에디트 거리 알고리즘을 이용한 디지털 도어락의 보안성 강화에 관한 연구

박준표*, 조병구*, 최현식*, 정연돈*
*고려대학교 컴퓨터·전파통신공학과
e-mail : {jp_park, byungku, hyunsikchoi, ydchung}@korea.ac.kr

A Study on Security Consolidation by using Edit Distance Algorithm

Jun Pyo Park*, Byungku Cho*, Hyunsik Choi*, Yon Dohn Chung*
*Dept of Computer Science, Korea University

요 약

디지털 도어락은 사용의 편이성와 안전성으로 인해 보편적으로 사용되고 있다. 본 논문에서는 에디트 거리 알고리즘을 활용하여 비밀번호를 사용하는 디지털 도어락의 보안성을 강화하는 방법을 제안한다. 즉, 에디트 거리 알고리즘을 이용하여 비용을 산출함으로써 사용자가 실수할 수 있는 범위들을 정의하여 인증된 사용자인지 인증 받지 않은 사용자인지 효과적으로 예측한다. 실험을 통해 본 논문에서 제안하는 방법을 통해 비밀번호를 사용하는 디지털 도어락의 보안을 강화할 수 있음을 확인할 수 있다.

1. 서론

디지털 도어락(Digital Doorlock)[1]은 사용의 편이성과 안전성으로 인해 보편적으로 사용되고 있다. 분실 염려가 있는 열쇠(key)를 이용하는 것보다 인증 받은 사용자(authenticted user)만 알고 있는 비밀번호, 지문인식, 음성인식 또는 홍채 인식 등을 통한 방법이 안전하고 편리하기 때문이다. 특히, 비밀번호를 이용한 디지털 도어락은 가장 일반화된 방식으로 사용의 편이성과 비용의 이점으로 인해 가장 많이 사용되고 있는 디지털 도어락의 한 종류이다. 그러나 비밀번호를 사용하는 디지털 도어락은 비밀번호가 쉽게 노출될 가능성이 있으며, 추측 또는 무작위 입력을 통한 도어락 해제 가능성 등 보안에 취약한 단점이 있다. 따라서, 비밀번호를 사용하는 디지털 도어락은 인증 받은 사용자와 인증 받지 않은 사용자를 효과적으로 구분하여 보안성을 강화할 수 있는 방법이 필요하다.

한편 비밀번호를 사용하는 디지털 도어락에서 틀린 번호를 입력하는 경우는 다음과 같은 두 가지 경우로 나타낼 수 있다.

경우 1. 인증 받은 사용자가 비밀번호 입력 시 실수를 범할 경우
경우 2. 인증 받지 않은 사용자가 임의로 번호를 입력할 경우

이때, 사용자가 비밀번호를 알고 눌렀는지(경우 1), 또는 임의로 눌렀는지(경우 2) 여부를 좀 더 정확하게 판별해낼 수 있다면 비밀번호를 사용하는 디지털 도어락의 보안을 강화할 수 있다. 본 논문에서는 에디트 거리 알고리즘을 이용하여, 사용자의 입력 값과 미리 정의된 비밀번호를 비교하여 비용을 산출함으로써 사용자를 구별하는 방법을 제안한다.

이 논문의 구성은 다음과 같다. 관련 연구는 2장에 기술되어 있다. 3장에서는 에디트 거리 알고리즘을 사용하여 비용을 계산하는 방법을 설명한다. 실험 및 결과는 4장에 기술되어 있다. 마지막으로 5장에서는 결론을 도출한다.

2. 관련 연구

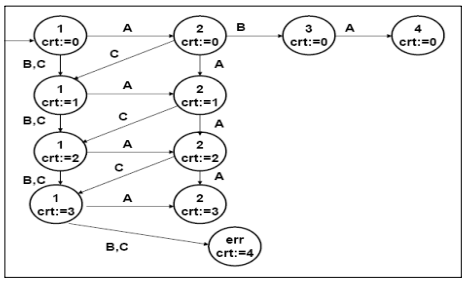
2.1 에디트 거리 알고리즘

```
01:  int EditDistance(char s[1..m], char t[1..n])
02:  // d is a table with m+1 rows and n+1 columns
03:  declare int d[0..m, 0..n]
04:
05:  for i from 0 to m
06:      d[i, 0] := i
07:  for j from 0 to n
08:      d[0, j] := j
09:
10:  for i from 1 to m
11:  {
12:      for j from 1 to n
13:      {
14:          if s[i] = t[j] then cost := 0
15:                         else cost := 1
16:          d[i, j] := minimum(
17:          d[i-1, j] + 1,          // deletion
18:          d[i, j-1] + 1,          // insertion
19:          d[i-1, j-1] + cost)     // substitution
20:      }
21:  }
22:  return d[m, n]
```

(그림 1) 에디트 거리 알고리즘

그림 1은 에디트 거리 알고리즘[2][3]을 보여주고 있다. 에디트 거리 알고리즘은 두 개의 문자열을 비교하여 수정, 삽입, 삭제를 몇 번해야 두 문자열이 같게 되는지 그 비용을 계산 해준다. 그림 1에서 loop를 돌고 있는 부분(10~21번째 줄)에서 두 문자열 대해서 각각의 문자 비교를 통해 삽입, 삭제, 교체 등의 비용을 3가지 경우(case)로 측정하고 그중에서 가작 작은 비용을 선택해 삽입한다. 에디트 거리 알고리즘이 사용되고 있는 곳으로는 철자검색, DNA 분석, 표절 탐색 등의 분야에서 쓰이고 있다. 비용 측정해 대한 자세한 설명은 3장에서 기술한다.

2.2 디지털 도어락 오토마타



(그림 2) 일반적인 도어락 오토마타

그림 2는 일반적으로 사용되고 있는 디지털 도어락의 오토마타이다. 예를 들어, 비밀번호가 ABA라고 가정하고 임의 사용자가 문자를 A, B, A순으로 올바르게 입력을 한다면 1, 2, 3, 4상태로 전이되어 문이 열리게 된다. 그러나 A를 입력해야 하는데 B나 C를 잘못 입력했을 경우 상태가 아래로 전이되며 crt는 1이 된다. 여기서 crt는 틀린 횟수만큼 1씩 증가하는 정수형 변수이다. 그래서 crt > 3이면 err 상태가 된다. 일반적인 디지털 도어락의 err처리는 3회 이상 틀렸을 경우에 경고음 발생 및 키블락 후 일정 시간 경과 후 다시 입력이 가능하도록 설계되어 있다. 이 경우는 사용자가 실제로 비밀번호를 알고 입력 했는지 아니면 임의로 입력했는지 구분할 수 없기 때문에 보안상 위험요소를 내포하고 있다.

3. 제안 방법

이 장에서는 에디트 거리 알고리즘[2][3]을 사용하여 비용 측정하는 방법과 그 비용으로 기반으로 인증 받은 사용자와 인증 받지 않은 사용자를 구분하는 방법을 제안한다. 또한 인증 받은 사용자의 실수 범위를 정의하여 2차적으로 유효 범위를 검사하는 방법을 제안하고, 마지막으로 에디트 거리 알고리즘과 사용자 실수 범위를 적용시킨 디지털 도어락의 상태도를 설명한다.

3.1 비용 계산 방법

비용 측정하는 방법을 간단하게 요약하면, 실제 비밀번호와 사용자가 입력한 번호가 같으면, cost는 0이고, 하나가 틀리면 cost는 1이다.

예제 1. 그림 3은 실제 비밀번호가 (9,4,5,7)이고 사용자가 입력한 번호가 (9,4,8,7,6)일 때, 에디트 거리 산출 방식을 보여주고 있다. 실제 비밀번호가 4자리 수이고 사용자가 입력한 비밀번호가 5자리 수 이므로 4x5의 크기를 가진 2차원 행렬을 생성하여 x축에는 실제 비밀번호(9,4,5,7)를 저장하고, y축에는 사용자가 임으로 누른 번호(9,4,8,7,6)를 저장한다. 이 때, 2차원 행렬을 distance[i][j]라면 x축은 실제 비밀번호의 길이만큼 i를 1씩 증가시키면서 번호를 0번부터 순차적으로 넣는다. 위 예제에서는 비밀번호가 4개이므로 0부터 4까지 넣었다. y축도 동일한 방식으로 넣고 왼쪽부터 열 단위로 실행한다. 표 1에서 cell(1,1) 자리의 비용을 계산하기 위해서는 에디트 거리 알고리즘에 근거해서 가장 작은 비용을 최소 비용으로 선택하여 삽입한다. cell(1,1)의 비용 계산은 다음과 같은 세 가지 경우로 나누어진다.

- cell(1,0) 자리로 부터 나오는 비용은 2 이다.
- cell(0,1) 자리로 부터 나오는 비용은 2 이다.
- cell(0,0) 자리로 부터 나오는 비용은 0 이다.

위 세 가지 경우의 비용이 각각(2, 2, 0)이므로, 가장 작은 비용을 선택하여 cell(1,1)자리에 삽입 한다. 여기에서는 0이 제일 작으므로 cell(1,1)자리에 들어가게 된다. 같은 방법으로 나머지 cell들을 채워보면 그림 3(b)와 같다.

		9	4	5	7
	0 cell(0,0)	1	2	3	4
9	1				
4	2				
8	3				
7	4				
6	5				

(a) 2차원 행렬 생성

		9	4	5	7
	0 cell(0,0)	1	2	3	4
9	1	0	1	2	3
4	2	1	0	1	2
8	3	2	1	1	2
7	4	3	2	2	1
6	5	4	3	3	2

(b) 산출된 에디트 거리 비용

(그림 3) 에디트 거리 비용 산출

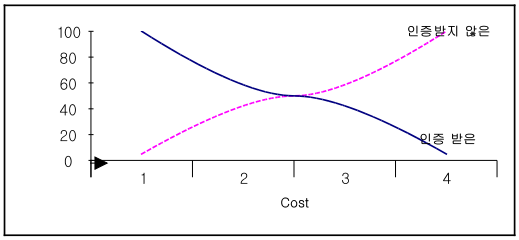
에디트 거리 알고리즘을 통해 전체 비용 = 2 라는 결과가 산출되었으므로 입력한 번호가 실제 번호와 같아지기 위해서는 (수정, 삭제, 삽입) 중에서 두 번의 비용이 소비되어야 실제 비밀번호와 같아진다는 것을 알 수 있다. 직관적으로, 사용자가 입력한 94876은 실제 비밀번호 9457이 되기 위해서 한 번의 수정과 한 번의 삭제가 되어야 실제 비밀번호와 같아질 수 있다.

3.2 사용자 유형 판별

이 절에서는 에디트 거리 알고리즘으로 산출된 비용으로 인증 받은 사용자와 인증 받지 않은 사용자를 구분하는 방법에 대해 설명한다.

가정 1. 사용자 입력 길이가 실제비밀번호의 길이와 같을 경우, 실제비밀번호와 사용자의 입력 값이 50%이상 정확하게 일치하면 인증 받은 사용자이다. 그렇지 않으면 인증 받지 않은 사용자이다.

그림 4는 이와 같은 가정을 바탕으로 사용자의 유형을 판별하는 그래프를 나타내고 있다. 인증 받은 사용자는 잘못 입력할 확률이 비용이 증가함에 따라 감소하며, 인증 받지 않은 사용자는 비용이 증가함에 따라 잘못 입력할 확률이 증가한다. 따라서, 에디트 거리 알고리즘을 통해 산출된 비용이 적을 경우 인증된 사용자일 확률이 높으며, 비용이 크게 나올 경우 인증 받지 않은 사용자로 간주한다.



(그림 4) 사용자 유형 판별

또한 사용자가 비밀번호를 밀려 쓰는 경우에 대해서 쉬프트 연산을 통해 처리한다.

3.3 에디트 거리 알고리즘이 적용된 디지털 도어락

표 1은 디지털 도어락의 상태를 나타내는 변수와 그 설명을 나타낸 것이다. 이 상태 변수는 그림 5의 디지털 도어락 상태 전이도[4]에서 사용된다.

<표 1> 전이 상태 및 변수 설명

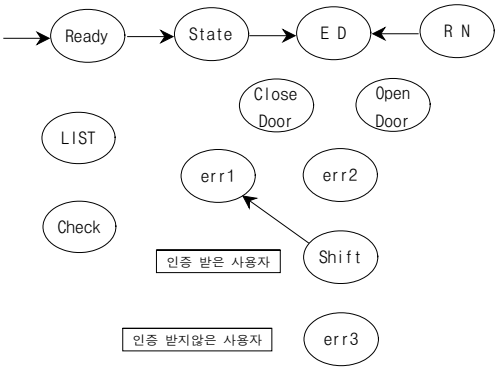
상태/변수	설 명
Ready	# 버튼
State	사용자 입력
ED	에디트 거리 알고리즘
RN	실제 비밀번호
Open door	문 개방
Close door	문 폐쇄
Shift	shift 연산
str	사용자가 입력한 문자열 저장
mid_value	실제 비밀번호 길이 중간 값
LIST	사용자 실수 유효범위 리스트
Check	실수 범위를 검사

그림 5는 에디트 거리 알고리즘을 디지털 도어락에 적용했을 때 디지털 도어락의 상태 전이도를 나타내고 있다. 디지털 도어락 상태는 다음과 같이 변화한다.

1) 임의 사용자가 #버튼을 누르면 Ready상태로 전이된다.

2) Ready상태에서 사용자가 번호키(버튼)를 누르면 State 상태로 전이되며, 입력된 번호는 str(사용자 입력 저장)변수에 stack방식으로 저장된다.

3) 사용자 입력이 완료 후, 다시 #버튼을 누르게 되면 ED(edit distance)상태로 전이되며, ED상태에서는 RN(실제비밀번호)와 str에 저장된 값을 에디트 거리 알고리즘을 사용하여 비용을 측정한다. 이 논문에서는 사용자가 입력한 값이 실제비밀번호의 50%이상 일치할 경우에만 인증 받은 사용자로 간주하므로 실제비밀번호의 길이를 구하고 그 길이의 반을 나누어 중간 값을 구한다.



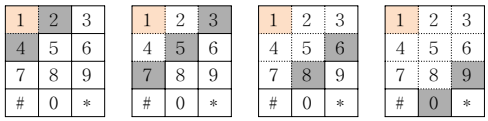
(그림 5) 디지털 도어락 상태 전이도

4) ED상태에서 출력한 비용이 0일 경우, 임의 입력한 문자열과 실제 비밀번호가 일치하므로 Open Door상태로 전이되어 문이 열리게 된다. 비용이 0이 아닐 경우에는 Close Door상태로 전이된다.

5) Close Door상태는 err1 상태와 err2 상태로 다시 분기되는데 이는 입력한 값이 실제비밀번호와 일치하는 정도에 따라 인증 받은 사용자와 인증 받지 않은 사용자로 판별하기 위해 구분된 것이다. 이때, 비용이 50% 미만일 경우 err1상태로 전이되며, 비용이 50% 이상일 경우 err2상태로 전이된다.

6) err1상태는 인증 받은 사용자로 인식하고, err2상태에서는 인증 받지 않은 사용자로 인식하여 인증 받지 않은 사용자가 입력한 값을 쉬프트 연산한다. 왜냐하면, 인증 받은 사용자가 비밀번호를 밀려 쓸 수 있기 때문이다. 이때, 쉬프트 연산하여 산출된 비용이 50% 미만으로 작아진다면 err1상태로 전이되고 그렇지 않으면 err3상태로 전이되어 인증 받지 않은 사용자로 분류된다.

추가적으로, 사용자 실수 허용 범위도 임의적으로 결정할 수도 있다. 그림 6은 사용자가 실수로 번호를 잘못 눌렀을 경우에 대한 예이다.



(그림 6) 실수 허용 범위

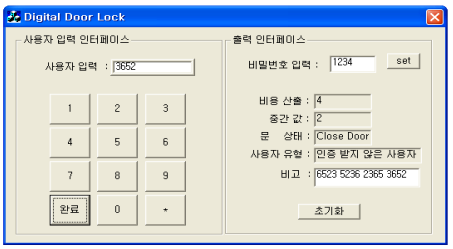
그림 6(a)와 같이 실제 입력해야 할 비밀번호가 1번인데 실수로 사용자가 다른 번호를 입력할 경우, 잘못 입력 할 수 있는 근접 번호들의 범위를 묶어 등급화 시킨다. 그림 6(a)에서 실제 눌러야 할 번호가 1이라면 실수 허용 범위는 2번과 4번이다. 따라서, 2번과 4번에 대한 비용은 1로 나타낸다. 같은 방법으로 1에 대해서 나머지 범위와 거리를 구해보면, 그림 6(b)의 3번, 5번, 7번에 대한 비용은 2가 되고, 그림 6(c)의 6번과 8번에 대한 비용은 3, 그림 6(d)의 9번과 0번에 대한 비용은 4가 된다. 이와 같은 계산을 통해 각각의 그룹에 대해서 사용자가 실수로 입력할 수 있는 확률은 (2, 4) > (3, 5, 7) > (6, 8) > (9, 0)이 된다. 이와 같은 방식으로 실수 허용 범위를 정의함으로써 인증 받은 사용자에 대한 신뢰도를 추출해낼 수 있다.

신뢰도를 추출하기 위해 LIST와 Check 상태 변수를 정의한다. LIST는 실수 허용 범위를 저장하고 있고, Check는 사용자가 입력한 값들 중에서 틀린 번호를 LIST와 대조하는 데에 사용된다. 두 변수의 대조를 통해 err1상태로 넘어온 사용자 입력 값이 실제비밀번호에 비해 거리가 얼마나 떨어졌는지 확인해 볼 수 있다.

실수 허용 범위의 정의는 디지털 도어락에서 사용자가 초기 설정 시 보안 수준 설정에 활용하거나 타이머를 두어 사용자가 자주 사용하는 시간대에만 도어락의 보안수준을 표준으로 설정해놓고, 사용자가 없는 시간 때에는 보안수준을 높이는 방식 등으로 활용될 수 있다.

4. 성능 평가

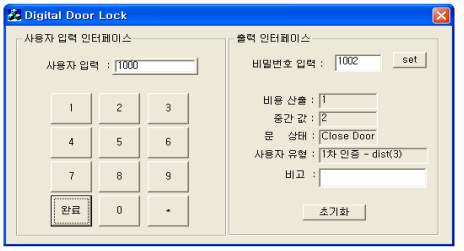
성능 측정은 모의실험을 통해 평가하였다. 비밀번호로 설정된 값과 사용자가 입력한 값을 바탕으로 제안한 방법을 통해 인증 받은 사용자와 인증 받지 않은 사용자를 판별하였다.



(그림 7) 모의실험1

그림 7은 모의실험 결과를 보여주고 있다. 초기 비밀번호가 1234로 설정되어 있고, 사용자가 3652를 입력했을 때, 산출 비용은 4가 되므로 사용자 유형은 인증 받지 않은 사용자로 판별된다. 따라서, 도어락 상태는 Close Door 상태가 된다. 비고란은 사용자가 밀려쓸 경우를 고려해 쉬프트 연산을 통해 비용을 재측정해본 것이다.

그림 8은 실제비밀번호가 1002이고, 사용자의 입력 값이 1000일 때의 모의실험 결과이다. 이 실험에서는 사용자의 입력 값에서 0를 2로 교체해야 하므로 비용 1이 나왔으며, 사용자가 입력한 번호가 50%이상 정확하므로, 1차 인증이 되었다. dist(3)은 사용자 실수 범위를 나타낸 것이다. 이는 사용자가 실수로 입력한 0과 실제 입력해야 할 번호 2의 거리가 3이라는 것을 뜻한다.



(그림 8) 모의실험2

5. 결론

본 논문에서는 에디트 거리 알고리즘을 활용하여 비밀번호를 이용한 디지털 도어락에서 인증 받은 사용자와 인증 받지 않은 사용자를 효과적으로 판별해낼 수 있는 방법을 제안하였다. 에디트 거리 알고리즘을 활용하여 사용자가 입력한 비밀번호에 대해 (추가, 삭제, 삽입 등)에 필요한 비용을 산출해 내고 또한 인증 받은 사용자의 실수 허용 범위를 정의함으로써 비밀번호를 이용한 디지털 도어락의 보안성을 강화할 수 있다. 실험을 통해 제안하는 방법이 실제 디지털 도어락에 적용되어 활용될 수 있음을 증명하였다. 향후 연구에서는 다양한 모의실험을 통해 제안 방법의 성능을 평가하고, 또한 타이머 설정 등의 방법을 통해 보안 수준을 설정하여 보안성을 강화시킬 수 있는 연구가 고려되어야 할 것이다.

6. 참고 문헌

[1] 유보현, “암호화기술을 적용한 무선 도어락시스템 디자인에 대한 연구”, 디자인학연구 통권 제55호(Vol.17 No.1), pp. 179-190, 2004.

[2] Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, Clifford, "Introduction to Algorithms," The MIT Press, 2001.

[3] Dan Gusfield, "Algorithms on Strings, Trees, and Sequences," Cambridge University Press, 1997

[4] Hopcroft, John E. and Motwani, Rajeev and Ullman, Jeffrey D., “Introduction to Automata Theory, Languages, and Computation,” Addison Wesley, 2000.