Password Maker Overview

Password Maker is software that creates complex passwords from easy to remember.

Password Maker basically creates a complex password from following three methods:

- Two keywords
- One keyword and one file
- Two files

This software creates complex passwords from those that are easy to remember and those that can be remembered.

You can also use the key file to generate a password.

This is to create a password that can only be generated by the person who has the key file

The "key file" used by this software is an encrypted 32-bytes (256-bits) random number.

When generating a password, the “key file” is decrypted and a random number with a length of 32 bytes (256 bits) is extracted.

And this random number also participates in password generation.

- 32-bytes random number + Two keywords
- 32-bytes random number + One keyword and one file
- 32-bytes random number + Two files

Therefore, the password will be generated from such a thing.

A password that can only be generated by the person who has the key file

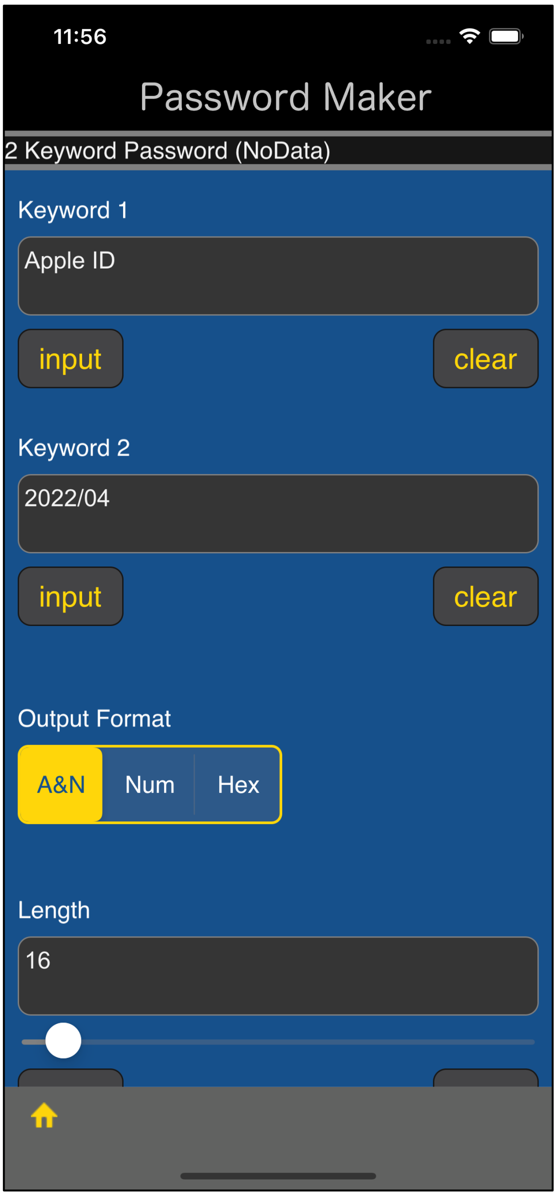
This allows you to generate such a password.

The "key file" is encrypted using an authenticated cipher (AES-256-GCM).

Therefore, it is possible to detect such a thing:

- The password used to decrypt the “key file” is incorrect.
- The “Key file” is corrupted. (Destroyed, tampered with.)

If this is detected, no password will be generated.



Password Maker

2 Keyword Password (NoData)

Keyword 1

Apple ID

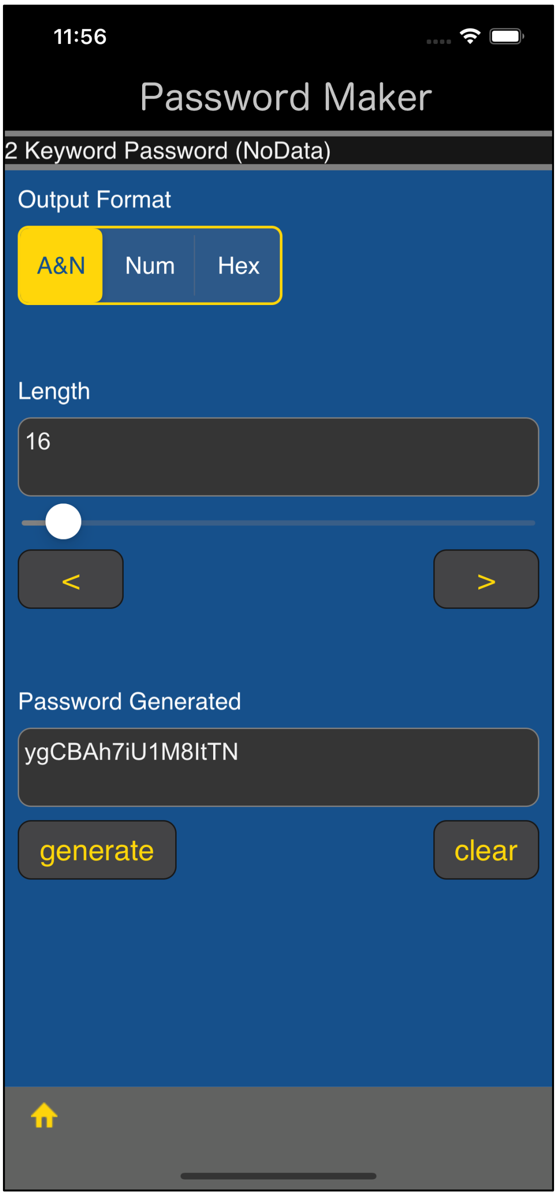
Keyword 2

2022/04

Output Format

Length

16



Password Maker

2 Keyword Password (NoData)

Output Format

Length

16

Password Generated

ygCBAh7iU1M8ItTN

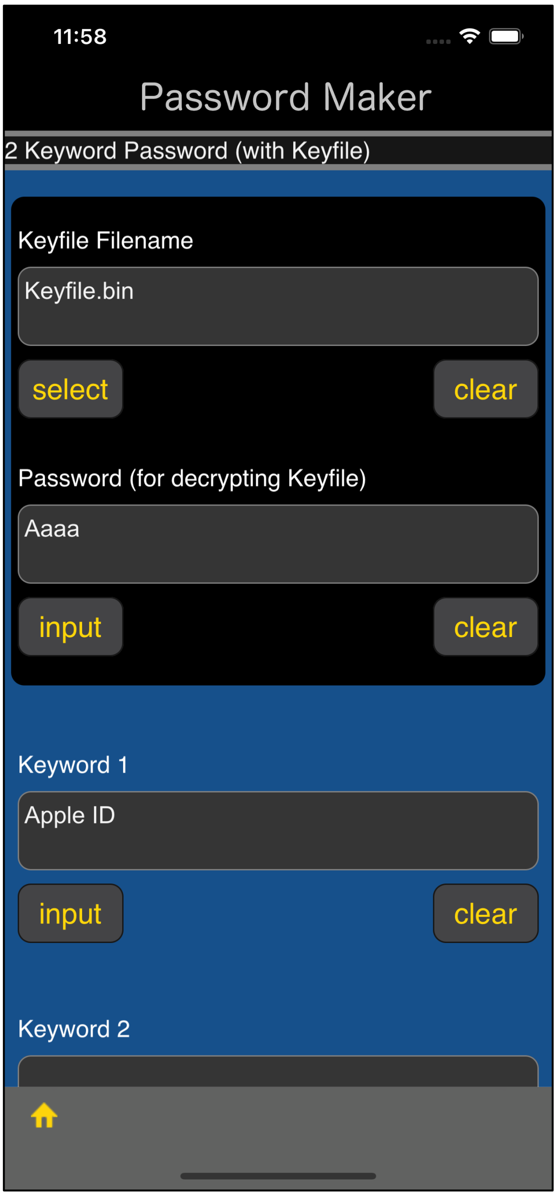
In this way,

Keyword 1	Apple ID
Keyword 2	2022/04
Length	16

from these, the password

ygCBAh7iU1M8ItTN

is generated.



11:58

Password Maker

2 Keyword Password (with Keyfile)

Keyfile Filename

Keyfile.bin

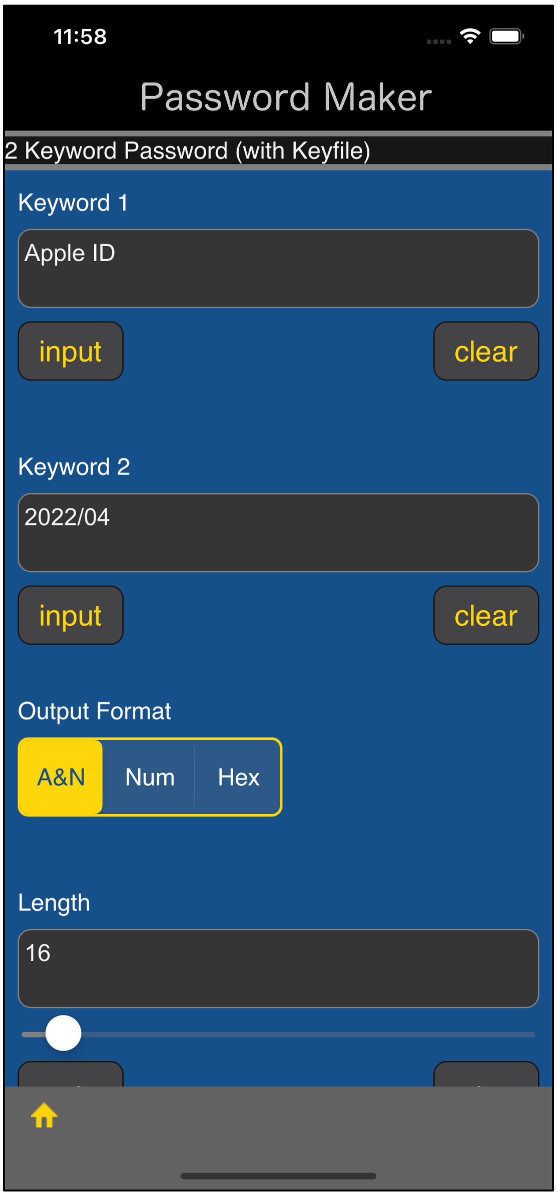
Password (for decrypting Keyfile)

Aaaa

Keyword 1

Apple ID

Keyword 2



11:58

Password Maker

2 Keyword Password (with Keyfile)

Keyword 1

Apple ID

Keyword 2

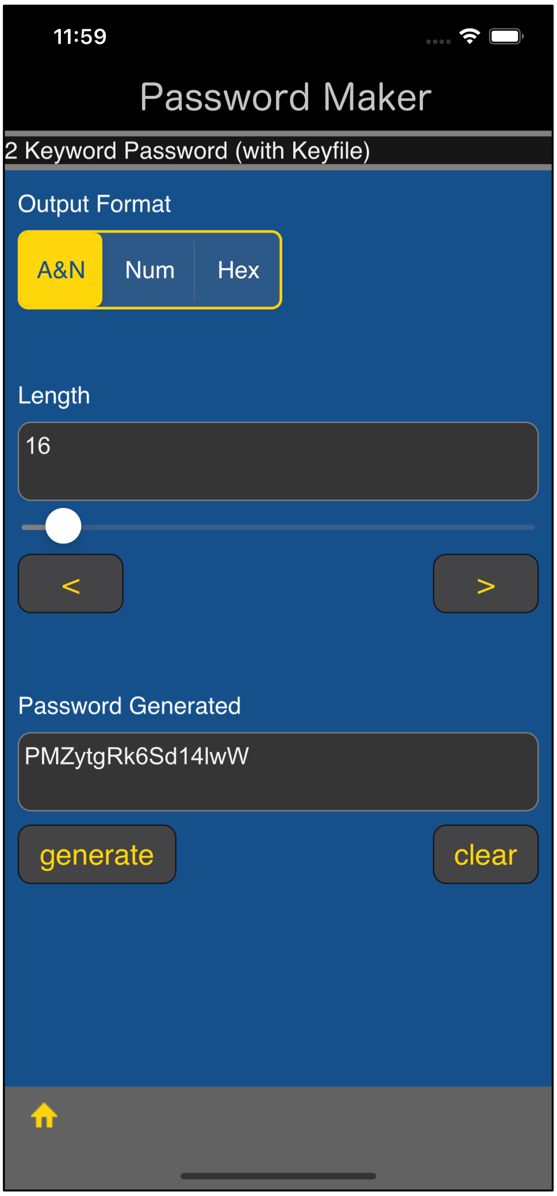
2022/04

Output Format

A&N | Num | Hex

Length

16



Output Format: [✓ A&N]

Length: 16

Password Generated: PMZytgRk6Sd14lwW

Also, in this way, when using a "key file",

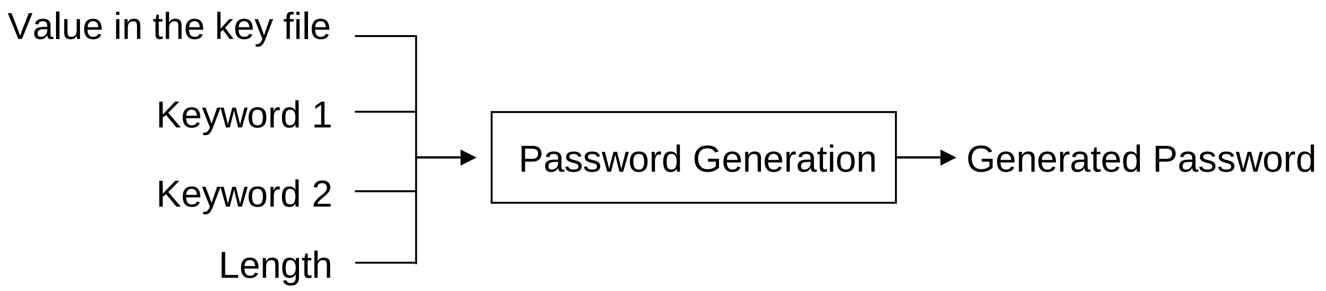
Keyfile	Keyfile.bin
Keyword 1	Apple ID
Keyword 2	2022/04
Length	16

from these, the password

PMZytgRk6Sd14lwW

is generated.

In password generation with “Key file”,



like this,

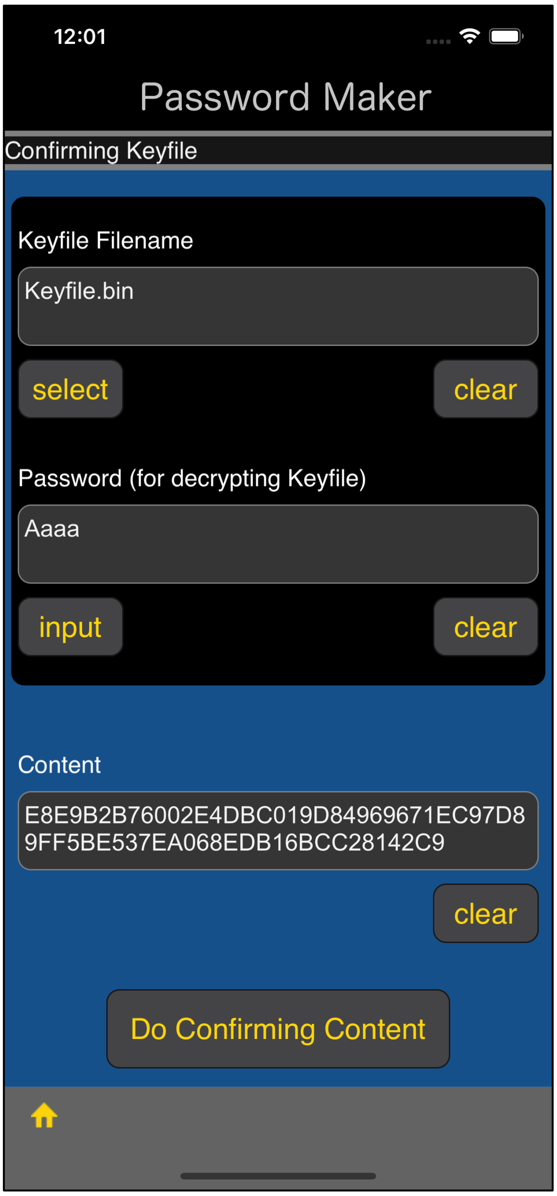
value in the key file

also participates in password generation.

As a result,

It is the person who have the key file can generate the password.

it will be like this.



Confirming Keyfile

Keyfile Filename

Keyfile.bin

Password (for decrypting Keyfile)

Aaaa

Content

E8E9B2B76002E4DBC019D84969671EC97D8
9FF5BE537EA068EDB16BCC28142C9

The value in the key file Keyfile.bin used in this example is

E8E9B2B76002E4DBC019D84969671EC9
7D89FF5BE537EA068EDB16BCC28142C9

this 32-byte value.

This 32-byte value also participates in the generation of the password PMZytgRk6Sd14lwW.

The password PMZytgRk6Sd14lwW can only be generated by people who have this 32-byte value.

The key file Keyfile.bin is created in such ways:

32-byte value

E8E9B2B76002E4DBC019D84969671EC9

7D89FF5BE537EA068EDB16BCC28142C9

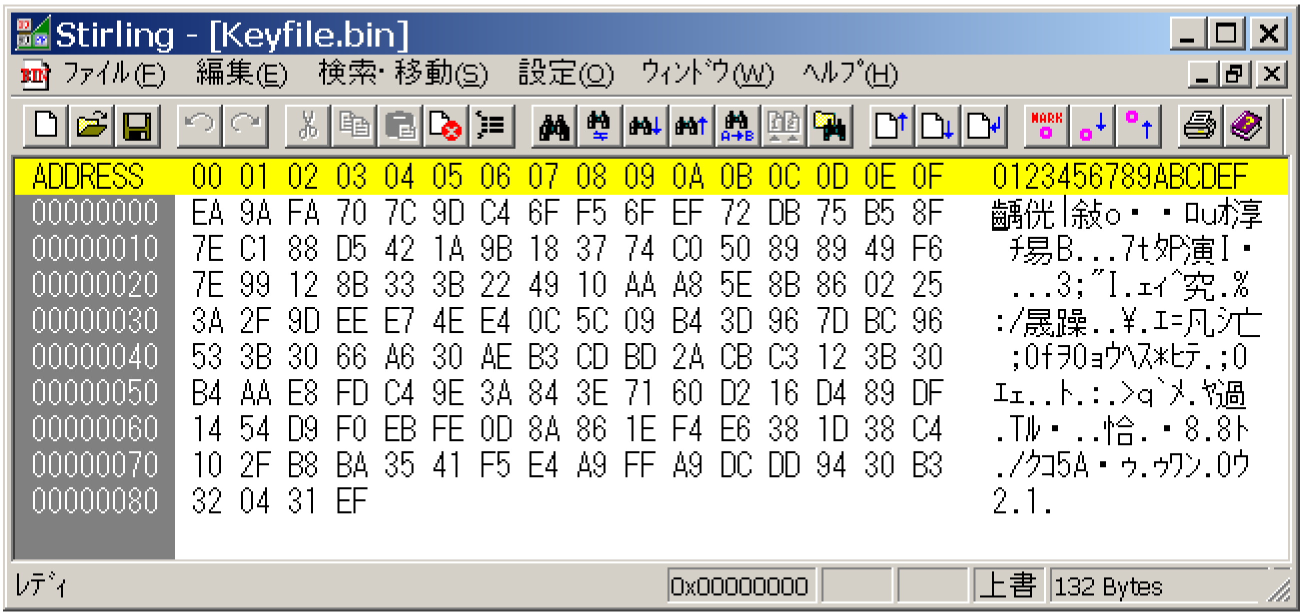
AES-256-Keywap is applied with this value and the wrapped value is encrypted with AES-256-GCM.

Cipher text created using AES-256-GCM can detect such things:

The decryption key is incorrect.

Cipher text has been tampered with.

If this is detected, password generation using a key file does not generate a password.



```
ADDRESS   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
00000000  EA 9A FA 70 7C 9D C4 6F F5 6F EF 72 DB 75 B5 8F   齲侊|敍o・ ・ﾛuｵ淳
00000010  7E C1 88 D5 42 1A 9B 18 37 74 C0 50 89 89 49 F6    ﾁ易B...7tﾀP演I・
00000020  7E 99 12 8B 33 3B 22 49 10 AA A8 5E 8B 86 02 25    ...3;"I.ｪｨ^究.%
00000030  3A 2F 9D EE E7 4E E4 0C 5C 09 B4 3D 96 7D BC 96   :/晨躁..¥.ｴ=凡ｼ亡
00000040  53 3B 30 66 A6 30 AE B3 CD BD 2A CB C3 12 3B 30    ;0fｦ0ｮｳﾍｽ*ﾋﾃ.;0
00000050  B4 AA E8 FD C4 9E 3A 84 3E 71 60 D2 16 D4 89 DF   ｴｪ..ﾄ.:.>q`ﾒ.ﾔ過
00000060  14 54 D9 F0 EB FE 0D 8A 86 1E F4 E6 38 1D 38 C4   .Tﾙ・..恰.・8.8ﾄ
00000070  10 2F B8 BA 35 41 F5 E4 A9 FF A9 DC DD 94 30 B3   ./ｸｺ5A・ｩ.ｩﾜﾝ.0ｳ
00000080  32 04 31 EF                                       2.1.
```

The contents of the key file "Keyfile.bin" itself are like this.

E8E9B2B76002E4DBC019D84969671EC9
7D89FF5BE537EA068EDB16BCC28142C9

Apply AES-Keywrap to the 32-byte value above.

Encrypt with AES-256-GCM after applying AES-Keywrap.

This is the ciphertext created in this way.

The contents cannot be retrieved by anyone who does not know the encryption key (decryption key) "Aaaa".

Also, if this ciphertext is inverted anywhere, even by one bit,
it can be detected that it has been tampered with.

If it is known that it has been tampered with, no password will be generated.