Proposal:

The feedback from TA focus on the volatile for the password. I searched the internet and find that it may a little difficult for me. However, I come up with an idea that avoids using external memory but can still implement function. I call it One-Time Password (OTP) Challenge-Response Electronic Lock. Instead of storing passwords in non-volatile memory, the design will generate a new random number for each authentication attempt and display it on LEDs. The user will then compute the correct response using a pre-defined algorithm and enter the response via buttons. If the response matches the expected value computed by the hardware, the lock will open.

In addition, I want to add more features:

1. If the user input a wrong password for three of times, the system will be locked for 60s. If the password is input again and again, the lock time will double each time. After it is unlocked, a new number will generate.
2. If there is a long time that the user doesn't input the password, for example 60s. The opt will regenerate and new number will be displayed on LEDs.
3. Use relative complex algorithm to compute the OPT. For example, CRC+ LFSR
4. To achieve similar effects of reset password, I want to implement several algorithms. If the user wants to change the algorithm, the correct password should be entered. And then the locker enters algorithm selection mode. The user can choose what kind of algorithm to be selected. All the modes and algorithm are included in the specification. Of course, the selected algorithm is only valid when the chips is on. If the chip is off and then on, the default algorithm is used.

Components:

1. Random number generator that uses shift registers and xor gate
2. Counter for locking system
3. 12 I/O distribution:

   6 LEDS: show the number. If entered password is correct, they are all on. If the password is incorrect, they will be shining for three times. If the system is locked, the odd one is on and the even one is off. It can generate and show 0-31 number. Based on 0-31 number, 0-10 bit password may be calculated. So the total possibility is 2046. It is impossible to try all of them in 60s.

   2 input buttons to enter binary password

   1 for clock

   1 for algorithm selection: when this button is pressed, the locker will ask the user to enter correct password. The LED will all on for 3s and all off. User can use 2 input buttons to enter the binary mode code. For example, 00, 01, 10, 11. At the same time, LED all display this mode. After pressing the confirm button. The mode selection is finished.

1 for confirm button

1 for clear current entered password

4. A FSM to control all the logic