달되는 값이 동일하면 공격자는 사용자의 응답을 재전송함으로써 인증을 무사히 통과할 수 있기 때문에, 전달되는 값이 완전한 무작위이거나 동일해서는 안 되는 딜레마에 빠지게 된다.

• 사용자가 선택하는 무작위성: 사용자가 의도하거나 의도하지 않은 무작위성은 공격자에게 전해지는 정보에 오류를 섞어 공격자에게 비밀을 알아내기 어렵게 하는 효과를 가지고 있다. 예를 들면, HB[5]에서 발생하는 사람이 저지르는 실수는 공격자에게 전해지는 정보에 오류를 삽입하여 공격자에게 LPN(Learning Parity with Noise) 문제로 나타나게 된다. Matsumoto의 연구[4]에서는 사용자가 미리 앞서 무작위성을 가진 비밀을 하나 더 가지고 있고 최종 입력을 만들 때 사용자가 무작위로 값을 할당하는 부분이 있다. 주의해야 할 것은 사용자가 선택하는 무작위성은 안전성에 크게 영향을 주긴 하지만, 높은 무작위성을 보장하진 않는다는 것이다.

• 비밀의 추상화(Abstraction): 몇몇 기법들은 충분히 납득할만한 수준의 보안성을 갖도록 하기 위해서 자신들의 기법에서 사용하는 비밀의 크기를 크게 늘리는 선택을 하고, 비밀의 추상화 과정을 거쳐 인증에 사용한다. 검증자 입장에서는 납득할만한 수준의 비밀을 사용자가 가지고 있음을 확인해야 하기 때문에, 과도한 추상화는 사용자에게 더 많은 인증 시간을 요구하게 될 수밖에 없다. 검증자가 확신하는 비밀 수준을 낮추면 상대적으로 사용자의 인증 시간이 짧아지



〔그림 1〕 행렬을 이용한 문자 간 * 연산의 예. (a) 'A'의 행과 'I'의 열의 교차점 = 'D', (b) 'A'의 대각선과 'I'의 열의 교차점 = 'V', (c) 'I'로부터 'A'로부터 'I'까지의 거리(오른쪽 3칸, 아래쪽 4칸)만큼 떨어진 문자 = 'M'

게 되며, 동시에 공격자에게 검증자에게 낮아진 비밀 수준만큼 어렵게 하는 효과가 있다. 때문에 비밀의 추상화 기법은, 안전성을 정교하게 제어할 때 사용될 수 있다.

• 컴퓨터 프로그램에 대한 인공 지능 문제: CAPTCHA와 같은 인공 지능 문제를 컴퓨터에 의한 외부 공격을 매우 어렵게 한다. 하지만 컴퓨터가 사용자의 비밀을 모두 얻어 공격자에게 바로 전달해주는 경우가 아니라면 이러한 방법은 크게 의미를 갖지 못한다고 할 수 있다.

많은 기법들은 위와 같은 접근 방법을 혼합하여 사용하고 있고, 어떠한 접근 방법을 사용하느냐에 따라 특정 부분에 대해서 많은 장점을 갖기도 하지만 다른 부분에 대해서는 단점을 드러내기도 한다. 예를 들어, 사용자의 편의를 위해서 기법을 지나치게 간단하게만 만들 경우 보안성이 크게 떨어지고, 보안성을 높이기 위한 높은 무작위성이 세션 교차 공격에 더 취약하게 만드는 등의 문제가 생긴다.

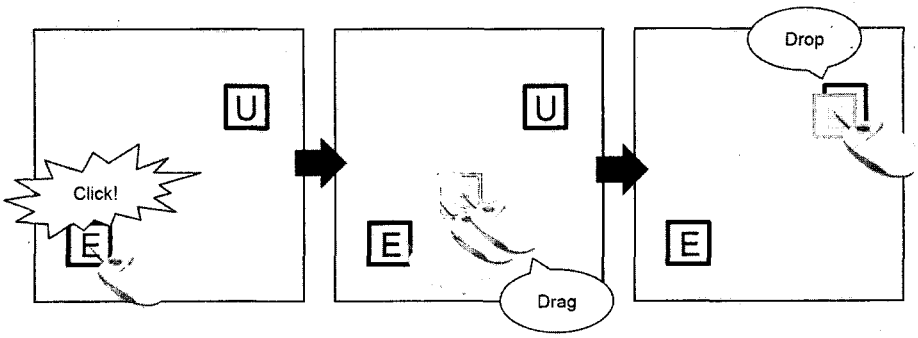
III. 행렬 위 문자 간 연산을 이용한 인증 기법

3.1 문자 간 연산

이 논문에서 제안하는 기법은 현재 가장 많이 사용하고 있는 알파벳과 숫자로 이루어진 패스워드를 비밀로 사용하면서 이를 서버와 같은 검증자에게 외부 공격에 자유롭지 않은 환경에서 안전한 인증을 수행하는 것을 목표로 한다. 사용자와 검증자는 패스워드 형태의 비밀을 공유하고 있으며, 검증자가 가지고 있는 비밀은 안전하다고 가정한다. 그러나 패스워드를 이루는 문자 사이에서는 일반적으로 사칙 연산을 정의하기가 매우 어렵다. 예를 들어, 자연수에서의 사칙연산 '1+2'의 결과가 '3'이 되는 것과 달리, 문자 간의 연산 'F+V'의 결과가 무엇이 될지는 아무도 모른다. 이는 패스워드에 대해서는 Matsumoto의 연구〔4〕나 HB〔5〕와 같은 기법이 적용될 여지가 별로 없음을 알려준다. 그러나 행렬을 이용하고 행렬 위에 위치한 문자 간 상대 위치를 이용하면, 이러한 문자 간의 연산도 특별한 훈련 과정이 없이도 쉽게 수행할 수 있다.

〔그림 1〕은 이러한 행렬을 이용한 문자 간 연산이 비교적 쉽고 다양한 형태를 가질 수 있음을 보여준다. 위의 예들에서는 연산의 결과로써 특정한 문자가 선택된다. 연산의 결과가 다시 문자일 경우, 이에 대해서 다른 연산을 수행할 수 있는 장점이 있다는 점은 주목할 만하다.

〔그림 2〕는 문자 간 상대적 위치를 이용하여 마우스로 연산의 결과를 입력하는 방법에 대해서 예를 들고 있다. 그림에서는 마치 사용자가 특정한 문자를 클릭하고 이를 다른 문자 위로 올려놓는 것처럼 예시되고 있으나, 실제 구현에서는 행렬 위로 마우스 포인터가 올라가게 되면 마우스 포인터가 사라지게 되어 클릭을 한다고 하더라도 사용자는 자신이 무엇을 클릭했는지 확신할 수 없게 된다. 이러한 경우 마우스의 이동에 따라 움직이는 행렬과 고정된 행렬, 이렇게 이중으로 레이어를 구성함으로써 사용자가 특정 문자를 다른 문자로 옮기는 행위를 도울 수 있다. 그러나 외부에서 보기에는 이중으로 구성된 행렬이 상대적으로 움직이는 것으로 보여 어떠한 문자를 어떠한 문자로 옮기는 지 확신할 수 없게 된다. S3PAS〔9〕에서 문자 셋을 이용하여 삼각형을 그리고 그 안의 다른 문자를 선택하는 것은 이와 같은 문자 간 연산의 한 방법이라



〔그림 2〕 문자 간 상대적 위치를 이용한 행렬에서의 'E'**'U' 연산. 마우스 포인터를 감출 경우 이중 레이어를 구성하면, 어떠한 문자를 어떠한 문자로 옮기는 지 특정하기 힘들어진다.

고 볼 수 있다.

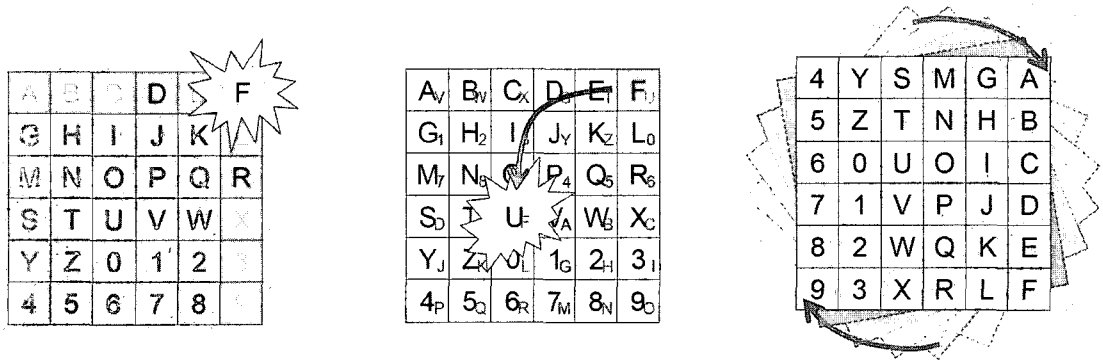
이와 같이 행렬에서 문자의 위치를 바탕으로 간단한 규칙만을 배우면 모든 가능한 연산 결과에 대해서 암기하는 것보다 쉽게 문자 단위 연산을 수행할 수 있다. 이 논문에서는 이러한 문자 간 연산을 이용하여 외부 공격에 강한 인증 프로토콜을 설계한다.

3.2 기본 기법

제안 기법에서 사용자는 자신의 패스워드를 두 자리씩 처리해야할 필요가 있다. 예를 들자면, 만약 사용자의 패스워드가 'DRAGON'이라면, 사용자는 이로부터 순서대로 'DR', 'RA', 'AG', 'GO', 'ON', 'ND'를 연상해 내야한다. 그리고 추가적으로 사용자는 이번 인증 세션에서 사용할 추가 비밀을 하나에서 최대 패스워드 길이의 절반만큼의 임시 비밀을 선택한다. 여기에서는 'U' 하나만을 선택했다고 가정한다.

검증자는 사용자를 위하여 패스워드가 사용하는 문자 공간에 따라 6×6, 7×7, 10×10 등의 행렬을 생성하여 이를 전송한다. 아래 〔그림 3〕의 예에서는 6×6 행렬을 사용하였다. 사용자는 우선 부분 패스워드 'DR'에 대하여 'D'가 위치한 행과 'R'이 위치한 열의 교차점을 눈으로 찾는다. 교차점이 'F'라고 하면, 이를 자신이 선택한 임시 비밀 'U'로 옮긴다. 입력이 끝나면 행렬은 오른쪽으로 90° 회전하여 다음 부분 패스워드 'RA'에 대해서 동일한 작업을 수행한다. 마지막 부분 패스워드까지 수행하면 이를 검증자에게 전송한다.

검증자는 사용자의 패스워드를 알고 있기 때문에, 위의 입력이 올바른지 아닌지 확인할 수 있다. 하나의 행렬에 대해서 올바른 교차점은 오로지 하나만 존재하



(1단계) 'D'*'R' = 'F' (2단계) 'F'**'U' (3단계) 오른쪽 90° 회전

〔그림 3〕 부분 패스워드 'DR'을 입력하는 방법에 대한 예시. (1단계)에서 'DR'의 교차점 'F'를 찾는 연산을 수행한 후, (2단계)에서 교차점 'F'와 임시 비밀 'U'에 대해 행렬 이동 연산을 수행한다. (3단계)에서는 연산을 수행했던 행렬을 오른쪽으로 90° 회전하여 다음 부분 패스워드 'RA'에 대한 입력을 계속한다.

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	0	1	2	3
4	5	6	7	8	9

(a) 일반 행렬

4	Y	S	M	G	A
5	Z	T	N	H	B
6	0	U	O	I	C
7	1	V	P	J	D
8	2	W	Q	K	E
9	3	X	R	L	F

(b) 오른쪽 90° 회전 행렬

2	1	Y	3	Z	0
K	J	G	L	H	I
8	7	4	9	5	6
E	D	A	F	B	C
W	V	S	X	T	U
Q	P	M	R	N	O

(c) 행/열 단위 변환 행렬

〔그림 4〕 행렬의 변형에 따라 'P'가 속한 집합의 변화. 최초의 행렬(a)에서 'P'는 {M,N,O,P,Q,R}과 {D,J,P,V,1,7}에 속해 있으며, 다른 변형(b,c)에 대해서도 동일한 집합에 속해 있음을 알 수 있다.

게 되고, 사용자의 입력과 실제 패스워드를 비교하여 임시 비밀을 추측할 수 있다. 만약 임시 비밀이무작위성을 갖는다면, 검증자는 사용자가 패스워드를 모르고 있다고 판단한다.

다음 번 인증에서 검증자는 이전에 사용되었던 행렬을 변형하여 사용자에게 전송한다. 변형은 무작위로 행렬을 선택하는 것이 아니며, 행 단위나 열 단위로 섞고, 여기에 몇몇 문자의 위치를 바꿈으로써 만든다.

IV. 설계 고려 사항 및 안전성 분석

4.1 교차점을 찾는 문자 간 연산(교차점 연산)

교차점을 찾는 연산(교차점 연산)은 부분 패스워드를 입력하는 사이사이에서의 행렬의 변형(즉, 오른쪽 90° 회전)과 인증과 인증 사이에서의 행렬의 변형(즉, 행 및 열 단위의 섞음)에 있어서 큰 장점을 가지고 있다. 오른쪽으로 90° 회전한 행렬은 이전 행렬에서 열과 행이 뒤 바뀌어 있음을 알 수 있다. 부분 패스워드는 두 자리로 이루어져 있고, 이전 부분 패스워드의 뒷자리가 이번 부분 패스워드의 앞자리로 바뀌게 된다. 이러한 경우 오른쪽으로 90° 회전함으로써 패스워드가 속한 집합을 그대로 유지해줄 수 있는 것이다. 만약 그렇지 않고 동일한 행렬을 유지하거나 완전 무작위로 행렬을 생성하여 사용할 경우, 패스워드가 속한 집합이 달라지어 부분 패스워드 사이의 교집합 공격이 이루어져 패스워드가 노출될 수 있다. 또한 인증과 인증 사이에서 행렬을 열과 행 단위로 섞는 것 또한 패스워드가 속한 집합을 그대로 유지하는 특징을 가지고 있어 교집합 공격에 강한 구조를 만들어 낸다. 또한 문자 간 상대적 위치는 이러한 변형에 의해서 변화하기 때문에, 재전송 공격에도 강할 수 있다.

4.2 임시 비밀과 행렬 이동 연산

만약, 교차점 연산만을 사용하여 패스워드를 입력하도록 했을 때, 사용자는 이와 같은 행렬의 변형에도 불구하고 항상 동일한 교차점을 선택하여 전송할 것이다. 공격자는 이를 관찰할 후, 사용자가 선택했던 문자(교차점)를 다시 선택함으로써 쉽게 인증을 통과할 수 있다. 제안 기법에서 교차점을 사용자가 정한 임시 비밀과 다시 연산하도록 하는 것은 이러한 공격을 어렵게 한다.

임시 비밀은 사용자로부터 제공되는 무작위성으로, 인증 세션 간 독립적이다. 임시 비밀은 교차점과 행렬 이동 연산이 이루어지므로, 인증 세션 간 교집합 공격으로는 교차점이 무엇인지 특정할 수 없다. 그러나 임시 비밀은 사용자가 직접 검증자에게 알려줄 수 없으므로, 두 개 이상의 부분 패스워드에 대해서 동일한 임시 비밀을 사용해야만 한다. 따라서 임시 비밀의 최대 수는 패스워드 길이의 절반이 된다.

임시 비밀의 수가 늘어남에 따라, 성공적인 인증 세션을 관찰한 공격자가 올바른 교차점을 찾아내는 것은 더욱 어려워진다. 만약, 임시 비밀이 하나라면 행렬의 크기만큼의 가능한 경우가 존재하지만, 임시 비밀이 둘로 늘어날 경우, 경우의 수가 패스워드 길이에 따라서 더 늘어난다. 사용자가 선택할 수 있는 임시 비밀의 나열 순서에 따른 입력 방법은 $n=6$일 때, 120가지가 되지만 이것이 인증 세션마다 다르게 선택될 때에는 여러 인증 세션을 바라볼 수 있는 공격자에게 보다 많은 교차점에 관한 정보를 주게 된다. 따라서 사용자마다 이러한 입력 방법을 다르게 하고, 동일한 사용자는 동일한 입력 방법을 사용하는 것이 좋지만, 이는 사용자에게 또 다른 비밀이 되므로 적합하지 않다. 따라서 제안 기법에서는 2자리나 3, 4자리씩 끊어서

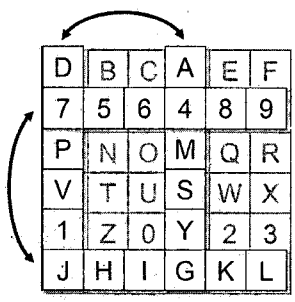
동일한 임시 비밀을 사용하도록 한다.

기본적으로 검증자는 실패한 인증 시도에 있어서 동일한 행렬을 재전송하게 된다. 그러나 일정한 횟수 이상의 인증 실패가 있을 경우, 완전하게 무작위로 섞은 행렬을 공격자로 추측되는 사용자에게 전송하고, 만약, 이러한 행렬에 대한 인증이 성공할 경우 패스워드를 바꿀 것으로 사용자에게 권고해야 한다.

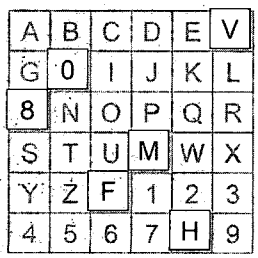
4.3 인증 세션간 행렬의 변형

인증 세션간 행렬의 변형은 검증자에서 제공하는 무작위성이다. 이 무작위성은 공격자에게 사용자의 비밀을 노출시키게 되는 근거가 된다. 하지만 무작위성이 없다면 공격자는 재전송 공격을 수월하게 수행할 수 있기 때문에, 사용자의 비밀을 조금씩 노출시키고 재전송 공격에도 안전할 수 있도록 하는 것이 중요하다. 따라서 행렬의 변형이 지나치게 급격하게 이루어지게 되면 공격자는 사용자의 비밀을 알아낼 수 있고, 행렬의 변형이 너무 안 일어나게 되면 재전송 공격이 성공할 확률이 높아지게 된다. 제안 기법에서 인증 세션 간 행렬의 변형은 이러한 부분을 고려하여 이루어진다.

제안 기법에서 사용하는 행렬의 변형은 [그림 5]와 같이 두 가지로 나눌 수 있다. 하나는 행렬 단위 위치 교환이고, 다른 하나는 문자 단위 위치 교환이다. 행렬 단위 위치 교환은 행과 열 단위로 서로 섞는 것을 의미한다. 단순히 행과 열을 섞는 것만으로도 문자들 사이의 상대적 위치가 변하기 때문에 재전송 공격이 어려워지지만 패스워드 문자가 속한 집합을 그대로 유지하기 때문에 사용자의 비밀이 노출되는 것을 막을 수 있다. 한편, 인증 세션을 여러 번 관찰하는 공격자는 사용자의 비밀이 만들어내는 교차점을 추측할 수 있기 때문에 이러한 교차점이 일정한 인증 세션마다 조금씩 바뀌도록 해야 한다. 문자끼리 서로 위치를 바꾸게 되면 한 문자가 속한 행렬 집합이 바뀌게 되는 현상이 일어나게 되는데, 이 경우 교차점이 바뀌게 된다. 문자 단위 위치 교환은 매번 일어나는 것은 아니며 일정한 인증 횟수에 한 번 씩 일어나게 된다.



(a) 행렬 단위 위치 교환



(b) 문자 단위 위치 교환

[그림 5] 두 가지 행렬 변형 방법. 행렬을 과도하게 바꿀 경우, 공격자는 이를 이용하여 패스워드를 알아낼 수 있다.

공격자는 여러 성공적인 인증 세션을 이용하여 교차점을 추측하려고 할지도 모른다. 공격자가 만약 전혀 다른 행렬에서의 두 성공적인 인증 세션을 관찰하였다면, 공격자는 성공적으로 전체 패스워드에 대한 교차점을 얻어낼 수 있다. 행렬의 문자 배열이 전혀 다른 두 행렬을 바라보았을 때 교차점이 노출된다고 볼 수 있다. 즉, 일정 횟수 이후에는 반드시 교차점을 다른 것으로 바꿔주어야만 한다. 제안 기법에서는 문자 단위 위치 교환을 통하여 이러한 일을 수행한다.

문자 단위 위치 교환은 일정한 인증 횟수에 한 번씩 발생하여, 교차점의 값을 다른 값으로 바꿔주는 역할을 수행한다. 문자 단위 위치 교환은 문자가 속한 집합을 바꿔줌으로써, 부분 패스워드에 해당하는 교차점을 다른 것으로 바꿔준다. 한편, 이러한 문자 단위 위치 교환은 공격자에게 사용자의 비밀에 대한 정보를 직접 노출시킨다. 이는 연속해서 공격자에게 인증 과정이 노출되는 경우를 이야기하는 것이기도 하지만, 공격자가 관찰한 최초의 인증과 현재 공격자가 관찰한 인증에서 사용되었던 행렬의 차이가 극명하여 패스워드가 노출되는 경우를 의미하기도 한다. 즉, 공격자에게 패스워드가 단 2회 노출되었다 하더라도 그 사이에 수많은 인증 과정이 있었다면, 공격자는 패스워드를 얻어낼 수 있음을 의미한다.

4.4 파라미터 선택에 따른 보안성 고려

앞서 살펴보았던 것과 같이 행렬의 크기, 패스워드의 길이 등의 파라미터에 따라서 실제로 보안 수준이 결정된다. 정방 행렬의 한 변의 길이를 m, 패스워드의

길이를 n으로 하고, 문자 단위 위치 교환에서 교환하는 문자의 수 k로 하고, q자리씩 동일한 임시 비밀을 사용한다고 가정하자. (따라서, n은 q의 배수이어야 한다.)

행렬의 한 변의 길이 m에 따라서 화면에 출력할 수 있는 문자의 수는 제한적이다. $m=6$일 경우 A~Z와 같이 대소문자를 구별하지 않는 26자의 영문자와 0~9와 같이 10자의 숫자로 이루어질 수 있다. $m=7$일 경우 $m=6$인 행렬에 추가적으로 자주 사용하는 기호(!,@,#,$,%,^,&,*,(,),-,+,=) 13자를 더해 만들어질 수 있다. 행렬의 크기가 작으면 작을수록, 사용자가 자신의 패스워드에 해당하는 문자를 찾는 시간이 단축되어 사용자 편의성이 높아지나, 반대로 행렬의 크기가 커지만 커질수록 보안성은 더 높아지게 된다.

패스워드 길이 또한 길면 길수록 보안성이 높아진다. 그러나 그와 동시에 인증에 소요되는 시간이 증가하고 실수로 인하여 인증에 실패할 확률 또한 높아진다. 사용자가 인증을 더 빨리 하려고 하면 할수록 실패할 확률이 높아지는 것은 당연하다.

임시 비밀의 수의 경우, 많으면 많을수록 사용자가 임시 비밀을 기억하고 있어야 하는 시간이 짧아지므로 사용자 편의성이 증대될 것 같지만, 여러 개의 임시 비밀을 선택하는 것에 대해서 불편함을 느낄 수 있어 단정 지어 말하기 힘들다. 임시 비밀의 경우, 많으면 많을수록 교차점 추측과 재전송 공격을 어렵게 하나, 무작위 인증 시도의 확률이 높아지는 특징이 있다.

행렬의 변형에 있어서, 문자 단위 위치 교환의 주기를 조절하는 것 또한 중요하다. 문자 단위 위치 교환의 주기가 빠르면 교차점 추측 인증 시도는 어려워지지만 최종적으로 패스워드가 공격자에게 노출되는 인증 횟수가 단축되는 효과가 있다. 반대로 주기가 느리면 중간에 교차점 추측 인증 시도의 확률이 높아지게 된다.

4.4.1 무작위 인증 시도

아무런 정보도 없이 무작위로 입력을 하여 공격자가 인증을 무사히 통과할 확률은 임시 비밀의 숫자만큼을 제외한 부분 패스워드가 일치할 확률이므로

$$\Pr[A^{brute}]=\left(\frac{1}{m^2}\right)^{n-n/q} \quad (1)$$

와 같다. [표 1]은 $q=2$로 할고, m과 n을 다양화 하였을 때의 $\Pr[A^{brute}]$의 변화를 보여준다.

[표 1] 행렬의 한 변의 길이와 패스워드의 길이에 따른 무작위 인증 성공 확률(m: 행렬의 한 변의 길이, n: 패스워드의 길이, 임시 비밀: 두 자리)

		n	
		6	8
m	6	2.14×10^{-5}	5.95×10^{-7}
	7	8.50×10^{-6}	1.73×10^{-7}
	8	3.81×10^{-6}	5.06×10^{-8}
	9	1.88×10^{-6}	2.32×10^{-8}
	10	1.00×10^{-6}	1.00×10^{-8}

4.4.2 교차점 추측에 따른 인증 시도

이벤트 E_1을 행렬 단위 위치 변화가 일어난 경우, 이벤트 E_2를 문자 단위 위치 변화가 일어난 경우라고 하자. 한 번의 인증 세션을 바라본 뒤 교차점을 찾아낼 확률은 임시 비밀을 정확히 추측할 확률과 같으므로

$$\Pr[A^{cross}]=\Pr[A^{cross}\cap E_1]=\left(\frac{1}{m^2}\right)^{n/q} \quad (2)$$

와 같은 데, $q=2$일 때 [표 1]에서 무작위 인증 성공 확률과 동일하다. 행렬 단위 위치 교환만 발생할 경우, 이 확률로 공격자는 인증을 통과할 수 있다. 문자 단위 위치 교환이 발생할 경우에는, 더 확률이 낮아지게 되며

$$\Pr[A^{cross}\cap E_1\cap E_2]=\left(\frac{1}{m^2}\right)^{n/q}\left(\frac{m^2-k}{m^2}\right)^{n} \quad (3)$$

와 같다.

4.4.3 재전송 공격을 통한 인증 시도

행렬 단위로 위치를 교환 하였을 경우, 이전 인증 세션을 재전송하여 통과하기 위해서는 바뀐 행과 열에 패스워드나 임시 비밀이 존재해서는 안 된다. 그러나 행렬을 행 단위로 바꾼다고 해도 홀 수 번째 패스워드 문자는 영향을 받지 않고, 열 단위로 바꾼다고 해도 짝 수 번째 패스워드 문자는 영향을 받지 않는다. 임시 비밀은 열과 행에 모두 영향을 받게 된다. 최소 수준의 행렬 단위 위치 교환만 발생한 행렬에 앞선 사용자의 입력을 이용하여 재전송 공격을 시도하여 성공할

확률은

$$\Pr[A^{replay} \cap E_1] = \left(\frac{m^2-2m}{m^2}\right)^n \left(\frac{m^2-4m+4}{m^2}\right)^{n/q} \quad (4)$$

와 같으며, 여기에 추가적으로 문자 단위 위치 교환이 발생하였을 때, 이렇게 바뀐 문자들에 패스워드나 임시 비밀이 속하지 않을 확률을 고려하면 재전송 공격이 성공할 확률은

$$\Pr[A^{replay} \cap E_1 \cap E_2] = \Pr[A^{replay} \cap E_1] \times \left(\frac{m^2-k}{m^2}\right)^{n+n/q} \quad (5)$$

와 같이 된다. 한편, 검증자가 다음 인증에 사용할 행렬을 인증 후에 바로 생성하면 재전송이 통과할 행렬을 사전에 검출할 수 있으므로, 사실 상 재전송이 성공할 확률은 존재하지 않는다. 행과 열을 바꿀 수 있는 모든 가능한 경우의 수는 $(m!)^2$이므로, 동일한 행렬이 다시 나타날 때까지 기다려 재전송 공격을 하려면 공격자는 꽤 오랜 시간을 기다려야 한다.

4.4.4 지속적인 인증 관찰

공격자는 지속적으로 인증 과정을 관찰함으로써, 사용자의 비밀을 알아내려고 할 수 있다. 이 때, 사용자의 패스워드가 공격자에게 정확하게 알려지는 인증 횟수는 패스워드의 길이보다는 행렬의 크기와 깊이 관련이 있다. 패스워드는 전혀 다른 행렬에 대한 두 번의 인증이 성공한 경우에 노출된다고 가정한다. 행렬 단위 위치 교환은 문자가 속한 집합에 대한 정보를 노출시키지만, 패스워드가 무엇인지 결정짓지 못한다. 따라서 전혀 다른 행렬이 만들어지기 위해서는 문자 단위 위치 교환이 일정 수 이상 이 일어나 패스워드가 7모두 다른 집합으로 옮겨져야만 한다. 따라서 한 번에 k씩 문자가 옮겨질 경우에 모든 문제가 옮겨질 때까지 걸리는 최소 횟수는 m^2/k번이고, 평균 횟수는 쿠폰 수집 문제에 의해서 $m^2 H_{m^2}/k \approx (m^2 \ln m^2 + 0.5m^2)/k$번이다. [표 2]는 $k=m$이고 2회에 1번씩 문자 단위 교환이 일어나는 경우를 가정하였을 때 패스워드가 완전히 노출될 때까지의 최소와 최대 인증 횟수를 요약하여 보여준다.

[표 2] 사용자의 패스워드가 노출될 때까지의 최소~평균 인증 횟수 (m: 행렬의 한 변의 길이)

	m				
	6	7	8	9	10
인증 횟수	12~49	14~61	16~74	18~88	20~102

이러한 결과를 근거로 검증자는 현재까지 몇 번의 인증을 수행했는지 확인하고 이를 사용자에게 알려, 패스워드를 바꾸도록 유도해야한다.

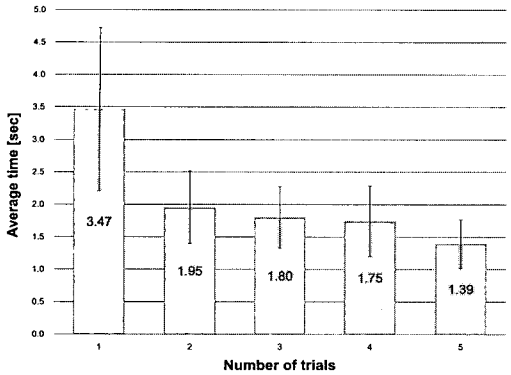
V. 사용자 실험 및 그 결과

제안 기법의 사용성 평가를 위해서 17명의 20~22세 대학생들을 대상으로 사용자 실험을 하였다. 피 실험자들에게는 6자리 이상의 패스워드를 자기가 원하는 대로 사용하도록 하였으며, 임시 비밀은 한 인증 세션에서는 하나만을 사용하도록 하였고, 6×6 행렬에 대해서 인증을 시도하도록 하였다. 인증 실험에 앞서 자신의 패스워드를 두 자리씩 차례로 입력하는 실험을 수행한 뒤, 행렬 단위 위치 교환이 일어난 행렬에서의 인증 시도 5회, 그 뒤에 문자 단위 위치 교환이 일어난 행렬에서의 인증 시도를 5회 실시하도록 하였다. 인증 시도에 있어서는 피 실험자들이 키보드나 마우스를 자유롭게 선택하여 사용할 수 있도록 하였는데, 마우스를 이용한 경우는 27.3%, 키보드를 이용한 경우는 72.7%였다. 마우스를 사용하다 키보드로 전환한 피 실험자는 4명이었으나, 키보드에서 다시 마우스로 전환한 경우나 피 실험자는 없었다.

5.1 부분 패스워드 입력 실험 결과

[그림 6]은 부분 패스워드 두 자리를 입력을 완료할 때까지(키보드로 입력하는 시간 포함) 걸린 시간에 대한 평균과 표준편차를 보여준다. 최초의 시도 이후에 피 실험자들의 입력 시간이 급격히 줄어듦을 알 수 있는데, 이는 한 번 패스워드로부터 부분 패스워드를 만든 경험이 이후의 입력에 있어서 크게 도움이 된다는 것을 의미한다. 시도 횟수가 늘어날수록 평균뿐만 아니라 표준 편차 또한 줄어드는 것은, 학습에 따라 피 실험자들이 더 안정적으로 부분 패스워드를 만들 수 있음을 알 수 있었다.

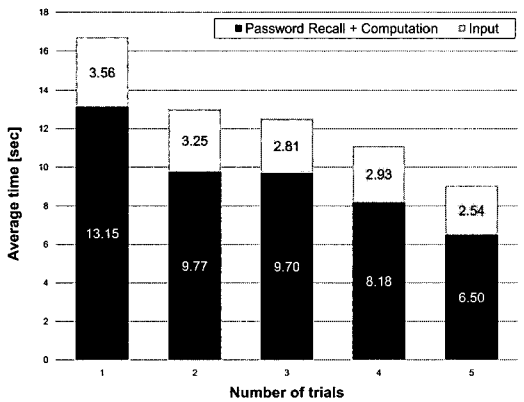
한편, 피 실험자가 부분 패스워드를 잘못 입력하는 경우는 1차 시도에서 10회가 있었고, 2차 시도에서는 9회, 3차 시도에서는 6회, 4차와 5차 시도에서는 각 1회 씩 있었다.



〔그림 6〕 입력 시도와 부분 패스워드를 입력하는데 걸리는 평균 시간의 변화. 피 실험자들은 부분 패스워드의 입력에 익숙해지는 경향을 보였다.

5.2 인증 시도 - 행렬 단위 위치 교환

실험에서 행렬 단위 위치 교환은 보안성과 상관없이 무작위로 행과 열을 섞어, 피 실험자들이 무작위로 섞인 것과 같은 느낌을 받도록 하였다. 〔그림 7〕은 실제 인증에서 하나의 부분 패스워드를 입력할 때 소요되는 평균 시간을 인증 시도 횟수에 따라 표시한 것이다. 행렬의 큰 변화에도 불구하고, 사용자들의 실행 속도는 시도 횟수가 많아질수록 더 빨라졌으며, 입력하는데 걸리는 시간 또한 빨라진 것을 확인할 수 있다. 〔그림 7〕은 〔그림 6〕과 비슷한 양상을 보여주는데, 인증 시도가 늘어날수록 피 실험자들이 인증 방법에 익숙해졌기 때문으로 보인다. 여기에서 특히나 다



〔그림 7〕 행렬 단위 위치 교환 행렬에 대한 인증 시도 평균 시간. 회색 영역은 키보드나 마우스를 움직인 뒤 입력을 끝냈을 때까지의 시간을, 검은 영역은 부분 패스워드와 임시 비밀을 기억하고 연산을 끝냈을 때까지의 시간을 의미한다.

〔표 3〕 행렬 단위 위치 교환 행렬에서의 인증 시도 성공 확률 및 부분 패스워드 입력 시도 성공 확률

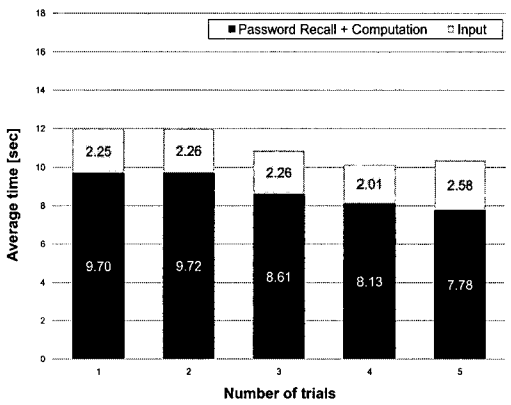
	성공	시도	성공 확률
전체 인증 시도 (1회 실패 인정)	58 (73)	85 (85)	68.24% (85.88%)
부분 패스워드 입력 시도	511	565	90.44%

섯 번째 인증 시도에서 인증 시간이 더 낮아지는 모습을 보이는 데, 이는 피 실험자들이 행렬의 큰 변화에도 불구하고, 교차점이 동일하다는 사실을 깨달았기 때문으로 풀이된다.

이때의 인증의 성공 확률은 〔표 3〕에서 확인할 수 있다. 완벽하게 전부 맞았을 경우에만 인증을 무사히 통과한 것으로 가정했을 경우, 피 실험자들은 68.24%로 인증을 성공할 수 있었다. 한편, 부분 패스워드를 입력하기 위해서 연산을 수행한 경우에 대한 성공 확률을 이것에 비해 높은 90.44%가 된다. 만약, 사용자에게 한 번의 입력 실패를 용인해줄 경우, 인증 성공 확률은 87.05%에 이르게 된다. 이러한 오류의 일부 인정은 HB 프로토콜이 그러하듯 공격자에게 패스워드 추출을 더욱 어렵게 만드는 효과가 있을 것으로 보인다.

5.3 인증 시도 - 문자 단위 위치 교환

문자 단위 위치 교환 행렬에 대한 인증은 행렬 단위 위치 교환 행렬에서의 인증과 조금 다르게 시도 횟수가 늘어난다고 해도 큰 폭으로 시간이 단축되지는 않았다. 어느 정도 학습이 이루어진 상황에서 연산을 수



〔그림 8〕 문자 단위 위치 교환 행렬에 대한 인증 시도에 걸린 평균 시간

〔표 4〕 문자 단위 위치 교환 행렬에서의 인증 시도 성공 확률 및 부분 패스워드 입력 시도 성공 확률

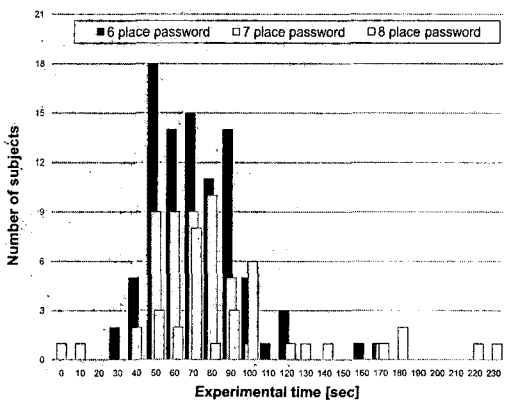
	성공	시도	성공 확률
전체 인증 시도 (1회 실패 인정)	37 (42)	85 (85)	43.53% (49.41%)
부분 패스워드 입력 시도	440	565	77.88%

행하는데 걸리는 시간을 단축시키는 데 한계에 다다른 것으로 보인다. 〔그림 8〕은 이러한 실험 결과를 보여준다. 입력하는데 걸리는 시간은 2초대 초반으로 피 실험자들이 어느 정도 입력 하는 것에는 익숙해진 모습을 보여준다.

이때의 인증 성공 확률은 〔표 4〕에서 확인할 수 있는데, 행렬 단위 위치 교환 행렬에 대한 인증 시도 보다 성공 확률이 크게 낮아졌음을 알 수 있다. 교차점이 달라질 수 있음에도 불구하고, 여덟 명의 피 실험자들이 동일한 교차점을 패스워드 대신 외워 사용하다. 뒤 늦게 이러한 사실을 알아차린 것이 실험 결과에 남아 있다. 이러한 실험 결과로 미루어 보건데, 사용자들에게 교차점이 바뀔 수 있음을 알려주어야 할 것으로 보인다.

5.4 패스워드 길이에 따른 인증 시간 및 성공 확률

17명의 피 실험자 중, 6자리 패스워드를 사용한 사람은 9명, 7자리 패스워드를 사용한 사람은 5명, 8자리 패스워드를 사용한 사람은 3명이었다. 패스워드를 9자리 이상 사용한 피 실험자는 없었다. 패스워드가



〔그림 9〕 패스워드 길이별 인증 완료 시간 구간별 측정. 패스워드가 길어질수록 인증 시간이 지체되는데, 8자리 패스워드에서 크게 늦어진다.

〔표 5〕 패스워드 길이에 따른 인증 성공 확률의 변화 (성공/시도)

	6자리 패스워드	7자리 패스워드	8자리 패스워드
전체 인증 (1회 실패 인정 시)	58/90 (72/90)	21/50 (26/50)	16/30 (17/30)
	64.44% (80.00%)	42.00% (52.00%)	53.33% (56.67%)
부분 패스워드	473/540	276/350	202/240
	87.59%	78.86%	84.17%

길어질수록 인증에 걸리는 시간은 증가할 것이라는 것은 쉽게 추측할 수 있는데, 6자리의 패스워드를 입력하는데 걸린 평균 시간은 76.69초가 소비되었고, 7자리 패스워드를 입력하는 데 걸린 평균 시간은 77.20초가 소비되었다. 이렇게 미미한 시간의 증가와 다르게 8자리 패스워드를 입력하는 게 걸린 평균 시간은 104.85초로 8자리부터는 빠르게 처리하지 못하는 모습을 보여주었다. 〔그림 9〕는 이러한 인증 시간의 변화를 패스워드의 길이에 따라 구간별 분포를 보여준다.

〔표 5〕는 패스워드 길이에 따른 인증 성공 확률의 변화를 보여준다. 7자리 패스워드를 사용한 피 실험자군에서 성공률이 낮아지는 모습을 보이는데, 이는 정상적으로 실험을 수행하지 못한 피 실험자 중 한 명이 해당 군에 속해 있기 때문이다. 패스워드의 길이가 길어지면 기본적으로 성공 확률이 낮아지는 모습을 확인할 수 있다.

VI. 기법 간 비교 분석

훔쳐보기와 같은 외부 공격에 안전하도록 설계된 많은 기법들 간에는 사용자가 외워야 하는 비밀의 양이나 종류가 다르고, 수행하는 과정이나 환경 파라미터가 서로 다르기 때문에 직접적인 비교는 적합하지 않다. 예를 들어, Weinshall의 기법〔6〕은 무작위 시도나 훔쳐보기 공격에 매우 강한 것처럼 보이지만, 인증에 필요한 라운드 수가 다른 기법보다 월등히 많을 뿐만 아니라, 일정 수 이상의 이미지를 비밀로써 온전히 기억하기 위해서는 다른 기법과 비교도 되지 않는 시간을 투자해야 되는 것으로 알려져 있다. 〔표 6〕은 외형적으로 보이는 몇몇 특징을 근거로 제안하는 기법이 가진 특징을 살펴본 것이다.

〔표 6〕에서 살펴보다시피, 다른 기법에 비해서 비밀의 종류나 길이에 있어서 큰 장점을 가지고 있을 뿐

〔표 6〕 다양한 인증 기법 간의 외형 비교
('-' 표시는 자료가 없거나 비교 대상이 아님, *: 비밀과 관련되지 않고, 입력을 생성하여 통과할 수 있는 확률)

		Matsumoto의 기법〔4〕	Weinshall의 기법〔6〕	S3PAS〔9〕	PAS〔10〕	Zheng의 기법〔11〕	제안 기법
비밀	종류	숫자, 16진수	이미지	A-Z, a-z, 0-9, 특수문자	A-Z, 0-9	stroke shape	A-Z, 0-9, (a-z, 특수문자)
	길이/수	≤ 18+5개	≤ 30개	≥ 4자리	≥ 14자리	1개	≥ 6자리
	술어 (predicate)	-	-	3자리 패스워드	특수한 형태	-	2자리 패스워드
사용성	화면 표시 정보	36~50	≥ 80	10×10 행렬	≥ 325+캡차	5×5 행렬	≥ 6×6행렬
	라운드 수	1회	≥ 11회	≥ 4회	2~5회	1회	≥ 6회
	인증 성공률	-	≥ 95%	-	91~94%	-	68~87%
	평균 인증 시간	-	≥ 180초	-	55~84초	-	50~100초
보안성	무작위 시도	10^{-9}	≤ 10^{-23}	≤ 10^{-5}	10^{-3*}	10^{-4*}	10^{-8}~10^{-5}
	훔쳐보기 공격	10^{-6}	≤ 10^{-15}	-	-	10^{-14}	10^{-8}~10^{-5}
	패스워드 노출 까지	-	≥ 6회	≥ 2회	≥ 10회	-	≥ 12~49회

만 아니라, 비밀로부터 술어를 생성하는 방법도 S3PAS에 비해서 수월해 보인다. 또한 화면에 표시되는 정보의 수도 상대적으로 적은 편에 속하며, 인증 시간도 사용하는 비밀의 종류에 비추어보자면 어느 정도 타당해 보인다. 보안성 또한 다양한 공격을 고려하여 균형 잡힌 확률과 안전성을 갖도록 제안 기법이 설계되었음을 알 수 있다. 단지, 예상 보다 낮은 인증 성공률을 사용자 편의성에 대한 고려가 더 필요함을 보여준다.

VII. 결 론

이 논문에서는 행렬 위에 문자를 배열하고, 문자 간 연산이 가능하게 하여, 사용자로 하여금 자신의 패스워드가 외부로 노출되지 않도록 하는 인증 기법에 대해서 제안하였다. 행렬을 사용하도록 하여, 사용자가 현재 자신이 사용하고 있는 패스워드를 그대로 사용할 수 있도록 배려하였으며, 훔쳐보기 공격에 따른 다양한 패스워드 노출 가능성을 고려하여 최대한 패스워드가 안전할 수 있도록 하였다. 또한 사용자 실험을 통하여, 제안하는 기법이 실제 사용자들이 사용하기에 큰 무리는 없음을 확인하지만, 동시에 원활한 사용자의 인증을 위하여 사용자가 범할 수 있는 오류의 허용 등에 대해서 사용자에게 더 배려해야 함을 확인할 수 있었다. 이를 바탕으로 앞으로 사용자 편의성이 향상됨과 동시에 보안성을 높일 수 있는 인증 방법에 대한 연구를 계속해야할 것으로 보인다.

참 고 문 헌

〔1〕 R. Dhamija and A. Perrig, "Déjà Vu: A User Study Using Images for Authentication," Proc. of 9th USENIX Security Symposium, p. 4, Aug. 2000.

〔2〕 S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskily, and N. Memon, "PassPoints: Design and longitudinal evaluation of a graphical passwords system," International Journal of Human-Computer Studies (Special Issue on HCI Research in Privacy and Security), vol. 63, pp. 102-127, May 2005.

〔3〕 S. Wiedenbeck, J. Waters, L. Sobrado, and J.C. Birget, "Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme," Proc. of Advanced Visual Interfaces (AVI), pp. 177-184, May 2006.

〔4〕 T. Matsumoto and H. Imai, "Human Identification Through Insecure Channel,"

Proc. of EUROCRYT 91, LNCS 547, pp. 402-421, 1991.

[5] N. Hopper and M. Blum, "Secure Human Identification Protocols," Proc. of ASIACRYPT, LNCS 2248, pp. 52-66, 2001.

[6] D. Weinshall, "Cognitive Authentication Schemes Safe Against Spyware (Short Paper)," Proc. of the 2006 IEEE Symposium on Security and Privacy (S&P), pp. 1-16, May 2006.

[7] P. Golle and D. Wagner, "Cryptanalysis of a Cognitive Authentication Scheme (Extended Abstract)," Proc. of the 2007 IEEE Symposium on Security and Privacy (S&P), pp. 66-70, May 2007.

[8] H. Jameel, R.A. Shaikh, H. Lee, and S. Lee, "Human Identification Through Image Evaluation Using Secret Predicates," Proc. of The Cryptographer's Track at RSA Conference (CT-RSA), LNCS 4377, pp. 67-84, 2007.

[9] H. Zhao and X. Li, "S3PAS: A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme," Proc. of 21st IEEE International Conference on Advanced Information Networking and Applications Workshop (AINAW), pp. 467-472, May 2007.

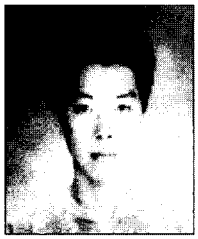
[10] X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan, and B. Ma, "PAS: Predicate-based Authentication Services Against Powerful Passive Adversaries," Proc. of 2008 Annual Computer Security Applications Conference (ACSAC), pp. 433-442, Dec. 2008.

[11] Z. Zheng, X. Liu, L. Yin, and Z. Liu, "A Stroke-based Textual Password Authentication Scheme," Proc. of 2009 First International Workshop on Education Technology and Computer Science, pp. 90-95, Mar. 2009.

[12] I. Jermynn, A. Mayer, F. Monrose, M.K. Reiter, and A.D. Rubin, "The Design and Analysis of Graphical Passwords," Proc. of the 8th USENIX Security Symposium, p. 1, Aug. 1999.

〈著者紹介〉

강 전 일 (Jeonil Kang) 학생회원
2003년 2월: 인하대학교 컴퓨터 공학과 졸업
2006년 2월: 인하대학교 정보통신대학원 석사
2006년 3월~현재: 인하대학교 정보공학과 박사 과정
〈관심분야〉 RFID 보안, 생체 인식 보안, WSN 보안, 무선 인터넷 보안, 웹 인증 보안



맹 영 재 (YoungJae Maeng) 학생회원
2006년 8월: 인하대학교 컴퓨터 공학과 졸업
2008년 8월: 인하대학교 정보통신대학원 석사
2008년 9월 ~ 현재: 인하대학교 정보공학과 박사 과정
〈관심분야〉 인터넷 보안, 네트워크 보안

양 대 헌 (DaeHun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월 ~ 2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월 ~ 현재: 인하대학교 정보통신대학원 조교수
〈관심분야〉 암호 이론, 암호 프로토콜, 인증 프로토콜, 무선 인터넷 보안

이 경 희 (KyungHee Lee) 정회원
1993년 2월: 연세대학교 컴퓨터과학과 학사
1998년 8월: 연세대학교 컴퓨터과학과 석사
2004년 2월: 연세대학교 컴퓨터과학과 박사
1993년 1월 ~ 1996년 5월: LG소프트(주) 연구원
2000년 12월 ~ 2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월 ~ 현재: 수원대학교 조교수
〈관심분야〉 바이오인식, 정보보호, 컴퓨터비전, 인공지능, 패턴인식

전 인 경 (Inkyung Jeun)
1998년 2월: 서울시립대학교 전산통계학과 졸업
2005년 8월: 성균관대학교 정보보호대학원 석사
2005년 9월 ~ 현재 : 성균관대학교 컴퓨터공학과 박사과정
1998년 1월 ~ 2000년 4월: 삼성전자 연구원
2000년 4월 ~ 현재 : 한국인터넷진흥원(KISA) 선임연구원
〈관심분야〉 ID관리, 암호, PKI, 인증 프로토콜, 정보보호