

(Going back to authentication)

Recall three basic types of authentication:

- 1) Something you know (already covered: passwords)
- 2) Something you have
- 3) Something you are

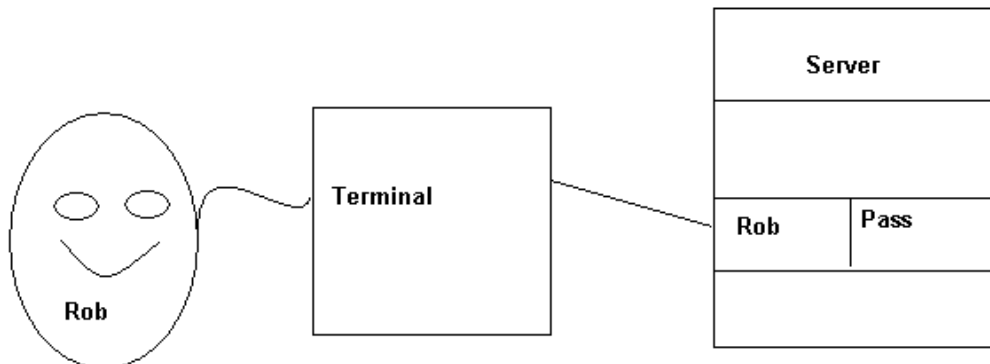
2 – Something you have

Here, this could be a key but in computer systems, we are usually talking about the following two things:

- Smart Cards
- Secure ID

Implementation of Secure ID

Secure Key:

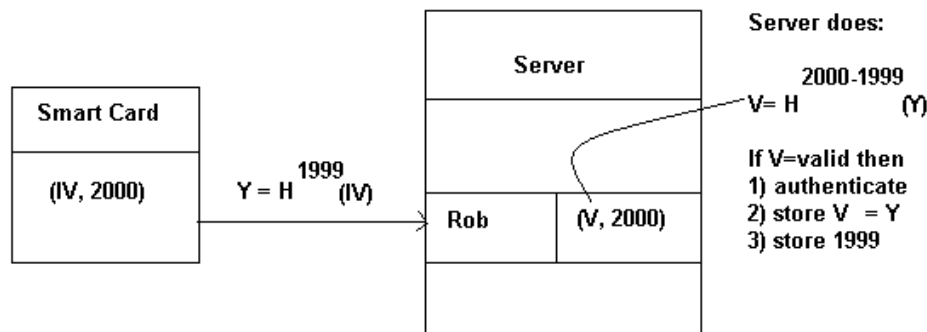


- It is clear that this scheme is not safe, but it can be made safer by using a challenge-response protocol
- Frequent password changes could also make this scheme safer, but they are a hassle for humans

Solution: Put pass on card – password is periodically changed on card, hence S/Key.

Method

- 1) Pick initial value (IV)
- 2) Compute $V = H^k(IV) = H(H(H(\dots H(IV)\dots)))$
- 3) Initially server stores (V, K), e.g. (V, 2000)
- 4) Initially store IV, K on smart card, e.g. (IV, 2000)
- 5) To login use card - on the first use: send $H^{1999}(IV)$
- 6) Card is automatically updated with new login parameters



Why is this secure?

- 1) We use a different password every login session
- 2) Attacker cannot get next password because rehashing won't work (note that our login sessions use the hashes from outer to inner, rehashing would be going in the other direction)
- 3) In this scheme, we must strictly enforce in-order-logins, so 2000, 1998, 1999 would be rejected because $H^{1999}(V)$ must come from a third party (most likely malicious).

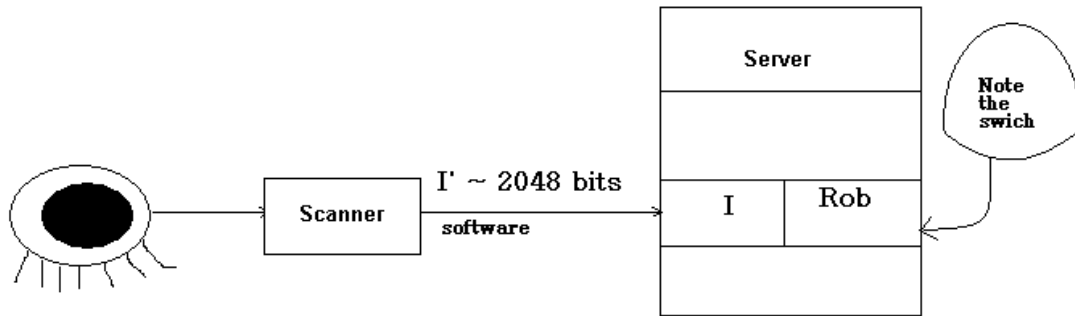
Why is this not secure?

It is vulnerable to a replay attack. If an attacker caches in one login attempts and then logs in using Y before the valid user's next attempt, the system has no way of knowing that this is a malicious user.

3- Something you are (Biometrics)

(Fingerprints, iris scans, voice print, face scans, palm scan, etc.)

Traditional Biometric Login



Valid if I' is close to I

Parameters

- If I' and I are from the same eye, then they should agree on about 1600 bits (about 20% difference)
- If I' and I are from two different eyes, then they agree on about 60% of the bits
- The problem that two different eyes agree on 80% is very small.

Problems

- Malicious scanner could store iris scan for later scan (note that a smart scanner could conduct a liveness test to make sure that the eye is a real one and not a picture)
- Usability depends on scanner – might need a higher acceptance threshold
- I stored in clear text on server

- Vulnerable to replay attack
- Iris is for life and supply is limited, so no frequent password changes
- Can't do challenge-response

Fixing Biometrics

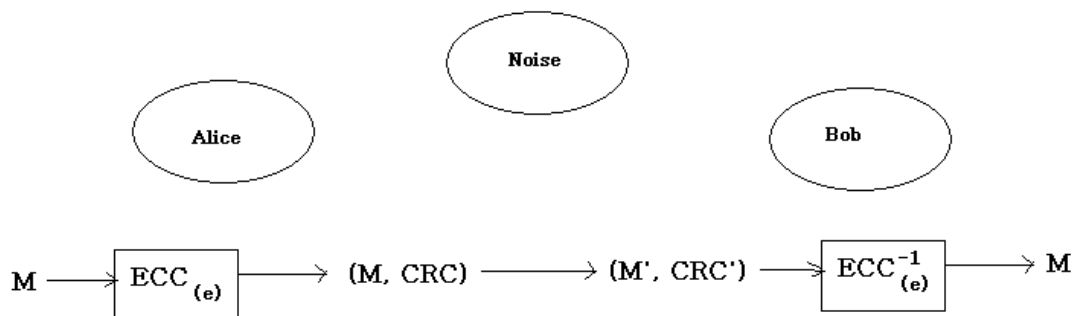
Step 1: Make biometric readings deterministic

$$I = b_1, \dots, b_{2048}$$

$$I' = b_1', \dots, b_{2048}'$$

The Hamming distance $d_h(I, I') \ll 400$

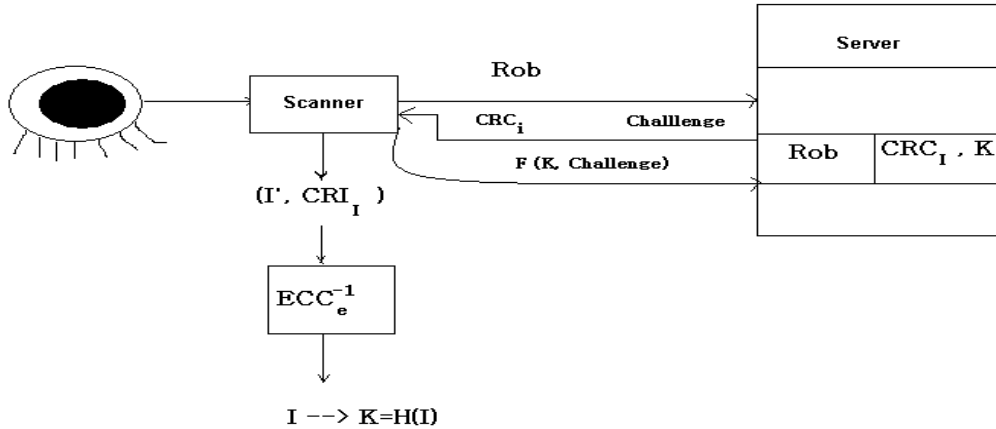
Error Correction



(CRC: Cyclic Redundancy Check)

In picture above, Bob recovers M if $d_h(M||CRC, M'|||CRC') \leq e$. In other words, the ECC can correct up to e -bit errors.

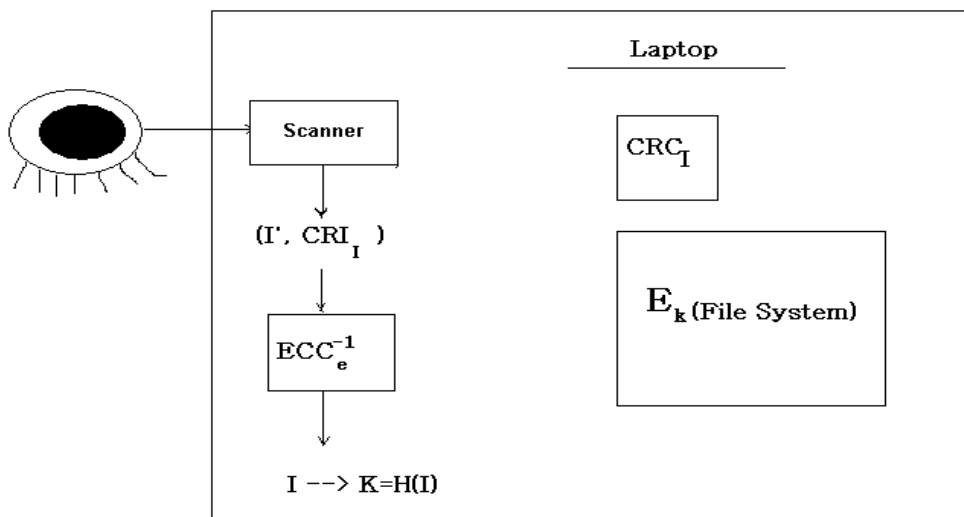
ECC Trick



Here I is not stored on the server. The scanner computes I by running I' and the CRC for I through the inverse of the error correction scheme. The key (K) is obtained by hashing I . Finally, the scanner responds to the challenge with $F(K, challenge)$.

Biometrics & Laptop (more secure laptops)

Sensitive information is sometimes stored on laptops, which can easily get stolen. Biometrics for laptop can protect the file system in a manner similar to the above scheme.



Now in order to get access to the file system, you must present a valid iris.

Step 2: Being able to change your key

- Don't: just add a key

Note that simply using a key in addition to the iris is not enough because the scanner, in the schemes above, only needs the CRC and the iris to find the key. In other words, this does not protect from a stolen iris.

- Do: Ensure that user brings a secondary input

(Something you have + Something you are)

