DID인증기술_CapStoneDesigm

난수를 이용한 세션 키 및 비밀번호 생성 프로그램 구축

• 지도 교수 : 신 승 수
• 학　　과 : 정보보호학과
• 팀　　명 : CDT

목 차

그림 목차

표 목차

Ⅰ.서론

1. 연구 배경

인터넷 통신의 발전과 함께 사용자들의 스마트폰 또는 개인 컴퓨터를 통한 다양한 온라인 서비스의 이용이 가속화되고 있다. 따라서 다수의 이용자가 온라인 서비스를 이용하는 환경에서 불법적 사용을 통제하기 위하여 사용자를 확인하는 사용자 인증 기술이 발달했고, 이에 대한 중요성도 증가되고 있다.[3]

온라인 서비스에서 사용되는 가장 일반적인 사용자 인증 기술은 비밀번호 인증 방식이다. 인터넷 사이트를 통하여 온라인 서비스를 이용할 때, 사용자는 자신이 설정하여 서버에 저장된 ID(Identifier)와 비밀번호(Password)를 이용하여 해당 인터넷 사이트에 로그인을 수행한다.[2]

대부분의 사용자가 비밀번호를 설정 행태는 심하게 좋지 않다. 비밀번호의 상위 항목 1,2위는 여전히 '123456'과 ' password'이며, 사용자들은 비밀번호를 너무 자유롭게 공유하고 자주 잊어버린다. 그에 따라 사용자들은 공통적인 하나의 비밀번호를 지정해 각종 웹사이트 및 로그인의 비밀번호로 사용한다. 사용자의 온라인 보안을 보장할 수 있는 비밀번호가 가장 큰 장애물이 된 것이다. 이는 좋은 비밀번호 관리자(Password Manager)를 사용해야 하는 이유이기도 하다.[1]

가장 강력한 비밀번호를 만들기 위해서는 길고 랜덤한 문자열이고 각각 영대소문자,숫자, 특수문자가 들어가야 한다. 사용하는 각 사이트에서 다른 비밀번호를 사용해야 한다는 것을 사용자들은 알고 있지만 막상 랜덤한 문자열을 생성한다는 것 자체가 사용자들에겐 어려운 일이다.

당연히 비밀번호 생성 요구 사항들을 맞춰 비밀번호를 생성한다고 해도 해킹을 당할 수 있고 키로거, 트로이목마 등 악성코드에 감염 되어 비밀번호를 유출당할 수 있다. 하지만 이러한 경우들은 본인들의 인터넷 서비스를 사용할 때 각별히 주의를 해야 한다.

이에 따라 세션 키로 사용할 값들 또한 난수를 통해 만들어져야 하며 사람이 인위적으로 새성해서는 안 된다.

2. 연구 목적

인터넷 서비스 사용자들은 각 요구 정책에 맞춰 비밀번호를 생성할 의무가 있다. 하지만 사용자들은 결국 그 요구사항마저 간단하게 만들어 비밀번호를 사용하는데 대표적인 예로 본인의 이름을 영대소문자1! , 자신의 생년월일을 영어로 치고,특수기호 하나를 넣고 숫자를 넣는 등 본인들이 쉽게 기억할 수 있도록 나름의 패턴을 만드는 것이다. 하지만 이러한 패턴을 가진 비밀번호는 결국 해커들의 첫 번째 공격 루트가 되므로 사용자들은 요구 정책에 맞는 랜덤한 문자열을 사용하여야 한다.

이러한 점들을 예방하기 위해 무작위성,불규칙성,우연성이 충족되는 난수를 생성하는 의사난수 생성기를 이용해 요구조건에 맞는 비밀번호 및 세션 키 길이를 유지하여 랜덤한 문자열을 생성해 해커들의 무작위 공격에 대한 예방을 하고자 본 연구를 진행한다.

Ⅱ. 동향 분석

본 장에서는 다양한 온라인 서비스를 이용하는 현 시점에서 사용자들의 비밀번호에 대한 인식에 대한 부족, 비밀번호 보안의 중요성에 대해 얼마나 무관심한지에 대한 관련 동향에 대해 파악한다. 해당 논문에서는 비밀번호의 생성 및 관리의 중요성에 대한 것을 위주로 기술한다.

1. 비밀번호 사용자들의 평소 사용 동향

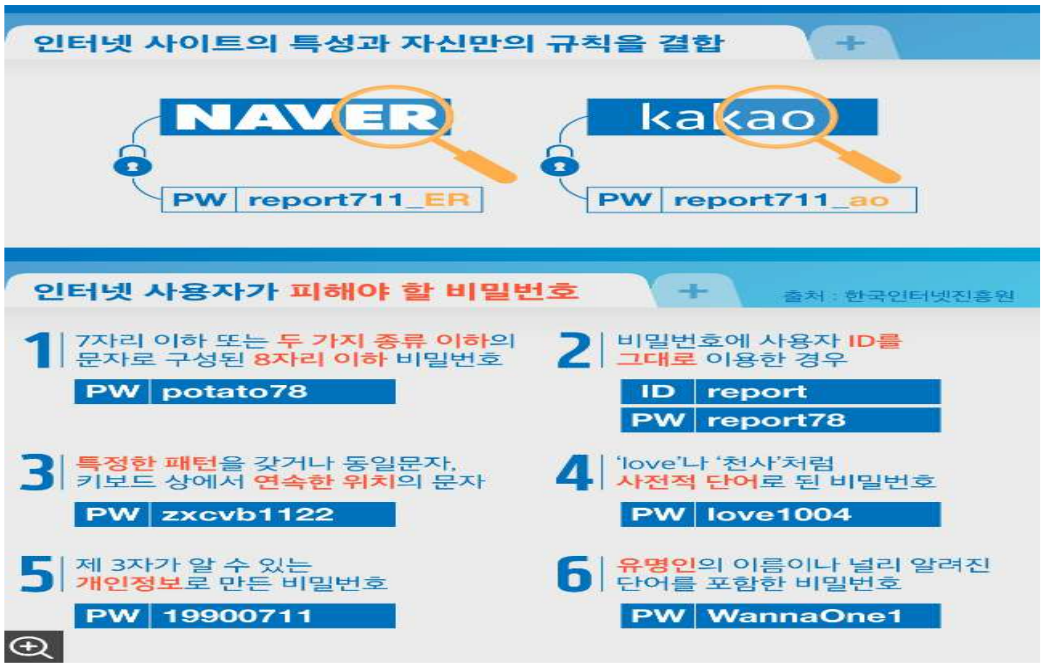
가. 사례

비밀번호란 비밀 인증 데이터의 한 형식으로 어떠한 자원에 대한 접근을 제어하는 데에 사용이 된다. 비밀번호는 허용되지 않는 접근으로부터 비밀을 보호하며, 접근권을 얻으려는 사람들에게 비밀번호를 물어서 접근 여부를 결정한다. 자신의 데이터에 접근하는 것을 보호하기 위해 생성하는 것이 비밀번호인데 사용자들은 그 비밀번호를 생성할 때 대게 무난하고, 일정한 패턴을 생성하여 만든다. 사람들이 가장 많이 사용하는 비밀번호는 다음 [그림2-1]과 같다.

[그림2-1] 사람들이 가장 많이 사용한 비밀번호

나. 피해야 할 비밀번호

그림[2-1]과 같이 사람들은 비밀번호 생성에 대해 무관심하고 큰 중요성을 느끼지 못한다. 그러기에 이와 같은 비밀번호를 생성하고 사용하는 것이다. 온라인 서비스를 이용하는 사용자들이 피해야 할 비밀번호들은 다음 [그림2-2]과 같다.



[그림 2-2] 인터넷 사용자가 피해야 할 비밀번호

다. 비밀번호 요구 정책

보안 네트워크 환경에서는 모든 사용자가 8자리 이상의 문자를 포함하고 문자,숫자 및 기호의 조합을 포함하는 강력한 암호를 사용해야 한다.

2. 난수

본 절에서는 비밀번호를 생성하기 위한 난수에 대한 기본 개념 및 성질과 난수를 생성하기

위한 설계 및 구현 과정에 대해 기술한다.

가. 난수의 정의

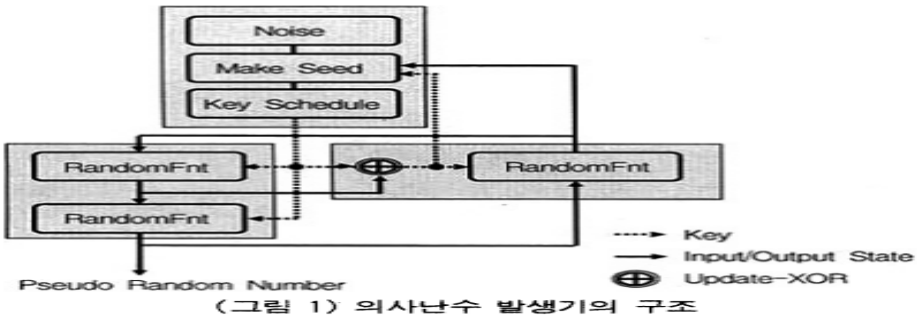
주사위를 던져서 나온 점의 수와 같이 어떤 확률적인 규칙에 따라 우연히 발생하는 수이며 숫자를 발생시키는 과정의 확률적인 성질에 대해서 기술되는 것이다. 이러한 난수가 생성되기 위해서는 난수의 3가지 성질을 만족하여야 하는데 무작위성, 예측 불가능성, 재현 불가능성이다. 또한 난수는 규칙성을 갖고 있으면 안 되며 특정 주기를 갖기 시작한 그 순간부터 그 숫자 및 문자는 참된 난수가 아니게 된다.

하지만 컴퓨터는 사람들이 일반적으로 알고 있는 의미의 랜덤으로 난수를 생성하는 것이 아니라 납득할만한 우연성을 만들어 컴퓨터는 이것을 랜덤이라고 정의한다. 이에 따라 참 난수(True random number)를 발생시키는 것은 현실적으로 불가능하기 때문에 난수와 구별하기 어려운 의사 난수(Pseudo random number)를 생성하여 난수로 활용한다.

의사난수는 처음에 주어지는 초기값을 이용하여 이미 결정되어 있는 메커니즘에 의해 생성되는 수이다. 난수는 생성 방법이 결정되어 있지 않으며 다음에 생성될 값을 전혀 예측할 수 없다. 하지만 의사난수 생성기에 의해서 생성되는 수는 초기값을 안다면 계산될 수 있으므로 난수와 구별하여 의사난수라 칭한다.

나. 의사난수의 생성과정

의사 난수 생성기는 크게 3개의 알고리즘으로 나누어진다. 첫 번째는 Noise(Time 값)을 이용해 생성한 seed(종자)를 새롭게 갱신시키는 reseeding 알고리즘, 두 번째는 seed로부터 의사 난수를 생성하는 랜덤화 알고리즘이며, 세 번째는 랜덤화 알고리즘의 입력 값을 새롭게 갱신시키는 업데이트 알고리즘이다. [4] 과정은 다음 [그림3-1]과 같다.



[그림3-1] 의사난수의 생성과정

다. 의사난수 생성시 고려 사항

의사난수를 생성하기에 앞서 고려할 사항은 다음과 같다.

(1) seed(종자)를 추정하기 힘들도록 엔트로피를 높인다.
(2) 키와 seed에 대한 관리가 필요하다.
(3) 키가 노출된 뒤 역추적 공격에 저항성이 필요하다.
(4) 긴 주기를 가지고 있어야 한다.[4]

Ⅲ. 난수를 이용한 세션 키 및 비밀번호 생성 프로그램

본 장에서는 난수를 이용한 비밀번호를 생성하는 프로그램의 설계 및 구현을 진행하였으며, 난수를 생성하는 코드 및 EXE 프로그램 파일 생성 과정으로 나누어 상세히 설명하고 C++과 Python 두 언어를 비교하여 효율적인 프로그램을 구축한다.

1. 설계 및 구현

본 절에서 난수를 생성하기 위한 EXE 프로그램 파일에 대한 설계 및 구현 과정을 기술한다

가. 개발 환경

프로그램 동작 환경은 Windows를 대상으로 하며 사용된 프로그램은 Visual Studio 2019, IDLE(Python 3.10 64bit)을 사용했으며 각각 언어는 C++, Python을 사용했다. Visual Studio 2019 및 IDLE(Python 3.10 64bit) 를 사용하기 위한 환경은 [표3-1]과 같다.

[표 3-1] 개발 환경

사용 프로그램	운영체제	하드웨어	추가 요구 사항
Visual Studio 2019	Windows 10 버전 1703 이상: Home, Professional, Education 및 Enterprise(LTSC 및 S는 지원되지 않음)	1.8GHz 이상의 프로세서 쿼드 코어 이상 추천 2GB RAM(8GB RAM 추천) 하드 디스크 공간: 설치된 기능에 따라 최소 800MB, 최대 210GB의 사용 가능한 공간이 필요하며, 일반적인 설치에는 20~50GB의 사용 가능한 공간이 필요하다. 최소 디스플레이 해상도 720p(1280x720)를 지원하는 비디오 카드.	C++, JavaScript 또는 .NET 워크로드를 사용하여 모바일 개발을 설치하려면 Windows 7 SP1에 PowerShell 3.0 이상이 필요하다.
IDLE(Python 3.10 64bit)	Windows 7 이상, Mac, Linux 운영체제 지원	1 GHz 이상의 프로세서. 인텔 i5 이상 최소 4기가바이트(GB) 메모리(RAM), 8기가 이상 256GB 하드 디스크 공간	x

나. 프로그램 설계 과정

난수 생성을 위해서 필요한 기능들을 동작 과정을 통해 기술한다.

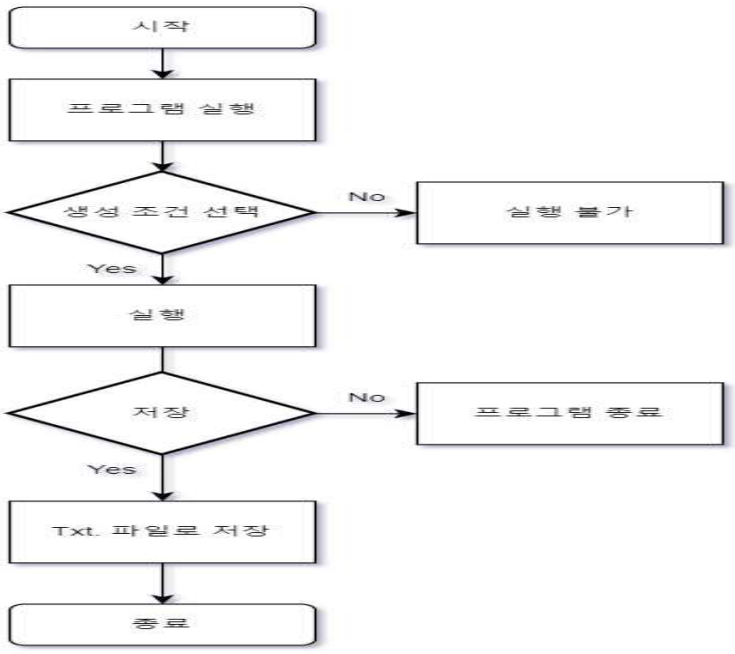
사용자는 프로그램을 이용해 자신이 생성하고자 하는 비밀번호의 개수, 비밀번호의 길이, 구별하기 어려운 문자 제거, 추가하고 싶은 문자열 추가, 생성한 비밀번호를 txt 파일로 저장으로 각 기능들에 대한 설명은 다음과 같으며, 참고 그림은 [그림3-2]과 같다.

(1) 생성 조건 선택

우선 비밀번호를 생성하기 위해 3가지 조건을 선택해야 한다. 첫 번째로 비밀번호를 생성하기에 앞서 생성할 비밀번호의 개수와 길이를 선택한다. 두 번째로 숫자, 특수문자, 영어 대소문자, 애매한 단어 제거 부분을 선택한다. 여기서 애매한 단어란 우리들이 눈으로만 봐서는 구분이 안 되는 문자들을 말하며 예로는 대소문자 I, 대소문자 L, 대소문자 O, 숫자 1과 2가 있다. 세 번째로는 “자신이 원하는 문자열을 넣을 것인가”이다. 자신이 원하는 문자열을 넣으면 기억하기 쉽기 때문에 이러한 기능을 추가한다.

(2) 생성한 비밀번호 저장

생성한 비밀번호(난수)들을 관리하기 편하게 하기 위해서 .txt 파일 형식으로 저장을 한다. 후에 생성한 파일을 따로 암호화를 진행하여 비밀번호(난수)를 관리하고 보관한다.



[그림3-2] 프로그램 동작 과정 순서도

다. 난수 생성 프로그램 구축 및 구현

설계 과정을 바탕으로 프로그램을 구축하며 두 언어의 결과물을 코드 위주로 기술한다.

(1) EXE 파일의 GUI 설정 및 랜덤 난수를 생성하기 위한 헤더 파일 입력

“mainwindow.h”와 “ui_mainwindow.h”는 Windows 운영체제에서 프로그램의 GUI의 동작 및 제공을 담당하는 헤더 파일로서 EXE 파일을 생성하기 위해 필요한 헤더 파일이다.

“gfile.h”, “string”, “regex”, “QFileDialog”, “QMessageBox”, “time.h”는 각각 파일 저장, 문자열 클래스 추가, 항목 검색 및 수정, 경로 선택, 대화창 추가, 마지막으로 “time.h”은 seed 값을 초기화하기 위해 사용되는 시간 관련 클래스를 추가하는 헤더 파일로서 가장 중요한 파일이라고 할 수 있다. 각 헤더 파일의 표기법은 [그림3-3]과 같다.

```C++
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <qfile.h>
#include <string>
#include <regex>
#include <QFileDialog>
#include <QMessageBox>
#include <time.h>
```

[그림3-3] 헤더파일 코드

(2) 비밀번호 개수 설정

비밀번호(난수)를 생성하기에 앞서 만들고자 하는 비밀번호의 개수와 길이를 선택한다. 개수와 길이를 선택함에 있어 비밀번호의 사용 용도에 따라, 혹은 제한된 자릿수에 맞춰 비밀번호

를 맞춰서 생성할 수 있다. 코드는 [그림3-4]와 같다.

```
MainWindow::MainWindow(QWidget *parent) // 메인 윈도우 호출
: QMainWindow(parent)
, ui(new Ui::MainWindow)
{
ui->setupUi(this);
ui->CountBox->setMinimum(1); // 비밀번호 만드는 갯수 최솟값
ui->CountBox->setMaximum(99999999); // 비밀번호 만드는 갯수 최댓값
ui->LengthBox->setMinimum(1); //비밀번호 길이 최솟값
ui->LengthBox->setMaximum(99999999); // 비밀번호 길이 최댓값
}
```

[그림3-4] 비밀번호 개수 설정 코드

(3) seed값 초기화

비밀번호(난수)를 생성할 때 마다 seed(종자)값을 계속 초기화 시켜줘야 하는데 time함수를 통해서 그 비밀번호를 생성할 때의 시간값을 랜덤값 2개와 함께 더해 seed값 하나를 생성하고 이 과정을 비밀번호를 만드는 개수만큼 반복이 되기 위해 for문을 사용하여 코딩을 진행한다. 코드는 [그림3-5]과 같다.

```
srand(time(NULL)+ran*ran2); // 시드를 시간값 + 랜덤값1 * 랜덤값2 으로 설정
for(int i = 0; i<ui->CountBox->value(); ++i) // 비밀번호 만드는 개수만큼 반복
{
    QString value2 = ui->AddText->toPlainText(); //비밀번호 추가값
std::string charSet(""), // 비밀번호 글자 모음 (숫자,특문,대문자,소문자)
            output(""); // 비밀번호
```

[그림3-5] seed값 초기화 과정

(4) 비밀번호 생성에 필요한 선택지 생성

비밀번호를 생성하기에 앞서 체크박스 안에 비밀번호를 생성 시 추가하고 싶은 조건을 체크한다. 코드와 같이 확인을 통해 사용자가 체크한 조건들로만 비밀번호를 생성하게 프로그램을 설정한다. 코드는 [그림3-6]과 같다.

```
    QString value2 = ui->AddText->toPlainText(); //비밀번호 추가값
std::string charSet(""), // 비밀번호 글자 모음 (숫자,특문,대문자,소문자)
            output(""); // 비밀번호

            if(ui->checkNum->isChecked()) // 숫자추가 체크 확인
            {
                charSet+="0123456789"; // 비밀번호 글자 모음에 숫자를 추가
                    }
            if(ui->checkSpe->isChecked()) // 특수문자추가 체크 확인
            {
                charSet+="!@#$%^&*"; // 비밀번호 글자 모음에 특수문자를 추가
            };
            if(ui->checkLow->isChecked()) // 소문자추가 체크 확인
            {
                charSet+="abcdefghijklmnopqrstuvwxyz"; // 비밀번호 글자 모음에 소문자를 추
가
            };
            if(ui->checkUp->isChecked()) // 대문자추가 체크 확인
            {
                charSet+="ABCDEFGHIJKLMNOPQRSTUVWXYZ"; // 비밀번호 글자 모음에 대문자를 추
가
            };
```

☐ Numbers ☐ Special
☐ Upper Case ☐ Lower Case

[그림3-6] 비밀번호 선택지 생성 코드

(5) 비슷한 문자 제거

비밀번호를 생성하다 보면 앞서 동작 과정에서 설명했듯이 사람이 육안으로 보고는 애매한 문자들을 제거하는 과정의 체크 박스이다. 이 박스를 체크하면 동작 과정에서 설명한 애매한 문자들은 제거된 상태로 비밀번호 생성에 들어간다. 코드는 [그림3-7]과 같다.

```
    };
    if(ui->checkExc->isChecked()) // 비슷한 문자 제거 체크 확인
    {
        charSet = std::regex_replace(charSet, std::regex("i"), "");
        charSet = std::regex_replace(charSet, std::regex("l"), "");
        charSet = std::regex_replace(charSet, std::regex("1"), "");
        charSet = std::regex_replace(charSet, std::regex("L"), "");
        charSet = std::regex_replace(charSet, std::regex("o"), "");
        charSet = std::regex_replace(charSet, std::regex("0"), "");
        charSet = std::regex_replace(charSet, std::regex("O"), "");
    }
```

☐ Exclude ambiguous characters

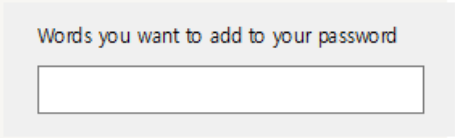
[그림3-7] 비슷한 문자 제거 코드

(6) 원하는 문자열 입력

비밀번호의 기억에 용이하게 하기 위해서 자신이 원하는 문자열을 입력이 가능하도록 한다. 이러한 기능은 상황별로 맞춰 비밀번호를 생성하기에 용이하며 코드는 [그림3-8]과 같다.

```
            if(value2.length() != 0) //추가 비밀번호값이 있으면 실행
            {
                output+=value2.toStdString(); //추가 비밀번호값 먼저 추가
                for(int i = value2.length(); i<ui->LengthBox->value(); ++i) //비밀번
호 길이만큼 반복
                {
                    output+=charSet.at(rand()%(int)charSet.length()); // 비밀번호 글
자 추가
                }
            }
            else // 추가 비밀번호값이 없으면 실행
            {
                for(int i = 0; i<ui->LengthBox->value(); ++i) // 비밀번호 길이만큼 반
복
                {
                    output+=charSet.at(rand()%(int)charSet.length()); //비밀번호 글자
추가
                }
            }
                ui->plainTextEdit->appendPlainText(output.c_str()); // 비밀번호 출력

        }
```



[그림3-8] 원하는 문자열 입력 코드

(7) 저장

최종적으로 비밀번호 생성을 마치면 생성한 비밀번호들을 save 버튼을 통해서 Txt, 파일로 전환하여 저장한다. 코드는 [그림3-9]과 같다.

```
void MainWindow::on_SaveButton_clicked() // 저장버튼이 눌렸을때
{
QString txtFile = QFileDialog::getSaveFileName(this, tr("Save file"), "",
tr("text file (.txt);;C++ File (.cpp *.h)")); // 파일경로 선택, 파일이름 정하기
if (txtFile != "") // 경로가 선택 되었을 경우
{
QFile file(txtFile);
```

```
        if (file.open(QIODevice::ReadWrite)) // 텍스트 파일 열기 성공시
        {
        QTextStream stream(&file);
            stream << ui->plainTextEdit->toPlainText(); // 나온 비밀번호들을 복사하기
            file.flush(); // 텍스트 파일에 붙여넣기(쓰기)
            file.close(); // 텍스트 파일 닫기
        }
        else // 텍스트 파일 열기 실패시
        {
            QMessageBox::critical(this, tr("Error"), tr("Error")); // 실패시 에러메세지
            return;
        }
    }
```

Save

[그림3-9] 저장 코드

Ⅳ. 분석

본 장에서는 C++ 언어로 진행한 의사 난수 생성기와 Python 언어를 이용한 의사 난수 생성기의 난수를 생성하는 속도 및 효율성을 비교하는 과정을 통해 비교하며 분석을 한다.

1. 언어 비교

제일 먼저 Python을 이용한 의사 난수 생성 코드를 작성하였는데 멀티 스레딩(Multi-threading)을 이용하여 자원의 생성과 관리의 중복성을 최소화하여 수행 능력을 향상시키고자 하였지만, Python의 GIL(Global Interpreter Lock) 정책으로 인해 Python의 인터프리터(interpreter)가 한 스레드만 하나의 바이트코드를 실행시킬 수 있도록 Lock 해주기 때문에 이 논문에서 추구하고자 하는 속도는 낼 수 없다고 판단하여 Python은 난수를 생성하기 적합하지 않은 언어라고 판단했다.

그러나 C++을 이용하였을 때는 Python 과는 달리 메모리의 관리가 보다 세분화되고 제어된 수준에서 처리가 되며, 속도적인 측면으로써는 Python보다 확실히 빠르고 효율적인 측면을 보여주었다. 두 언어 간의 차이점은 <표4-1>과 같다.

<표4-1> Python과 C++ 언어 비교

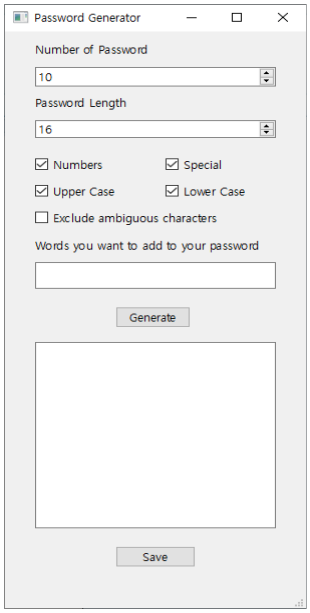
	Python	c++
속도	느림	빠름
메모리 효율	낮음	높음
용도	웹 어플리케이션 개발	순수 응용프로그램 개발
오류 발생	높음	낮음

2. 프로그램 모의 테스트

사용자가 프로그램에서 신규 비밀번호를 생성을 할 때 선택지에 따라 특정 선택지로만 만들어지는지 확인한다. 다음은 사용자가 16자리의 비밀번호를 숫자, 소문자, 대문자, 특수문자

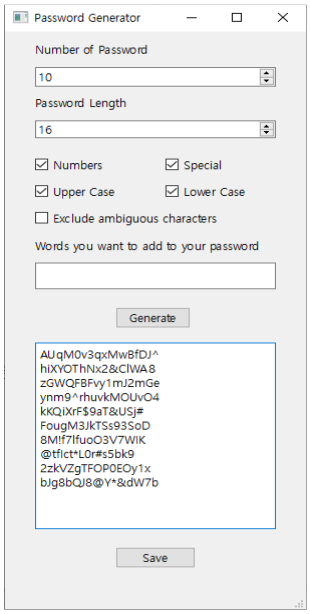
를 조합하여 10개의 비밀번호를 만들고 저장하는 과정을 기술한다.

(1) 사용자가 원하는 비밀번호의 개수와 길이를 설정하고, 조합할 문자들을 선택한다. 선 택한 화면은 [그림4-1]과 같다.



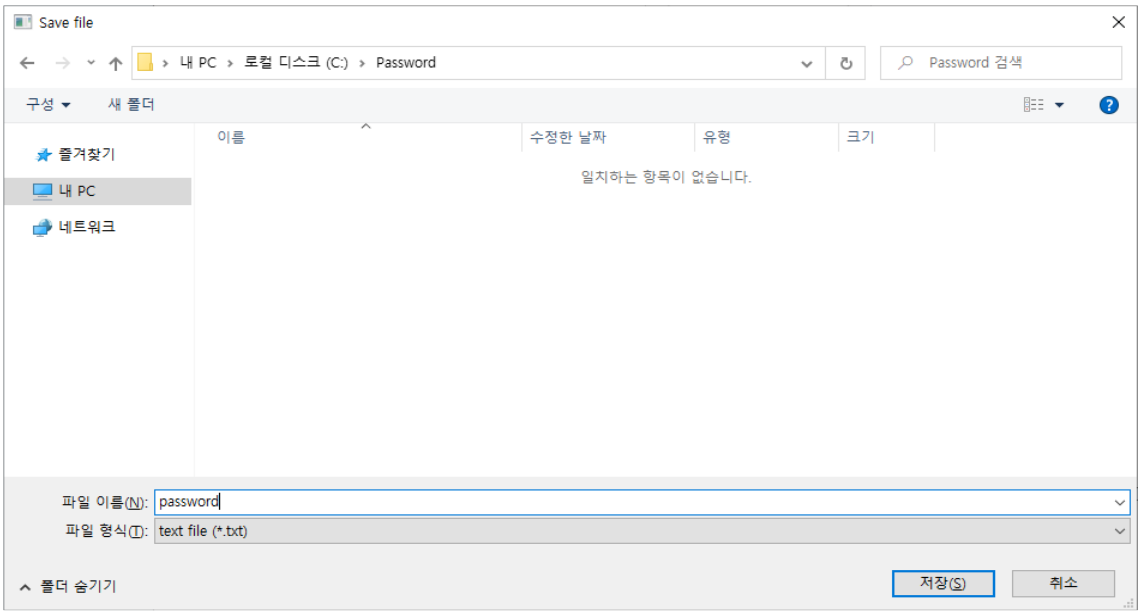
[그림4-1] 프로그램 UI

(2) 사용자가 설정을 마쳤으면 “Generate“ 버튼을 눌러 비밀번호를 출력한다. 출력된 화면은 [그림4-2]과 같다.



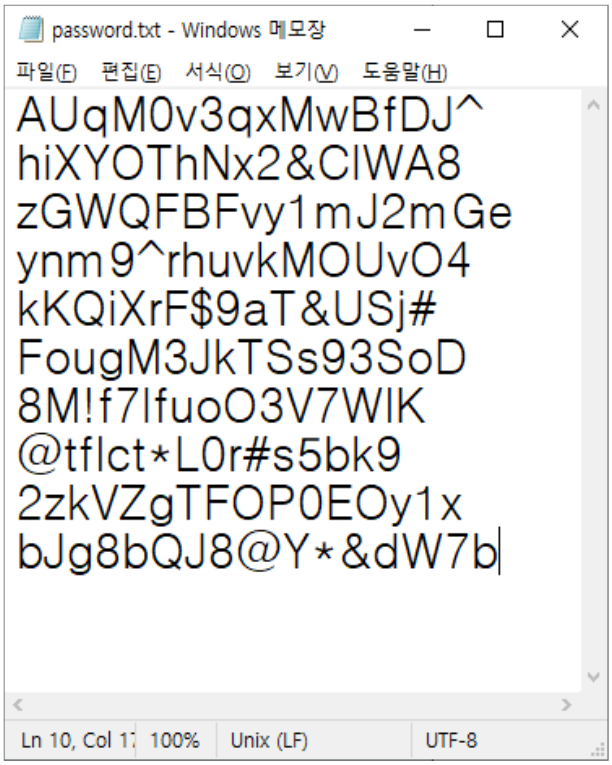
[그림4-2] 출력 과정

(3) 비밀번호가 출력된 값을 저장을 하기 위해 Save 버튼을 누른다. 그 후 원하는 경로를 설정하고 사용자가 원하는 파일의 이름을 정한다. 저장하는 화면은 [그림4-3]과 같다.



[그림4-3] 저장 과정

(4) 비밀번호가 텍스트 파일로 잘 저장되었는지 확인한다. 텍스트 파일의 화면은 [그림4-4]과 같다.



[그림4-4] 저장된 .txt파일 확인

Ⅴ. 결론

본 논문에서는 난수를 이용한 세션 키 및 비밀번호를 생성하기 위해 rand 함수 및 Time 함수를 이용해 랜덤한 난수를 생성 하는 프로그램을 제안하였다.

현재 온라인 서비스 사용자들의 자주 사용하고, 사용하면 안 되는 비밀번호에 대한 사례들을 알아보고 그로 인한 해킹의 위험성과 비밀번호 설정의 중요성을 상기시켜주었다. 프로그램 생성을 위해 Visual Studio 2019, IDLE(Python 3.10 64-bit)를 통해 각각 C++, Python으로 비교를 진행하고자 하였으나 Python의 Gil(Global Interpreter Lock) 정책으로 인해 Python은 효율성을 따지는 난수 생성 프로그램으로서는 적합하지 않다고 판단하였다.

난수를 생성 할 때에 각종 선택지를 주어 본인들이 원하는 난수(비밀번호)를 만들게 하였으며, 프로그램을 통해 만들어진 난수를 .txt 파일로 저장하는 기능 또한 추가하며 생성한 난수의 관리를 용이하게 만들었다.

난수 생성 프로그램으로 인해 전수 키 공격 법, 통계적 분석 공격 등으로부터의 공격에 대해 안전한 세션 키 및 비밀번호를 생성할 수 있을 것이다.

현재 비밀번호를 생성하기 위해서는 영문, 숫자, 특수문자를 조합하여 비밀번호를 생성하고 사용하고 있지만 추후 범용성을 고려하지 않아도 되는 비밀번호를 생성을 할 때 한글을 비밀번호로 사용한다던가, 혹은 다른 나라의 언어나 단어들을 조합하여 비밀번호를 생성한다면 그에 대한 경우의 수 또한 무수히 많이 늘어날 것이며 앞서 말한 전수 키 공격 법, 통계적 분석 공격들에 대한 대비책이 될 수 있을 것이다.

참고문헌

[1] ITWORLD,“최고의 비밀번호 관리자와 선택 시 고려해야 할 기능“, 보안뉴스, 2021.08
[2] KarenScarfone, Murugiah Souppaya,“Guide to EnterprisePasswordManagement”, p.11-13, NIST,2009
[3] 강정하, 김재영, 김은기 / 변동형 비밀번호 생성방법 및 이를 이용한 인터넷 인증 시스템에 관한 연구/ 한국산학기술학회 / 한국산학기술학회논문지 제 14권 제 3호 / 2013.03
[4] 송정환, 현진수, 구본욱, 장구영, 블록 암호 알고리즘기반 의사난수발생기 제안과 안전성 분석, 한국정보처리학회, 2002.12 ㅡ나 부분의 의사난수 생성 과정