Change log:
1. Slide 43. More examples on blacklisting.
2. Slide 47. Add one way to remember password.

Topic 2: Authentication Credential

2.1. Overview

2.2 Password

- 2.2.1 Attack on bootstrapping
- 2.2.2 Attack on password reset
- 2.2.3 Searching password (Dictionary, guessing, exhaustive)
- 2.2.4 Stealing of password
- 2.2.5 Password Strength
- 2.2.6 Example on ATM

2.3 Biometric

2.4 Multi-factor authentication & Multi-step authentication

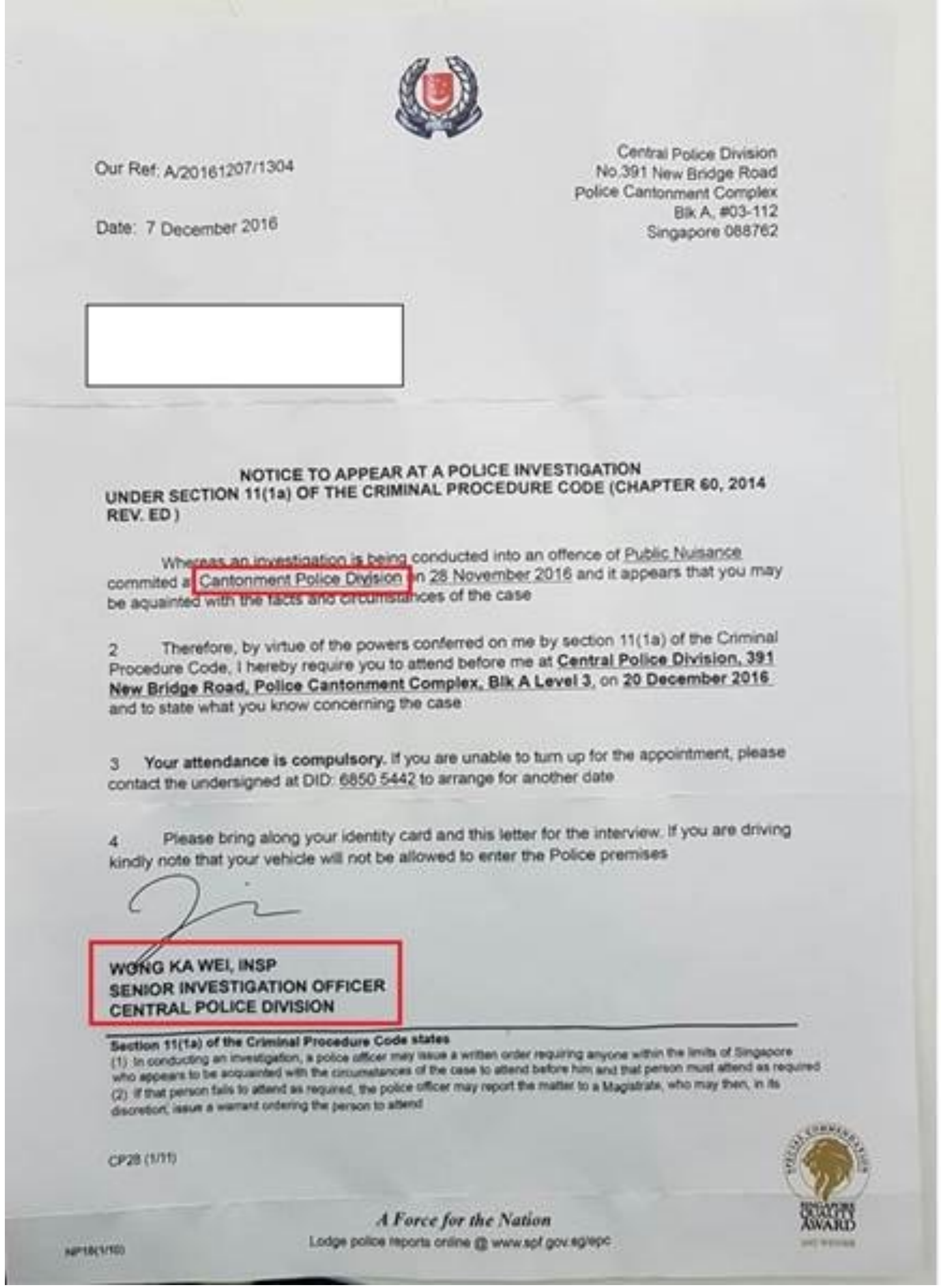
(Topic 2,3 are on “I”, Topic 4 combine the “I” and “C” to secure communication)

Summary and takeaways

- Data origin vs Communication Entity Authentication.
- Role of authentication credential. Something (data, device, etc) held by entity for authenticity verification. E.g. password, smart card, biometric. The entity who knows (what you know), holds (what you have), being (who are you) the credential is deemed to be authentic.
- Password strength
  - Online vs offline dictionary attack.
- Attacks on password.
  - Phishing. Bootstrap. Default password. *Phishing* is very effective.
- 2-factor vs 2-steps verification.
  - 2-factor/2-steps is better than single factor/step. E.g. online banking.
  - Compare different combinations.
  - Examples.

2.1 Overview

Is this letter Authentic?



Our Ref: A/20161207/1304

Date: 7 December 2016

Central Police Division
No.391 New Bridge Road
Police Cantonment Complex
Blk A, #03-112
Singapore 088762

[redacted]

NOTICE TO APPEAR AT A POLICE INVESTIGATION
UNDER SECTION 11(1a) OF THE CRIMINAL PROCEDURE CODE (CHAPTER 60, 2014 REV. ED)

Whereas an investigation is being conducted into an offence of Public Nuisance commited at Cantonment Police Division on 28 November 2016 and it appears that you may be aquainted with the facts and circumstances of the case

2 Therefore, by virtue of the powers conferred on me by section 11(1a) of the Criminal Procedure Code, I hereby require you to attend before me at **Central Police Division, 391 New Bridge Road, Police Cantonment Complex, Blk A Level 3**, on **20 December 2016** and to state what you know concerning the case

3 **Your attendance is compulsory.** If you are unable to turn up for the appointment, please contact the undersigned at DID: 6850 5442 to arrange for another date

4 Please bring along your identity card and this letter for the interview. If you are driving kindly note that your vehicle will not be allowed to enter the Police premises

WONG KA WEI, INSP
SENIOR INVESTIGATION OFFICER
CENTRAL POLICE DIVISION

Section 11(1a) of the Criminal Procedure Code states
(1) In conducting an investigation, a police officer may issue a written order requiring anyone within the limits of Singapore who appears to be acquainted with the circumstances of the case to attend before him and that person must attend as required
(2) If that person fails to attend as required, the police officer may report the matter to a Magistrate, who may then, in its discretion, issue a warrant ordering the person to attend

CP28 (1/11)

A Force for the Nation
Lodge police reports online @ www.spf.gov.sg/epc

NP18(1/10)

From: https://www.police.gov.sg/news-and-publications/media-releases/20161217_others_advisory_spf_letters,

December 16, 2016

Authentication

Authentication: The process of assuring that the communicating entity, or origin of a piece of information, is the one that it claims to be.

Authentic (adjective): the claimed origin/entity is assured by supporting evidences.

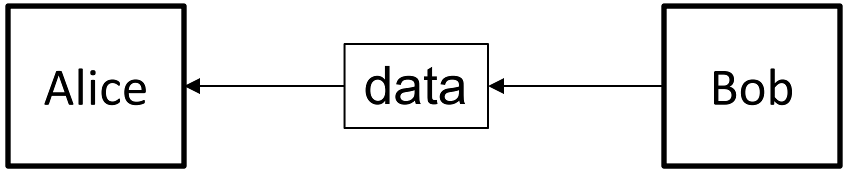
Authenticity: condition of being authentic.

Authenticity implies integrity.

(Many documents use the term “integrity” to mean authenticity. For instance, authenticity is grouped under “I” in “C-I-A”. However, some documents argue that we can’t compare authenticity with integrity. When reading a document, pay attentions to the context and the applications involved.)

Two types of authentication studied in this course

Data Origin (Topic 3)

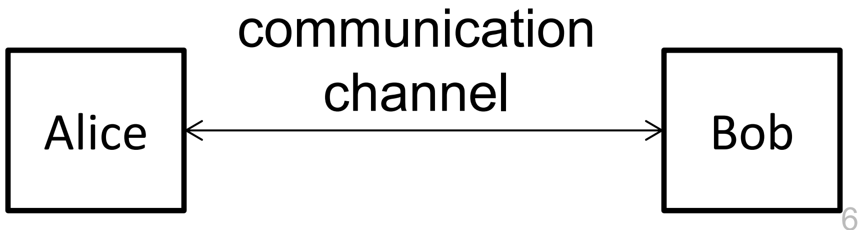


Is a piece of data generated by an authentic entity?

Eg.

- Alice downloaded an apps, say SingPass from some apps store. Is the app authentic?
- Is an email authentic?

Communicating Entity (Topic 4)



Is the entity interacting with the verifier an authentic entity?

Eg.

- Alice received a phone call, which claimed to be from the police department and asked for information regarding her brother. Authentic?
- Alice logged-in to “Canvas”. Alice wondered, was the server indeed the authentic “Canvas”? Conversely, the Canvas’s server might wonder whether the entity logged in is the authentic “Alice”?
- Alice tried to connect to wifi using her phone while in NUH’s bus-stop. Among the available wifi Network Name (SSID), an item “NUS” is listed. Alice connected and keyed in her userid and password. Is that wifi access point authentic?

Authentication process

- For data-origin authentication, one way is to use crypto primitives, such as Signature or MAC (Message Authentication Code). (Topic 3)

- For communication authentication, we would need some “authentication protocol” which employ the above crypto primitives. (Topic 4)

Credential

- Some information bound to the owner is required for authentication. If owner can convince the verifier that it has access to that info, it is deemed to be authentic. That info is called *Credential.*

- Password is an authentication credential. Other examples to be mentioned in multi-factor authentication.

2.2 Credential: Password

- Password system
- Attack

Password system

Passwords vs key in encryption.

Similarity: Secrets shared between two entities.

Differences: Passwords are generated by human and can be remembered by human. Secret keys are binary sequence that are infeasible to be remembered by human.

- Hence “entropy” of passwords is significantly lower than entropy of key.
- It is possible to generate keys from passwords (Key Derivation Function KDF). But the entropy of the generated key remain the same as the original passwords.

Password system

(1) Bootstrapping.

A user and the server establish a common password. The server keeps a ***password file*** keeping the *identity (aka userid, username)* and the corresponding *password*.

The identity information is not considered to be secret, although it is not advisable to voluntarily make it public. E.g. of identity: username in computer system, bank account number,   customer id, etc.

(2) Authentication.

The server authenticates an entity. An entity who can convince the server that it knows the password, is deemed to be authentic.   Note that this is a special case of communication entity authentication.

(3) Password reset.

There are many reasons to reset password. E.g user forgets the password. A password policy could require regular changes of password.

(1) Bootstrapping

- The password is to be established during bootstrapping.

- This can be done by

  1) The server/user chooses a password and sends it to the user/server through another communication channel.

  2) Default password.

- Describe some bootstrapping mechanisms that you have encountered.
- optional: What is WPS (Wi-Fi Protected Setup)? Describe an attack scenario where WPS could fail (Describe attacker's capability. Attacker's goal is to either know the password, or cause the device to use a password chosen by the attacker). See https://www.digitalcitizen.life/simple-questions-what-wps-wi-fi-protected-setup

(2) Authentication using password

- Authentication Protocol. (communicating entity)

  User → Server: My name is ***Alice***

  Server → User : What is your password

  User → Server: ***OpenSesame***

  Server verifies whether password is correct and takes corresponding subsequent actions.

- Authentication can also be carried out without interactions. (Data Origin)

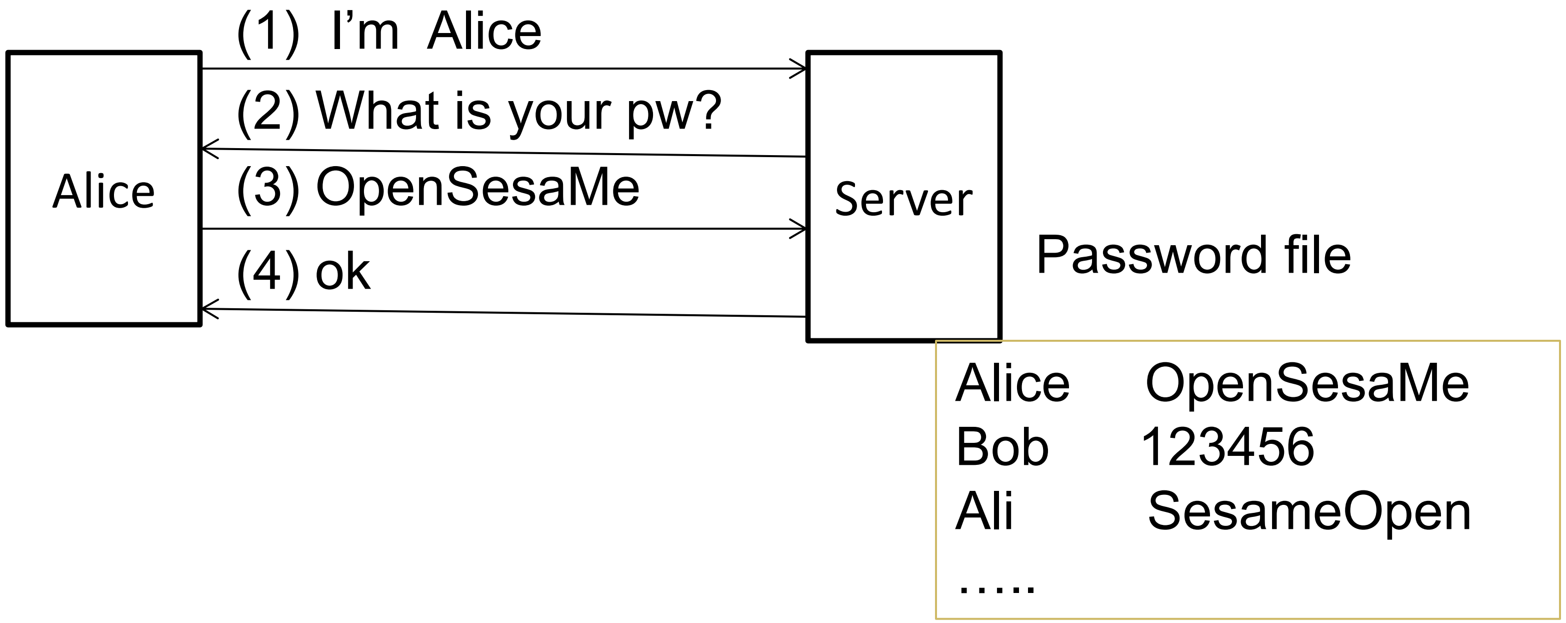
User sent the following sms to a server.

```
“ID:alice@nus.edu.sg PW:OpenSesame INSTRUCT:Unsubscribe.”
```

User visited a website

```
“https://pizzashop.com.sg/online?user=alice&pw=OpenSeasame&type=cheese”
```

Password file



Password file

Alice	OpenSesaMe
Bob	123456
Ali	SesameOpen
…..	

Replay attack

- If the attacker has the capability to ***sniff*** the communication channel, then the previous protocol is not secure. It is is subjected to the simple "***replay attack***": information sniffed from the communicated channel by an attacker can be replayed to impersonate the user.

It is possible to have an authentication where information sniffed can't be used to impersonate the user. (Topic 4)

Terminologies. What are "***Sniff***", "***Spoof"***?

(3) Password reset

- Resetting password is trickly. Only the authentic entity can reset the password. How to verify that the entity is authentic? What if the user forgets the password?

- We need to authenticate the entity before allowing the entity to change password.
  1. Anyone who know the old password.
  2. Need another credential (other than the old password) to authenticate.

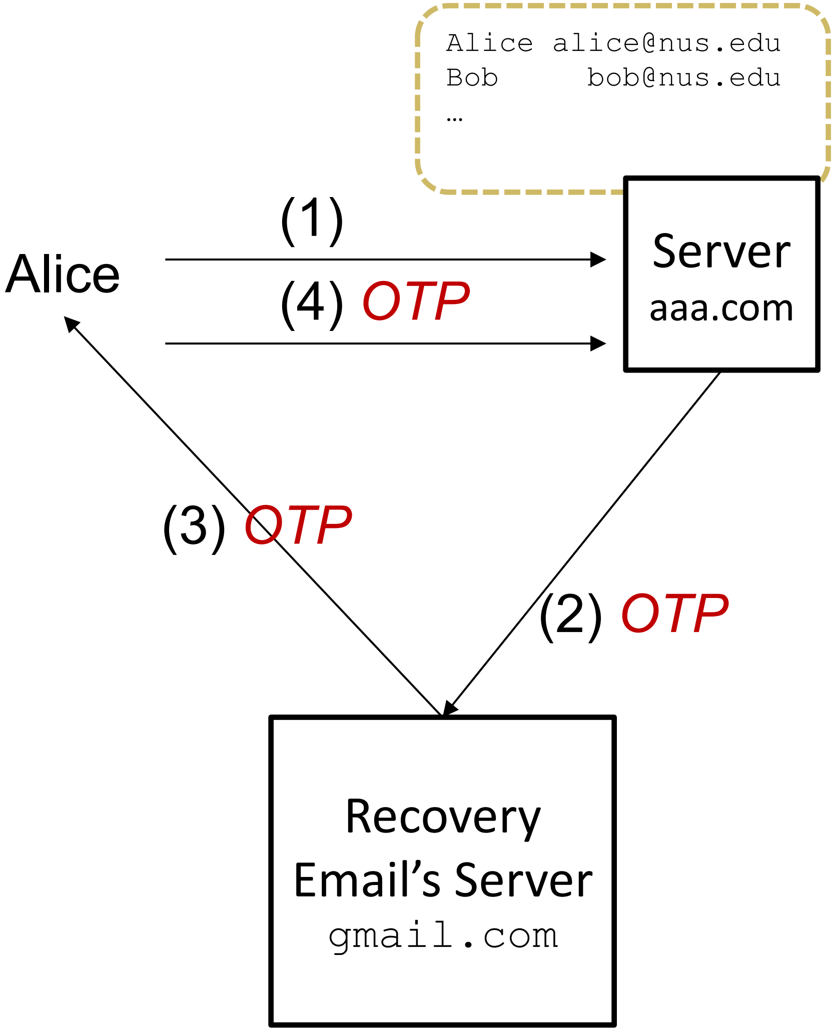
- One method is to have the person goes to the helpdesk (e.g. NUS IT) to be authenticated for password reset. However, having human involved in resetting password is expensive and incovenient. Many systems provide “self-help password reset”.

Password reset using recovery email's account (using another credential)

Many systems link an account to a recovery email. If a user forgets the password, the password is reset in the following way:

1) Alice → Server: *"I'm Alice. I want to reset."*
2) Server → Recovery: Server sends an email to the recovery account containing *"https://aaa.com/reset?OTP=13ac92DadvSEga5"*

   Note that the url of the link contains an OTP (one-time-password). Essentially, the system sends an OTP to the user. Some call this "identifier". In this course, we call it OTP.
3) Recovery → Alice: Alice reads the email.
4) Alice → Server: Alice clicks on the url and enters new password
5) Server checks whether the OTP is correct. If so, reset.



Here, ownership of the email address proves that the entity is authentic. The ownership of the email address is the "credential". This is "What-you-have" in 2-factor (to be covered later).

Password reset using Security Question.

One self-help method is security questions. This mechanism was very common and, fortunately, less common now.

E.g. of security questions.

- What is your first car?
- What is your favorite cake?

Security questions facilitate self-help password reset and improve “usability”. However, it weakens security. Answers to the questions essentially are another password. Very often the “entropy” of the answers is lower than the password.

Attack on passwords

- Attack the bootstrapping.
- Attack the password reset process.
- Search for the correct password (two modes: online vs offline)
  - Guessing
  - Dictionary attacks
  - Exhaustive attacks

- Steal the password:
  - Eavesdropping: sniff the network, key-logger.
  - Phishing
  - Spear-phishing
  - Spoofing login screen
  - Password Caching
  - Insider attacks

- 2.2.1 Attack on Bootstrapping

Default Password

Attacker may intercept the password during bootstrapping. For example, if the password is sent through postal mail, an attacker could steal the mail to get the password.

Attacker uses the "default" passwords. There are many reported incidents on this simple attack.

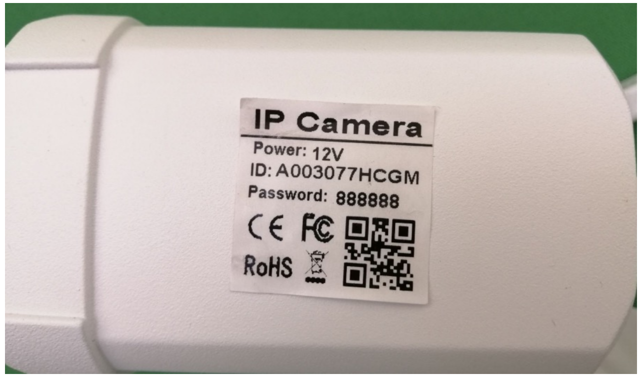
(for e.g. IP camera, Wifi router)

see http://www.pcworld.com/article/2033821/widely-used-wireless-ip-cameras-open-to-hijacking-over-the-internet-researchers-say.html

Read (Mirari attack, Sep 2016)

http://www.computerworld.com/article/3134097/security/chinese-firm-admits-its-hacked-products-were-behind-fridays-ddos-attack.html

Default Password



Power: 12V
ID: A003077HCGM
Password: 888888

https://www.sricam.com/srihome/article/id/5f7f120ac0b641a1aa7e3b051fe0026c.html

- There are challenges on usability and logistic in replacing default password with different passwords. E.g. in the above sticker, each device has a different password. Hence, in practice, most devices are not shipped with different passwords.

- Mitigation: Require the user to change password after first login.

Example of vulnerability in bootstrap/reset with recovery email Zoom’s account hijacking

- Here is an example of a vulnerability in an implementation of recovery email. The attack worked due to a bug.

- Read “Zoom Flaw allowed account hijacking” in https://www.tomsguide.com/news/zoom-security-privacy-woes

- The above attack bootstrapping (sign up process), although potentially, Zoom could employ the same mechanism in password reset.

Excerpt:

“Zoom flaw allowed account hijacking

A Kurdish security researcher said Zoom paid him a bug bounty -- a reward for finding a serious flaw -- for finding how to hijack a Zoom account if the account holder's email address was known or guessed.

The researcher, who calls himself "s3c" but whose real name may be Yusuf Abdulla, said if he tried to log into Zoom with a Facebook account, Zoom would ask for the email address associated with that Facebook account. Then Zoom would open a new webpage notifying him that a confirmation email message had been sent to that email address.

The URL of the notification webpage would have a unique identification tag in the address bar. As an example that's much shorter than the real thing, let's say it's "zoom.com/signup/123456XYZ".

When s3c received and opened the confirmation email message sent by Zoom, he clicked on the confirmation button in the body of the message. This took him to yet another webpage that confirmed his email address was now associated with a new account. So far, so good.

But then s3c noticed that the unique identification tag in the Zoom confirmation webpage's URL was identical to the first ID tag. Let's use the example "zoom.com/confirmation/123456XYZ".

The matching ID tags, one used before confirmation and the other after confirmation, meant that s3c could have avoided receiving the confirmation email, and clicking on the confirmation button, altogether.

In fact, he could have entered ANY email address -- yours, mine or billgates@gmail.com -- into the original signup form. Then he could have copied the ID tag from the resulting Zoom notification page and pasted the ID tag into an already existing Zoom account-confirmation page.

Boom, he'd have access to any Zoom account created using the targeted email address.

"Even if you already linked your account with a Facebook account Zoom automatically unlink it and link it with the attacker Facebook account," s3c wrote in his imperfect English.

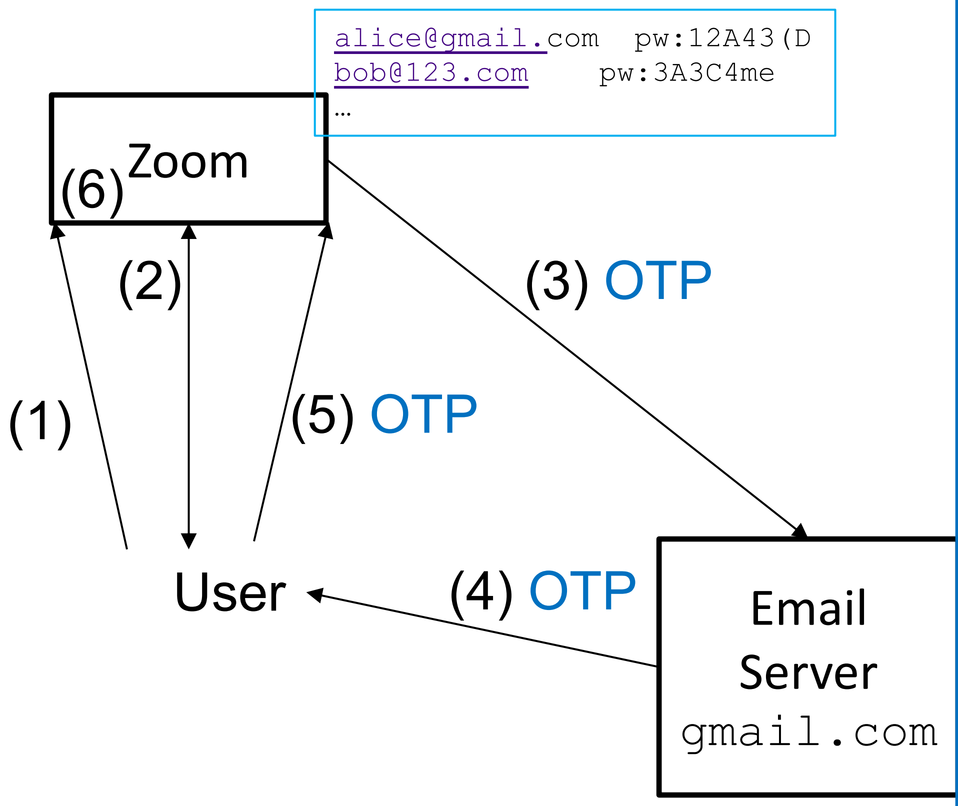
And because Zoom lets anyone using a company email address view all other users signed up with the same email domain, e.g. "company.com", s3c could have leveraged this method to steal ALL of a given company's Zoom accounts.

"So if an attacker create an account with email address attacker@companyname.com and verify it with this bug," s3c wrote, "the attacker can view all emails that created with *@companyname.com in Zoom app in Company contacts so that means the attacker can hack all accounts of the company."

Zoom is fortunate that s3c is one of the good guys and didn't disclose this flaw publicly before Zoom could fix it. But it's such a simple flaw that it's hard to imagine no one else noticed it before.

STATUS: Fixed, thank God.”

Normal usage



1) “I want to signup. My ID is alice@nus.edu ”
2) Ok. Choose a pw (notification website)
3) OTP. (send confirmation email)
4) OTP. (read confirmation email)
5) OTP. (click confirmation email)
6) Check whether OTP in (3) matches OTP in step (5). if so, carry on.

Excerpt:

“Zoom flaw allowed account hijacking

A Kurdish security researcher said Zoom paid him a bug bounty -- a reward for finding a serious flaw -- for finding how to hijack a Zoom account if the account holder's email address was known or guessed.

The researcher, who calls himself "s3c" but whose real name may be Yusuf Abdulla, said if he tried to log into Zoom with a Facebook account, Zoom would ask for the email address associated with that Facebook account. Then Zoom would open a new webpage notifying him that a confirmation email message had been sent to that email address.

The URL of the notification webpage would have a unique identification tag in the address bar. As an example that's much shorter than the real thing, let's say it's "zoom.com/signup/123456XYZ".

When s3c received and opened the confirmation email message sent by Zoom, he clicked on the confirmation button in the body of the message. This took him to yet another webpage that confirmed his email address was now associated with a new account. So far, so good.

But then s3c noticed that the unique identification tag in the Zoom confirmation webpage's URL was identical to the first ID tag. Let's use the example "zoom.com/confirmation/123456XYZ".

The matching ID tags, one used before confirmation and the other after confirmation, meant that s3c could have avoided receiving the confirmation email, and clicking on the confirmation button, altogether.

In fact, he could have entered ANY email address -- yours, mine or billgates@gmail.com -- into the original signup form. Then he could have copied the ID tag from the resulting Zoom notification page and pasted the ID tag into an already existing Zoom account-confirmation page.

Boom, he'd have access to any Zoom account created using the targeted email address.

"Even if you already linked your account with a Facebook account Zoom automatically unlink it and link it with the attacker Facebook account," s3c wrote in his imperfect English.

And because Zoom lets anyone using a company email address view all other users signed up with the same email domain, e.g. "company.com", s3c could have leveraged this method to steal ALL of a given company's Zoom accounts.

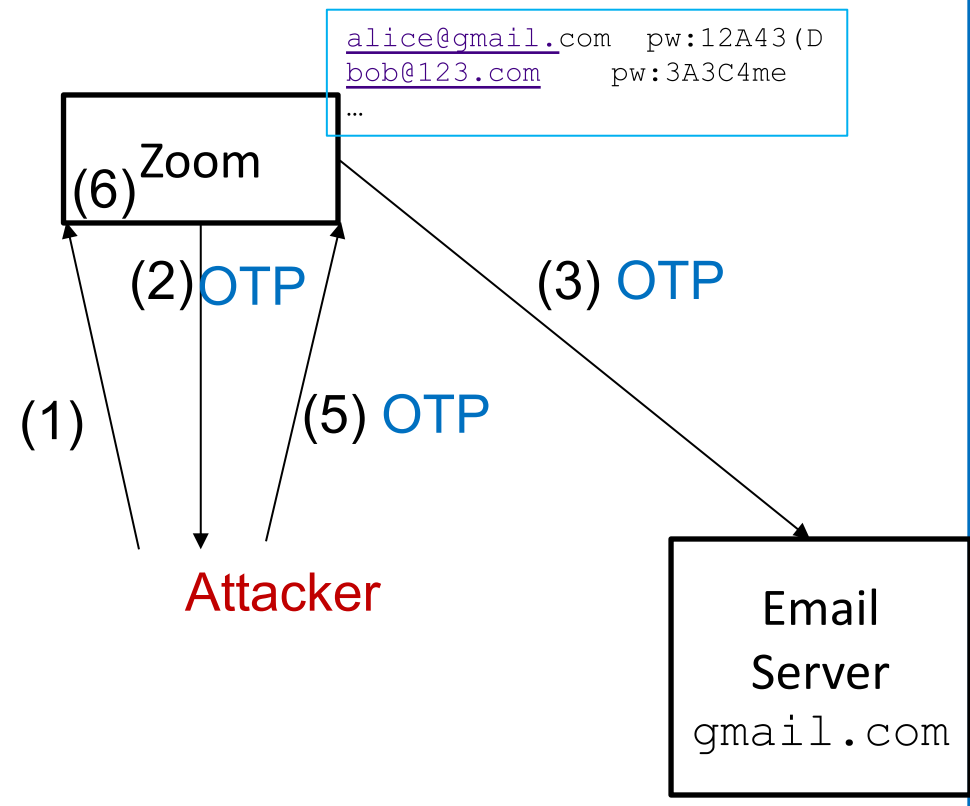
"So if an attacker create an account with email address attacker@companyname.com and verify it with this bug," s3c wrote, "the attacker can view all emails that created with *@companyname.com in Zoom app in Company contacts so that means the attacker can hack all accounts of the company."

Zoom is fortunate that s3c is one of the good guys and didn't disclose this flaw publicly before Zoom could fix it. But it's such a simple flaw that it's hard to imagine no one else noticed it before.

STATUS: Fixed, thank God.”

This is a rather silly mistake.

Under Attack



1) “I want to signup. My ID is alice@nus.edu ”
2) Ok. Choose a pw. (notification website contains OTP)
3) OTP. (send confirmation email)
~~4) OTP. (read confirmation email)~~ attacker skips
5) OTP. ~~(click confirmation email)~~ directly visits
6) Check whether OTP in (3) matches OTP in step (5).

- 2.2.2 Attack on password reset

Security Questions

- The mechanism of security questions weakens the password system [Rabkin2008].
- Fortunately, it is less common now.

[Rabkin2008] Ariel Rabkin, *Personal knowledge questions for fallback authentication: security questions in the era of Facebook*, Usable privacy and security 2008.

Social engineering + password reset

Someone told me this attack. Unfortunately, I can't verify the info, and hence redacted and replaced name of the platform with XXXXX.

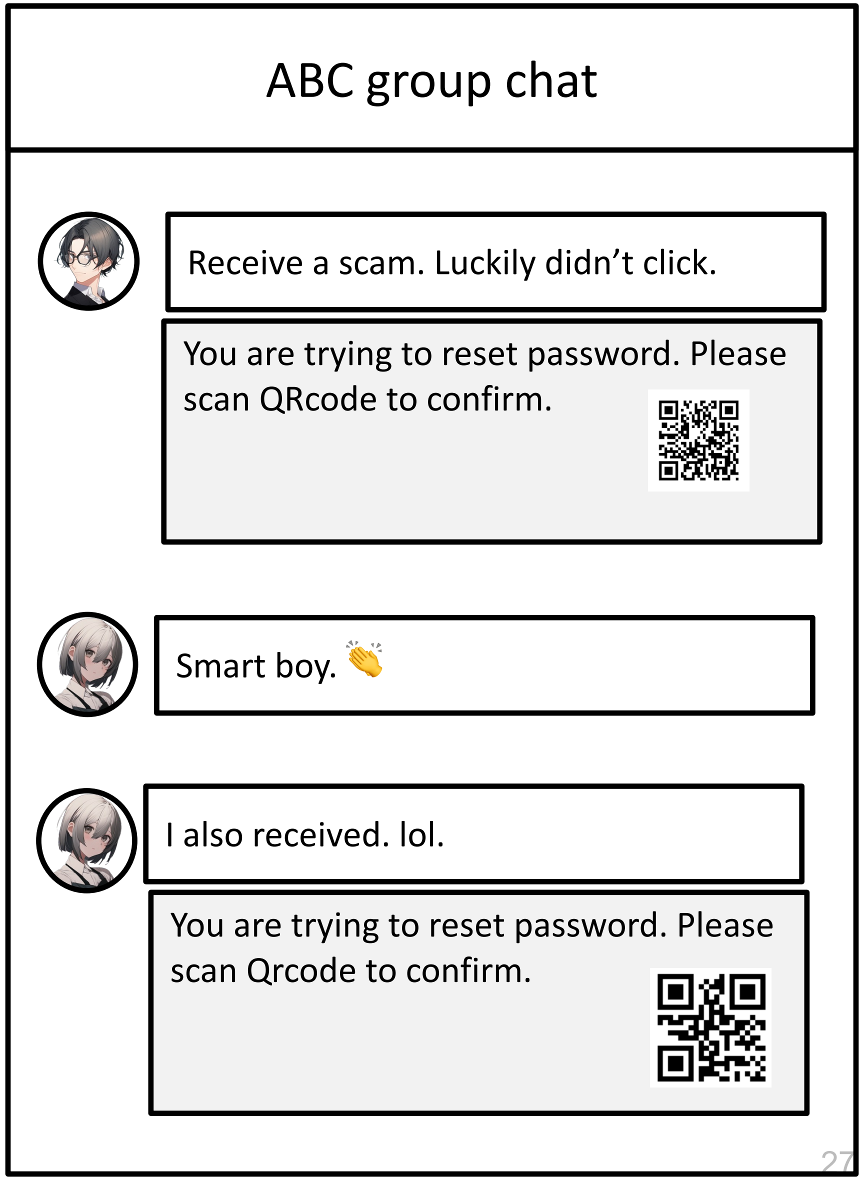
- Suppose an attacker had compromised Bob's account of a social media platform XXXXX (i.e. attacker knew Bob's pw). Bob was in a private chat-group ABC in XXXXX.

- The attacker (using Bob's account) submitted a post to the group ABC, mentioning that he received a phishing email who claimed to be from XXXXX, and the email showed an QR code. The post also mentioned that this is likely a phishing attempt and included the email with the QR code with some ha, ha and lol. Other members in the group agreed and clapped.

- Attacker went to XXXXX's site and initiated password reset of ABC members' accounts, prompting XXXXX to automatically send confirmation emails to the members. The emails contained some info embedded in QR code and asked the members to scan the QR code using XXXXX's apps.

- A member in ABC replied to Bob's post "I also received this" and posted the QR code.

- Attacker used the QR code to confirm the password change.



ABC group chat

Receive a scam. Luckily didn't click.

You are trying to reset password. Please scan QRcode to confirm.

Smart boy. 👏

I also received. lol.

You are trying to reset password. Please scan Qrcode to confirm.

- 2.2.3 Searching for the Password

Dictionary attacks

- An attacker could test whether a password is correct by feeding it to the login session. With this ability to probe, the attacker can search for the correct password.
- ***Dictionary attack*** tests the passwords using a “dictionary”. The dictionary could contain words from English dictionary, known compromised passwords etc.
- Dictionary attack also tests combinations of words in the dictionary. For e.g. it tries all combinations of 2 words from the dictionary; try all possible capitalizations of letters in each word; substituting “a” by “@”, etc
- There are tools for dictionary attack.

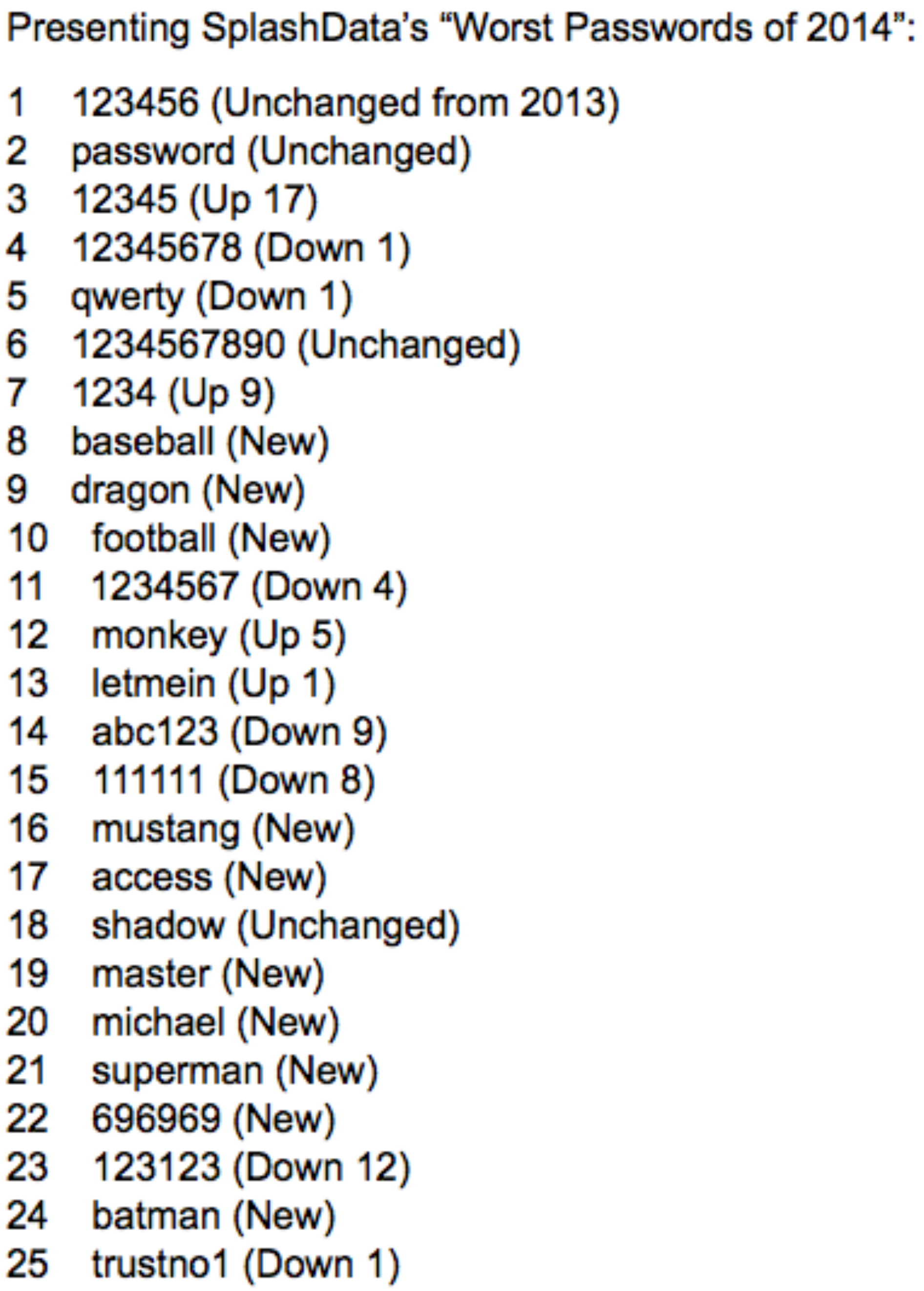
- Next slide shows list of “2014 worst password” reported by SplashData

http://www.prweb.com/releases/2015/01/prweb12456779.htm

- **see** the 2023 list.

https://sea.mashable.com/tech/28930/worst-passwords-of-2023-include-some-familiar-favorites-see-the-list

Hint on Assignment: One question on this.



Presenting SplashData's "Worst Passwords of 2014":

1 123456 (Unchanged from 2013)
2 password (Unchanged)
3 12345 (Up 17)
4 12345678 (Down 1)
5 qwerty (Down 1)
6 1234567890 (Unchanged)
7 1234 (Up 9)
8 baseball (New)
9 dragon (New)
10 football (New)
11 1234567 (Down 4)
12 monkey (Up 5)
13 letmein (Up 1)
14 abc123 (Down 9)
15 111111 (Down 8)
16 mustang (New)
17 access (New)
18 shadow (Unchanged)
19 master (New)
20 michael (New)
21 superman (New)
22 696969 (New)
23 123123 (Down 12)
24 batman (New)
25 trustno1 (Down 1)

Dictionary attacks

Two scenarios in dictionary attacks:

- ***Online dictionary*** *attack*: To test a password, attacker must interact with the authentication system.
  - Attacker obtained a list of 1000 valid nusnet id. The attacker wants to find the password for some of them. The attacker writes an automated script that attempt to login to Canvas using guessed passwords for each of these 1000 valid nusnet id.

- ***Offline dictionary*** *attack*: There are two phases.
  1. The attacker obtains some information ***D*** about the password, possibly by sniffing the login session of an authentic user, or by interacting with the server. (we will illustrate this when covering authentication protocol. Now, let's just assume that somehow the "hashed" password is sent over, and the attacker manage to obtain the hash)
  2. Next, the attacker carries out dictionary attack using ***D*** without interacting with the system.

  (e.g. compare with the hashed words in dictionary. "hash" to be covered next week.)

Example of offline dictionary attack:

1. Attacker has an AES encrypted pdf file. The AES key is derived from a password. The attacker wants to find the password.
2. In some password authentication protocols, a "hash" of the password is sent in clear. The attacker first obtained the hash by eavesdropping a valid login session. Next, the attacker went offline and searched for the password. WPA2 personal is vulnerable to this form of offline dictionary attack.
3. The password file contains hashes of password. Attacker somehow obtained a password file, and then carryout offline dictionary attack.
4. WPA2-personal employs a protocol whereby offline dictionary attack can be carried out. (*more on this later*)

Guessing the password from social information

- The attacker gathers some social information about the user, and infer the password, e.g. mobile phone number.

- This can be done by constructing a dictionary using words extracted from the social media.

Hint: One assignment question on this.

2.2.4 Stealing the password

1. Sniffing

- ***Shoulder surfing***: This is the look-over-the-shoulder attack.

- ***Sniffing*** the communication: Uncommon now. Some systems simply send the password over the public network in clear (i.e. not encrypted). E.g. FTP, Telnet, HTTP. (secure version sftp, ssh)

Other methods:

- ***Sniff*** wireless keyboards that employ insecure encryption method.

see http://arstechnica.com/security/2015/01/meet-keysweeper-the-10-usb-charger-that-steals-ms-keyboard-strokes/

- Using sound made by keyboard. (Using information extracted from physical world is known as ***side-channel attack.***).

(L. Zhuang, F. Zhou, J.D. Tygar, Keyboard Acoustic Emanations Revisited, 2005.
http://www.cs.berkeley.edu/~tygar/papers/Keyboard_Acoustic_Emanations_Revisited/ccs.pdf)

Viruses, Keylogger.

A key-logger captures the keystrokes and sends the information back to the attacker.

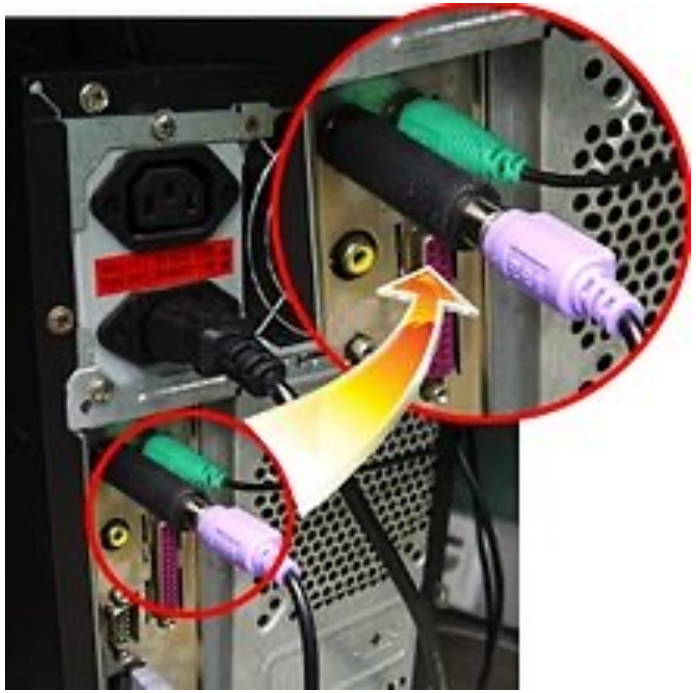
- (software) Some computer viruses are designed as a ***key-logger****.*

- (hardware) Hardware key-logger: the image in the next slide is self-explanatory.

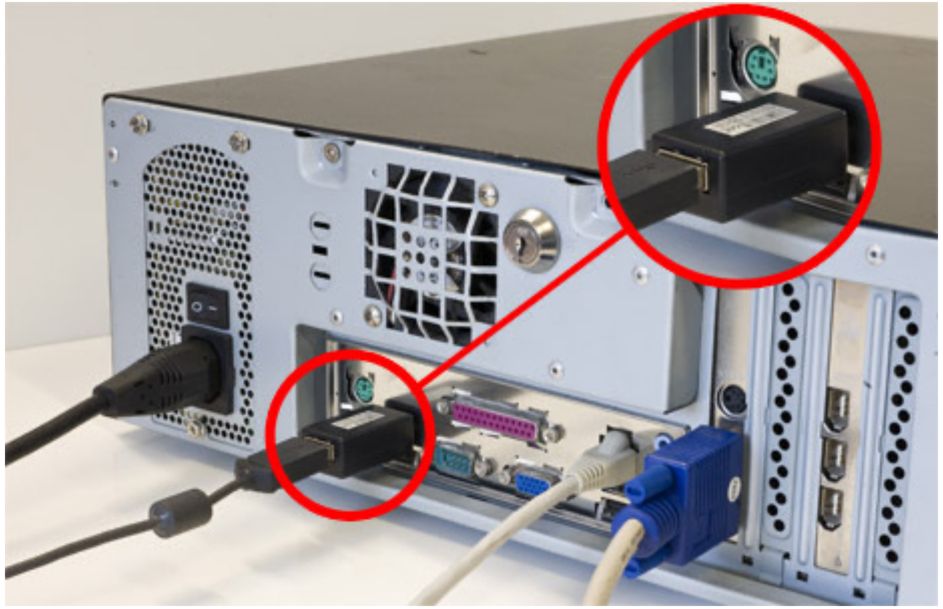
see “Hardware-based keyloggers” in

http://en.wikipedia.org/wiki/Keystroke_logging

The software key-logger needs to send the captured data back to the attacker. This should be done in a stealthy way, i.e. via a “covert channel”.



http://en.wikipedia.org/wiki/Keystroke_logging

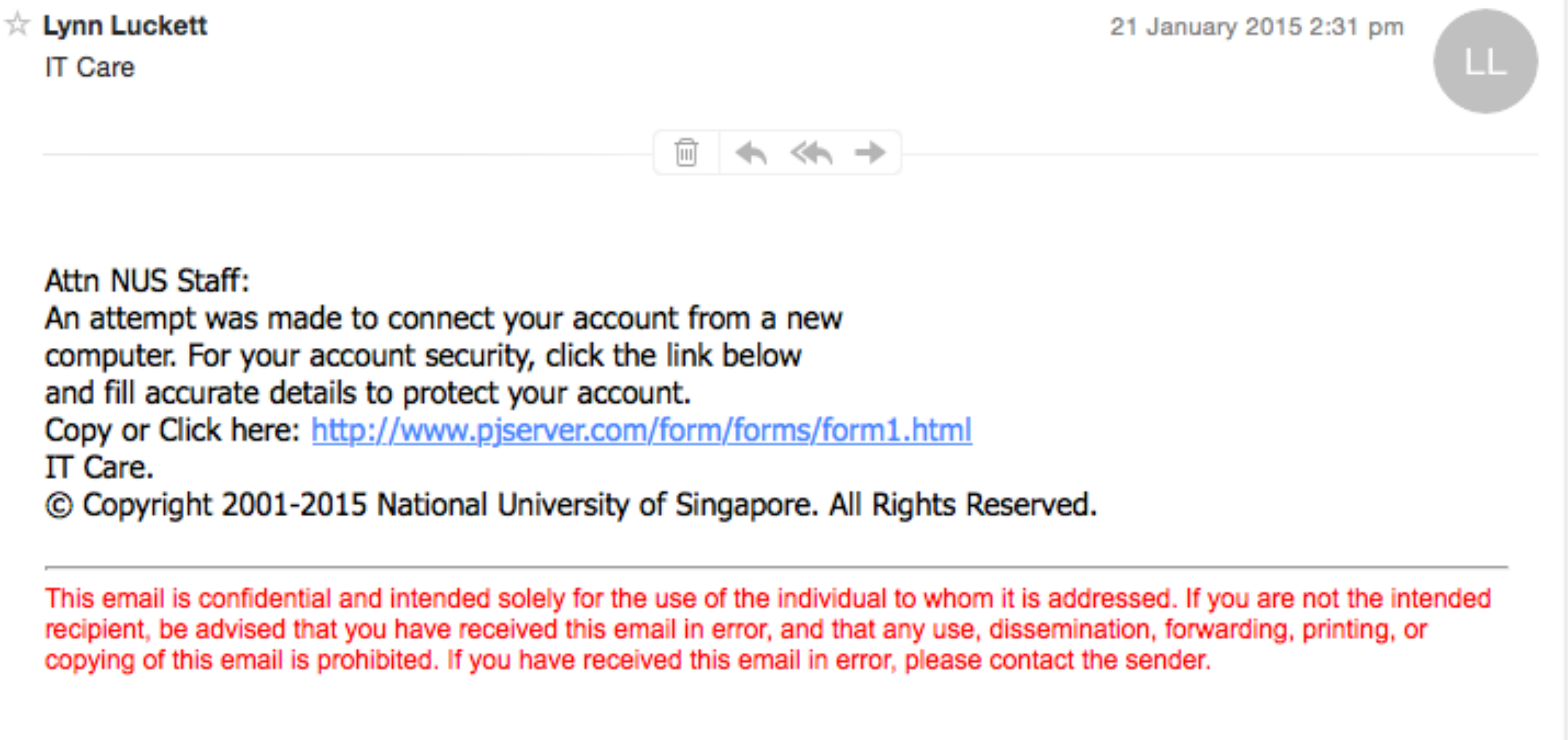


https://en.wikipedia.org/wiki/Hardware_keylogger

2. Phishing

- The victim is tricked to voluntarily sends the password to the attacker.

- Phishing attacks ask for password under some false pretense. Typically, it tricks the user to visit a website, which is a spoofed login web.



☆ **Lynn Luckett**
IT Care

21 January 2015 2:31 pm

Attn NUS Staff:
An attempt was made to connect your account from a new computer. For your account security, click the link below and fill accurate details to protect your account.
Copy or Click here: http://www.pjserver.com/form/forms/form1.html
IT Care.
© Copyright 2001-2015 National University of Singapore. All Rights Reserved.

This email is confidential and intended solely for the use of the individual to whom it is addressed. If you are not the intended recipient, be advised that you have received this email in error, and that any use, dissemination, forwarding, printing, or copying of this email is prohibited. If you have received this email in error, please contact the sender.

Phishing attack is a ***social engineering*** attack.

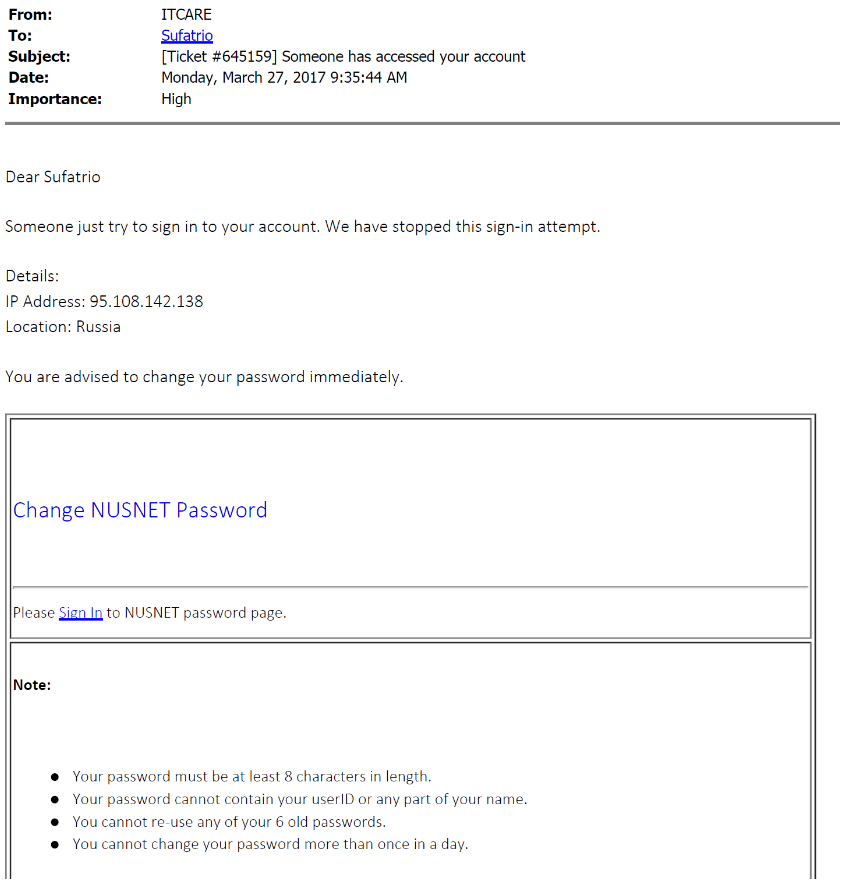
Wiki definition of social engineering:

“**Social engineering**, in the context of information security, refers to psychological manipulation of people into performing actions or divulging confidential information.”

http://en.wikipedia.org/wiki/Social_engineering_%28security%29

3. Spear Phishing

Phishing can be targeted to a particular small group of users (for example, NUS staff). Such attack is generally known as ***spear phishing,*** which is an example of ***targeted attacks.***



From:	ITCARE
To:	Sufatrio
Subject:	[Ticket #645159] Someone has accessed your account
Date:	Monday, March 27, 2017 9:35:44 AM
Importance:	High

Dear Sufatrio

Someone just try to sign in to your account. We have stopped this sign-in attempt.

Details:
IP Address: 95.108.142.138
Location: Russia

You are advised to change your password immediately.

Change NUSNET Password

Please Sign In to NUSNET password page.

Note:

- Your password must be at least 8 characters in length.
- Your password cannot contain your userID or any part of your name.
- You cannot re-use any of your 6 old passwords.
- You cannot change your password more than once in a day.

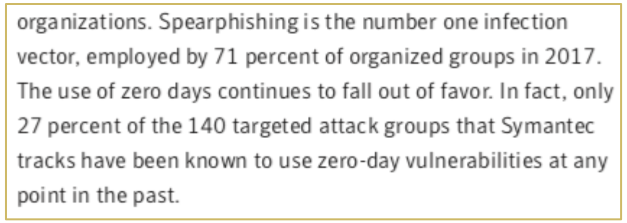
Although just a few slides and low tech….

Spear-phishing is extremely effective

“Spear phishing is the number one infection vector employed by 71 percent of organized groups in 2017.” *Internet Security Threat Report*, Symantec, Vol 23, 2018.

See the paragraph on phishing.
https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf

Many major incidents e.g. Singhealth, starts with phishing.



organizations. Spearphishing is the number one infection vector, employed by 71 percent of organized groups in 2017. The use of zero days continues to fall out of favor. In fact, only 27 percent of the 140 targeted attack groups that Symantec tracks have been known to use zero-day vulnerabilities at any point in the past.

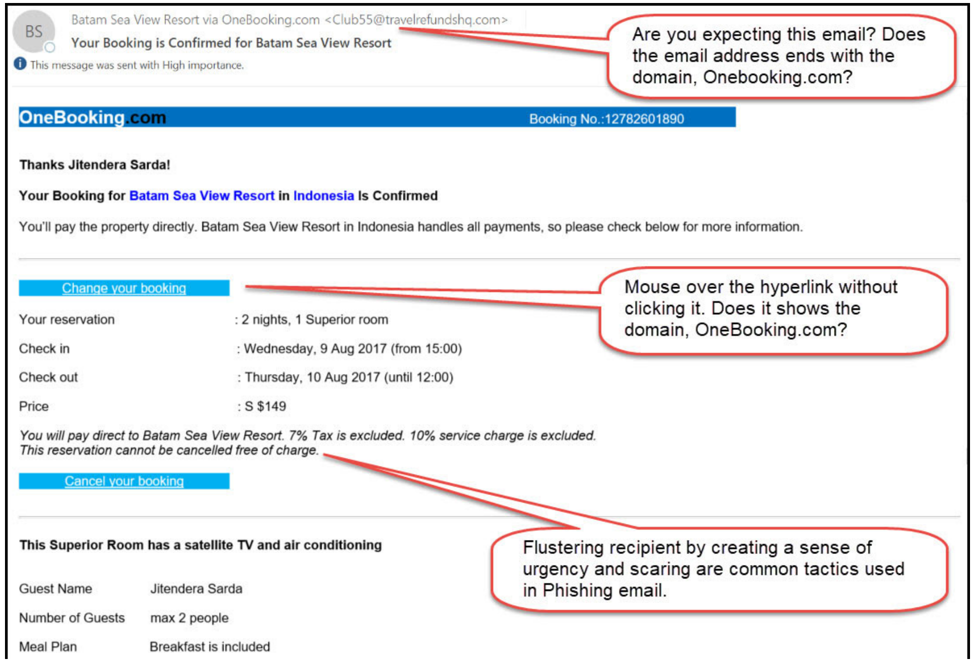
- Phishing can also be carried out over phone calls.

Terminologies: ***Phishing***, ***Pharming***, ***Vishing*** and ***Smishing***.

See http://csbweb.com/phishing.htm)

Phishing Prevention: User training.

- Training workshop, reminders.



Batam Sea View Resort via OneBooking.com <Club55@travelrefundshq.com>

Your Booking is Confirmed for Batam Sea View Resort

This message was sent with High importance.

OneBooking.com Booking No.:12782601890

Thanks Jitendera Sarda!

Your Booking for Batam Sea View Resort in Indonesia Is Confirmed

You'll pay the property directly. Batam Sea View Resort in Indonesia handles all payments, so please check below for more information.

Your reservation : 2 nights, 1 Superior room

Check in : Wednesday, 9 Aug 2017 (from 15:00)

Check out : Thursday, 10 Aug 2017 (until 12:00)

Price : S $149

You will pay direct to Batam Sea View Resort. 7% Tax is excluded. 10% service charge is excluded.
This reservation cannot be cancelled free of charge.

This Superior Room has a satellite TV and air conditioning

Guest Name Jitendera Sarda

Number of Guests max 2 people

Meal Plan Breakfast is included

Are you expecting this email? Does the email address ends with the domain, Onebooking.com?

Mouse over the hyperlink without clicking it. Does it shows the domain, OneBooking.com?

Flustering recipient by creating a sense of urgency and scaring are common tactics used in Phishing email.

From: NUS IT Care

- Embedded Phishing Exercise:
  - Like fire drill, authorized entities send out “phishing” emails to employees.

Phishing Prevention: Blacklisting

- Blacklisting. Repository site keeping lists of phishing site.
  - Example: phishtank.com
  - Organization actively monitor for phishing site. When a site is found, blacklist it.
  - Blacklist used by browser or firewall.

Way to enforce blacklisting:

1. Browser detect and block it.
2. Email client/server detect and move the email to a designated folder.
3. See right…. (I think this is not so common)

Send an email containing an url to your NUSNET email account. Some would be replaced with "trendmicro…". What is going on?

Original sent email:

Hi Alice,
I would like to share with you
http://www.nus.edu.sg.13452.com/as3
Bob

Received email:
The original url is replaced by

Hi Alice,
I would like to share with you
https://ddec1-0-en-ctp.trendmicro.com/wis/clicktime/v1/query?url=http%3a%2f%2fwww.nus.edu.sg.13452.com%2fas3&umid=fe07c1cf-a9b4-4d60-a4d4-484347a59038&auth=8d3ccd473d52f326e51c0f75cb32c9541898e5d5-8638dca46e331a551f059839819332e778199371
Bob

How to visually spot phishing website?

1) Check that there is a padlock in the address bar (we will spend a few weeks explaining this).
2) Check that the ***domain name*** in the url is correct.

With (1) + (2), and assuming browser is malware-free, then website is authentic.

Determining domain name may not be straightforward for many users. E.g. Which are correct domain names of DBS?

www.dbs.com.sg.1010.com
www.dbs.internet1.com.sg

www.dbs.com
www.dbs.com.sg
www.internet.dbs.com.sg

4. Cache

- When using a shared workstation (for e.g. a browser in airport), information keyed in could be cached. The next user can access the cache.

  (close the browser when using shared workstation)

5. Lost of password file

- The password file is stolen.

2.2.5 Password Strength

Remark:

In encryption, we “quantify” the key-strength by the size of the key if best known attack is exhaustive search.
E.g. Strength of 128-bit AES key is: 128 bits. (exhaustive search goes through $2^{128.}$ keys)

If best known attack is faster, then we quantify it by its equivalent in exhaustive search.
E.g. Strength of 1024-bit RSA key is: 80 bits. (time to break 1024 ~ time to exhaustive search 2^{80} keys)

What about password?
For password, we use an imprecise way to estimate measure: entropy. As it is difficult to describe entropy to the public, it is common to use length of randomly chosen characters.

Using Strong Password

- Truly random password: The password is chosen randomly & uniformly among all possible passwords of a certain length. High “entropy” but difficult to remember.

  3n5d!cvUD9cfm (10 characters)

- User selection:

  - Mnemonic Method Pbmbval!
  - Altered Passphrases Dressed*2*tge*9z
  - Combining and Altering Word B@nkC@mera

  (see **see** https://en.wikipedia.org/wiki/Password_strength#Guidelines_for_strong_passwords)

- Usability:
  - Strong passwords are difficult to remember.
  - It is difficult to enter alphanumeric passwords into mobile devices. There are alternatives, e.g. graphical or gesture-based.
  - Services that help users by storing passwords in cloud or locally. Nonetheless, need a credential for the service.

Remark: Password Entropy

- We often encounter this term "entropy" when quantifying strength of password. Entropy is a measurement of randomness. In this class we won't go into the definition of entropy. We can use the following example to have a sense of its meaning.

- Suppose a set ***P*** contains ***N*** unique passwords. Alice chooses her passwords by randomly & uniformly picking a password from the set ***P.*** Every password in ***P*** has an equal chance to be chosen (i.e. 1/N). In this case, by definition, the entropy of Alice's password is:

  $(log_2\ N)$ bits

- What if Alice doesn't choose the passwords uniformly, for e.g., the probability that she picks a word starting with letter "A" is higher than the probability that she picks a word starting with "z"? In such cases, the entropy is not ($\log_2$ N). By the definition of entropy, it is

$$-\sum_{i=1}^{N} p_i \log_2 p_i$$

  where p_i is the probability that Alice picks the *i*-th word in ***P***. (if we put p_i = 1/N, then we get $\log_2$ N.)

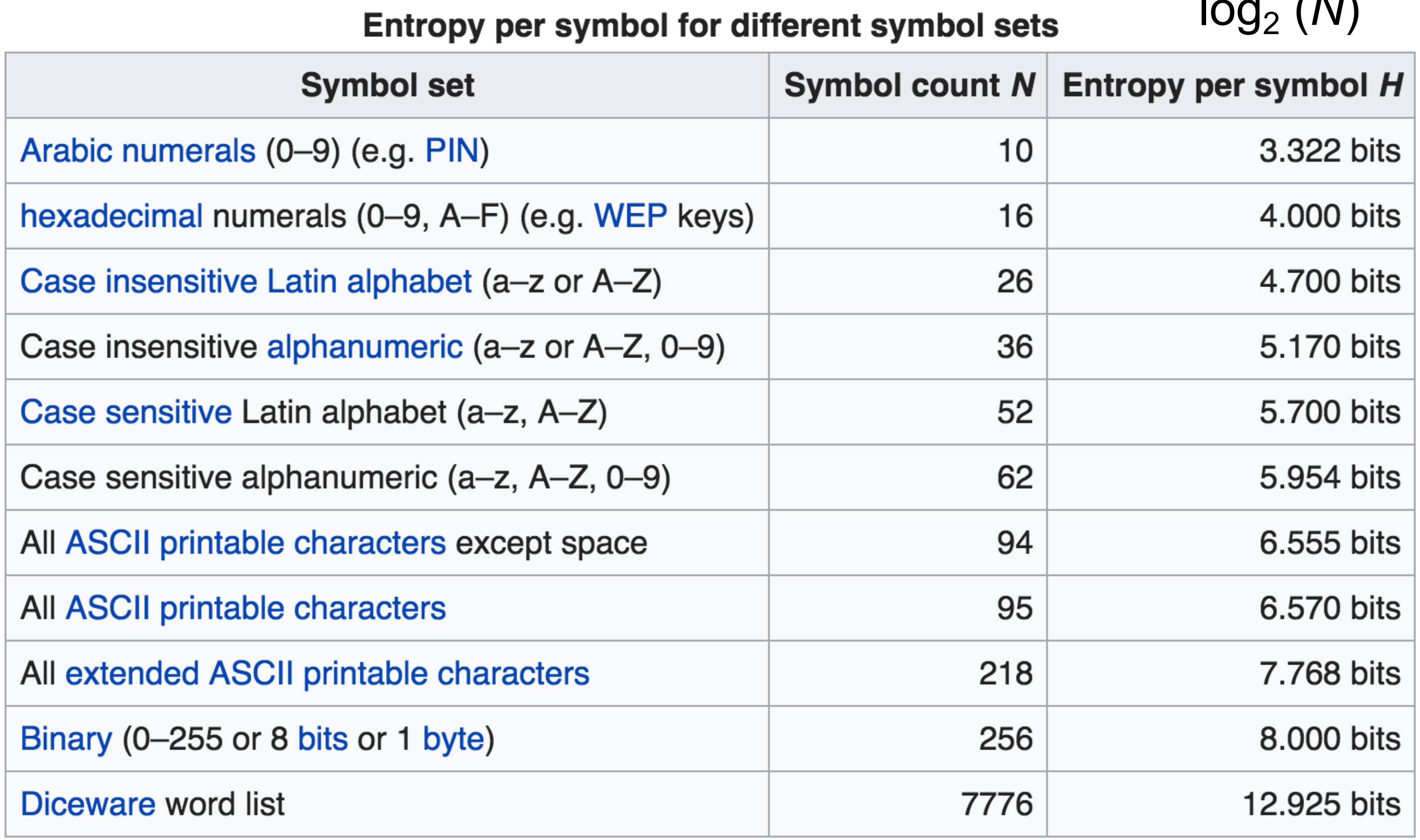
- It can be shown that, for the entropy to be highest for a set of *N* items, p_i must be 1/*N*. In other words, uniform choices.

- (omit if this further confuse you) Another way to measure randomness is min-entropy, which is

$$\min_i(-\log_2 p_i)$$

Suppose with probability 0.5, Alice picks her password as "Alice", and for probability 0.5, she uniformly chooses from ***P.*** That is, each word in P has probability 1/(2N) being chosen, and "Alice" has 0.5. Now, the entropy is roughly ((log N)- 1), which is high. However, there is good chance in correctly guessed her password. So, entropy might not be a good measure of password strength. Note that in this case, the min-entropy is low and is 1, correctly reflects the poor choice. (here, the string "Alice" is not in ***P***)

$\log_2(N)$



Entropy per symbol for different symbol sets

Symbol set	Symbol count *N*	Entropy per symbol *H*
Arabic numerals (0–9) (e.g. PIN)	10	3.322 bits
hexadecimal numerals (0–9, A–F) (e.g. WEP keys)	16	4.000 bits
Case insensitive Latin alphabet (a–z or A–Z)	26	4.700 bits
Case insensitive alphanumeric (a–z or A–Z, 0–9)	36	5.170 bits
Case sensitive Latin alphabet (a–z, A–Z)	52	5.700 bits
Case sensitive alphanumeric (a–z, A–Z, 0–9)	62	5.954 bits
All ASCII printable characters except space	94	6.555 bits
All ASCII printable characters	95	6.570 bits
All extended ASCII printable characters	218	7.768 bits
Binary (0–255 or 8 bits or 1 byte)	256	8.000 bits
Diceware word list	7776	12.925 bits

From https://en.wikipedia.org/wiki/Password_strength#Guidelines_for_strong_passwords

Recap Online vs offline attack:

- **Online**: To check whether a password is correct, the attacker needs to interact with a server.

- **Offline**: After obtained the necessary info, to check whether a password is correct, the attacker does not need interactions with a server.

Guideline on Password strength to guard against online, offline dictionary attack

see https://en.wikipedia.org/wiki/Password_strength#Guidelines_for_strong_passwords

- Human generated password are not truly random.

- It is difficult to estimate entropy of human-generated passwords. NITS (**NIST Special Publication 800-63-2)** suggested a way. https://en.wikipedia.org/wiki/Password_strength#Guidelines_for_strong_passwords**.**   However, a revision **SP 800-63** (Revision 3) in 2017 drops this approach.

- **Online**: Recommendation by RFC 4086 (Randomness Required for Security) https://tools.ietf.org/html/rfc4086 suggests the password to have *at least 29 bits of entropy* to be secure against online attacks.   It recommends *at least 49 bits* for "higher security".

- **Offline**: When offline attacks are possible, the requirement on passwords must be stricter. Somehow, I can't find guideline on passwords under offline attacks. One would expect that it should be equivalent to requirement of symmetric key. Since NIST recommend 128 bits for crypto keys, in this course, let's take *128 bits* entropy as the requirement.

Enhancing Password system

- To make online dictionary attack more difficult, many systems intentionally add delay into login session, (for example, wait for 1 second before next attempt), or locked the account after a few failed attempts.

- To make offline dictionary attack more difficult, a KDF can be applied to the password. The KDF forces intensive computation but incur high overhead during legitimate usage.

- Many systems checks for weak password when user registers/changes password.

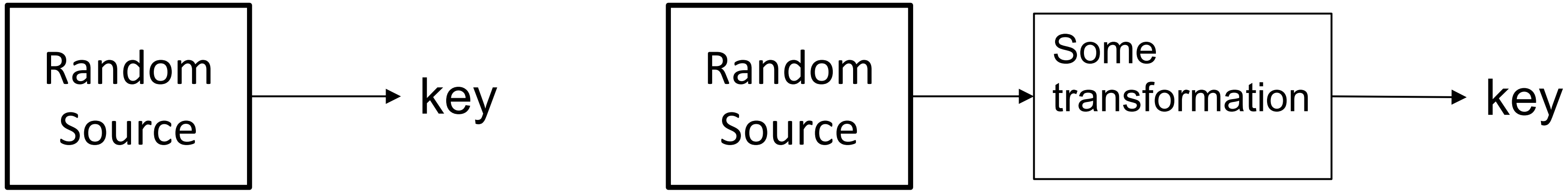
- Some systems require regular changes of passwords, which is controversial. (Many believe that frequent changes of passwords could lower security.)

  See https://www.schneier.com/blog/archives/2016/08/frequent_passwo.html

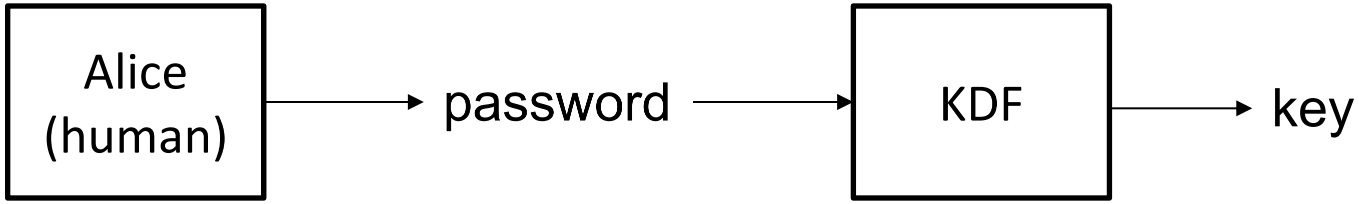
Password Vs Secret Key in crypto (revisit slide 53-59 during topic on hash)

revisit after Topic 3

- Password are generated by human and to be remembered by human.
- Secret keys in crypto (e.g. 128-bit AES encryption key, 1024-bit RSA key) are machine generated. The key is directly generated from some truly random source or derived based on the source.



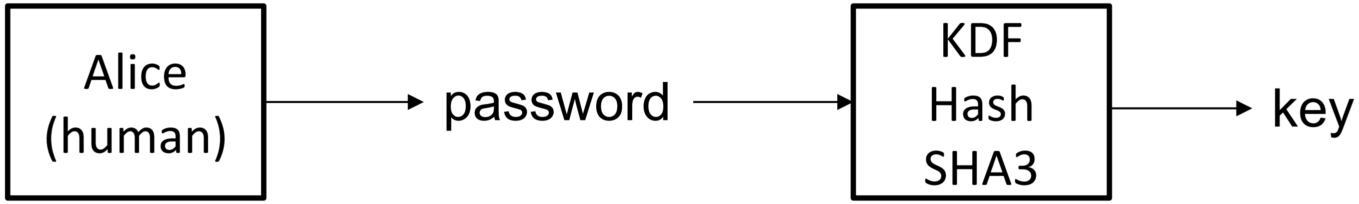
- Sometime, the password are used as the source for crypto secret key. E.g. using password to encrypt a file. The transformation is called KDF key derivation function.



KDF (revisit after we covered Hash)

revisit after Topic 3

- A typical choice of KDF is cryptographic hash such as SHA3. i.e.



Alice (human)

password

KDF Hash SHA3

key

- To make offline dictionary attack more difficult, it is desired to have a KDF that consume a lot of compute time. So, ironically, hash like SHA3 is designed to be efficient, but KDF wants it to be very slow.

- This is achieved by iterative applications of hash, say *n* times.

$$\text{E.g. key} = \underbrace{\text{H (H (... H}}_{n} \text{ (password) ...)).}$$

Now, a test carried out by dictionary attack would incur an overhead factor of *n.*

Additional protection to password files

revisit after Topic 3

- The *password file* stores userid and password

- The password file could be leaked, due to insider attack, accidental leakage, system being hacked, etc. Recap: the password file store the userid+password.

- There are many well-known incidents where unprotected or weakly protected password files are leaked, leading to a large number of passwords being compromised. (2012 Linkedin https://en.wikipedia.org/wiki/2012_LinkedIn_hack)

- Hence, it is desired to add an additional layer of protection to the password file. (not all files are equally important.)

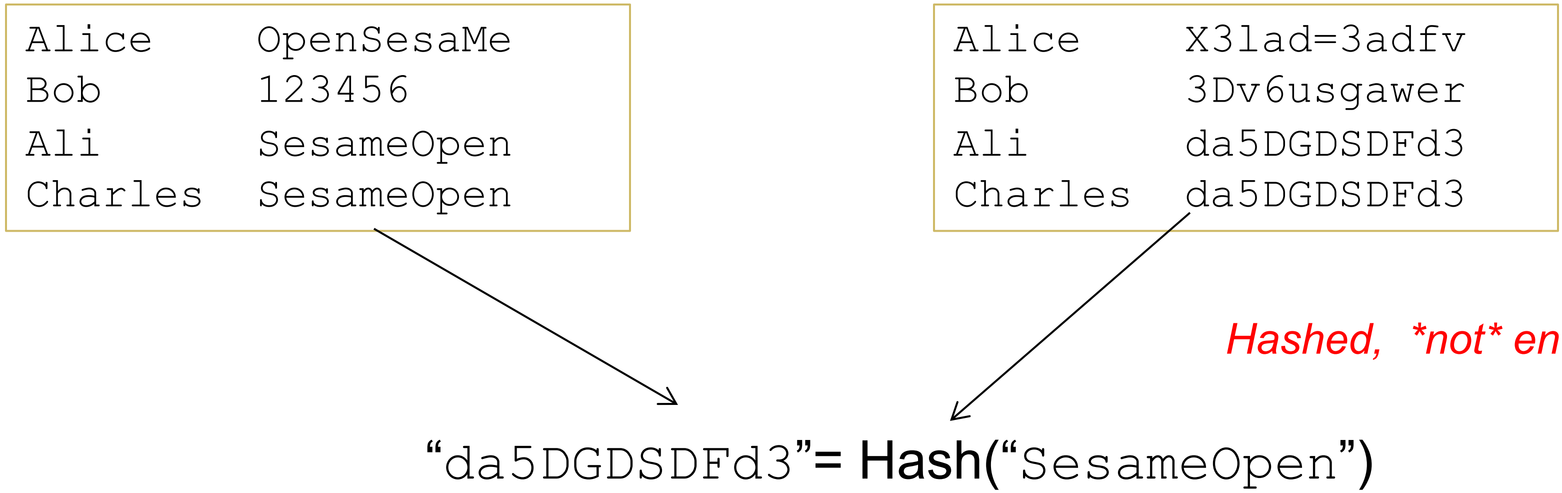
(revisit this slide after hash is covered)

revisit after Topic 3

Password file should be "hashed" and "salted".

(textbook ([PF]pg 46) uses the term "encrypted". This is a wrong choice of term. For encryption, by definition, there is a way to decrypt and get back the original password. For cryptographically secure hash, it is infeasible to recover the password from the hashed value. In fact, to be secure, we don't want to have a way to recover the password.)

- During authentication, the password entered by the entity is being hashed, and compared with the the value stored in the password file.



Password in clear

Alice	OpenSesaMe
Bob	123456
Ali	SesameOpen
Charles	SesameOpen

Hashed Password

Alice	X3lad=3adfv
Bob	3Dv6usgawer
Ali	da5DGDSDFd3
Charles	da5DGDSDFd3

*Hashed, *not* encrypted.*

"da5DGDSDFd3"= Hash("SesameOpen")

revisit after Topic 3

To verify whether a password ***P*** belongs to a user ***U***, the following are carried out:

1. Compute *d* = Hash (***P***).
2. If <***U***, *d*> is in the password file, then accept, else reject.

(revisit this slide after hash is covered)

revisit after Topic 3

It is desired that the same password would be hashed to two different values for two different userid. Why? (*rainbow table*)

This can be achieved by using salt.

Password in clear

Alice	OpenSesaMe
Bob	123456
Ali	SesameOpen
Charles	SesameOpen

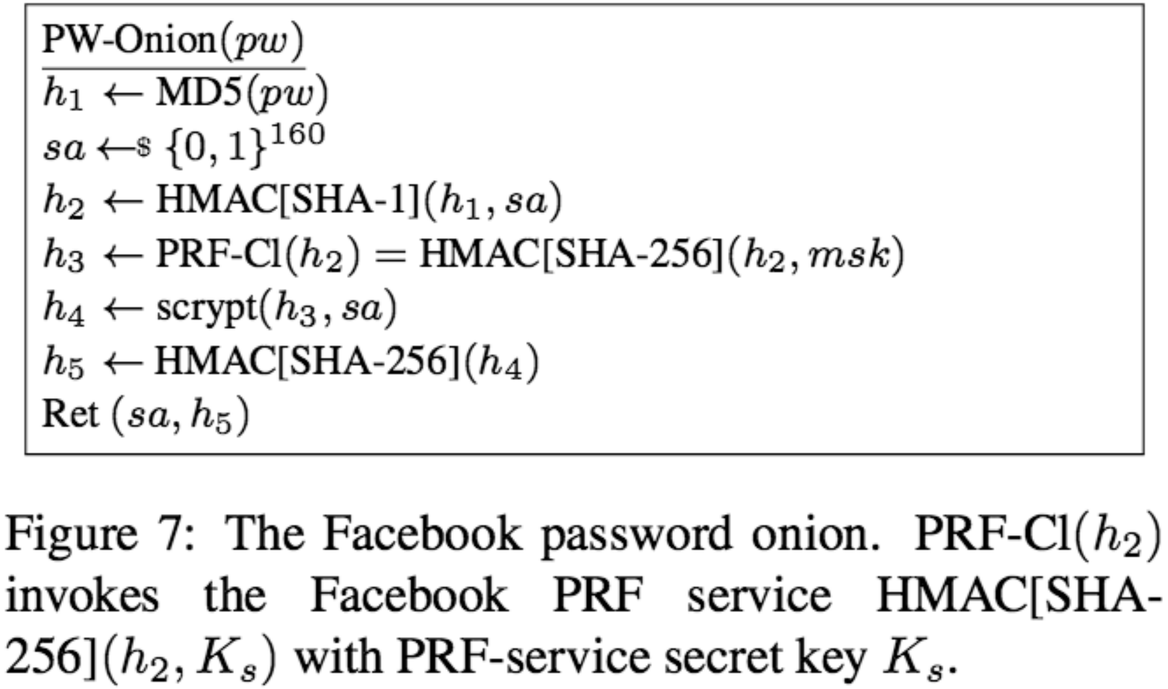
Salted Password

Alice,	Adf3,	39Gkaj10Dmf
Bob,	a3gh,	d978bjklDFD
Ali,	f8ad,	DJk34hoaev7
Charles,	10vd,	K108ELvio2B

“DJk34hoaev7”= Hash(“f8adSesameOpen”)
“K108ELvio2B”= Hash(“10vdSesameOpen”)

(Optional) How Facebook protects the passwords

revisit after Topic 3



```
PW-Onion($pw$)
$h_1 \leftarrow \mathrm{MD5}(pw)$
$sa \leftarrow_{\$} \{0,1\}^{160}$
$h_2 \leftarrow \mathrm{HMAC[SHA\text{-}1]}(h_1, sa)$
$h_3 \leftarrow \mathrm{PRF\text{-}Cl}(h_2) = \mathrm{HMAC[SHA\text{-}256]}(h_2, msk)$
$h_4 \leftarrow \mathrm{scrypt}(h_3, sa)$
$h_5 \leftarrow \mathrm{HMAC[SHA\text{-}256]}(h_4)$
Ret $(sa, h_5)$
```

Figure 7: The Facebook password onion. PRF-Cl(h_2) invokes the Facebook PRF service HMAC[SHA-256](h_2, K_s) with PRF-service secret key K_s.

from A. Everspaugh et. al. The Pythia PRF Service, USENIX Security 2015

PRF-C1(h2) is computed by a remote server, using a master-key stored only in that server.

SCRYPT: a hash that is computationally expensive to compute. (ironically, crypto hash such as SHA3 is designed to be efficient.)

not included. Keeping it here for optional info.

2.2.6 ATM Skimming

- This demonstrate password stealing.
- Was very common and fortunately less prevalent now. Still have many reported cases.

(2023 incidents in Australia https://news.sophos.com/en-us/2023/08/15/grab-hold-and-give-it-a-wiggle-atm-card-skimming-is-still-a-thing/)

ATM Card

not included.
Keeping it here for
optional info.

- To get authenticated, the user presents (1) a card, and (2) a PIN.
  - The card contains a magnetic strip, which stores the user account id. Essentially, the magnetic strip simplifies the input of account id into the ATM system: instead of keying it in, just inserting the card.
  - The PIN plays the role of password.

Data are encoded into the magnetic strip using well-known standards. Given physical access to a card, anyone (including attackers) can “copy” the card by reading the info from the card and writing it to the spoofed card. So, it is easy to forge the card.

this card can be purchased from ebay.

http://colnect.com/en/bank_cards/bank_card/4130-ATM_Card-Bank_of_America-United_States

ATM skimmer

not included.
Keeping it here for
optional info.

An ATM skimmer steals info in the card and PIN (password).

The skimmer consists of:

1. a card-reader attached on top of existing ATM reader;
2. a camera overlooking the keypad, or a spoofed key-pad on top of existing keypad;
3. some means to record and transmit the information back to the attacker.

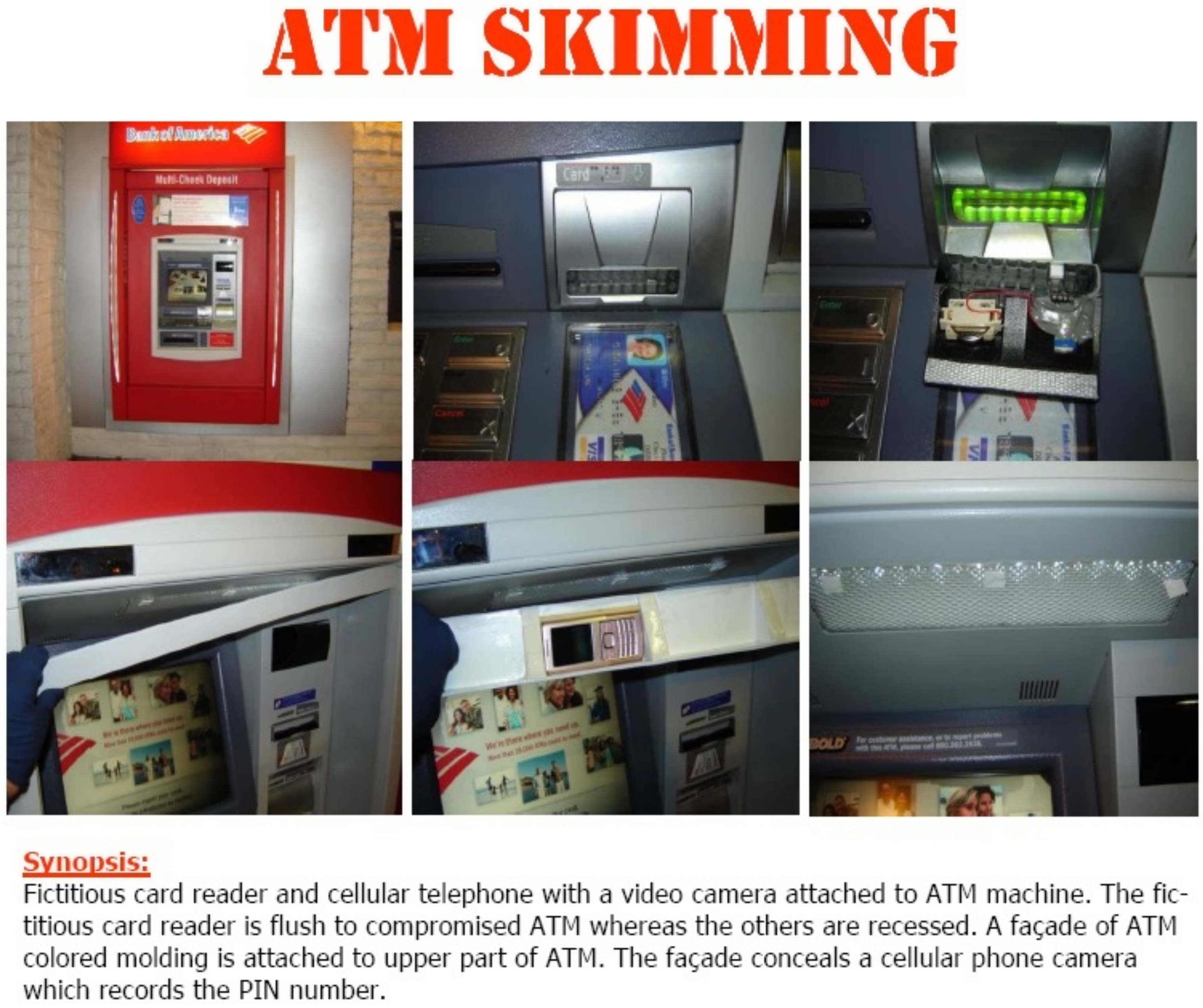
With the information obtained from (1), the attacker can forge the victim's ATM card.
With (2), the attacker obtain the PIN.

Well known incidents in Singapore: DBS in 2012.

"$1 million stolen from the bank accounts of 700 DBS and POSB customers." - See
http://news.asiaone.com/News/Latest+News/Singapore/Story/A1Story20120223-329820.html

Yet another self-explanatory image

not included.
Keeping it here for
optional info.



ATM SKIMMING

Synopsis:
Fictitious card reader and cellular telephone with a video camera attached to ATM machine. The fictitious card reader is flush to compromised ATM whereas the others are recessed. A façade of ATM colored molding is attached to upper part of ATM. The façade conceals a cellular phone camera which records the PIN number.

http://pbgcrimewatch.org/images/reports/ATM_Skimming.jpg

Some Fun Videos to Watch: POS Skimmer Installation

not included.
Keeping it here for
optional info.

https://www.youtube.com/watch?v=_BFRD8_LrcM

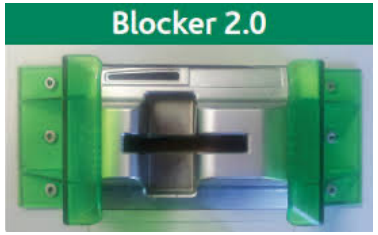
CCTV caught someone deploying a Point-Of-Sale skimmer (similar to ATM skimmer)

More video:
“Why Chip Credit Cards Are Still Not Safe From Fraud”
https://www.youtube.com/watch?v=gJo9PfsplsY

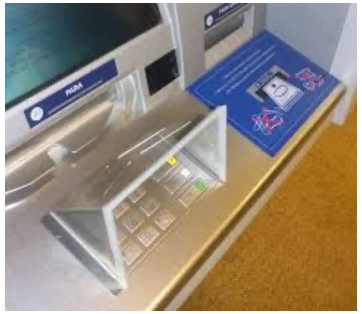
Measures

not included.
Keeping it here for
optional info.

- Anti-Skimmer device: A device that prevents external card reader to be attached onto the ATM.



Shielding the keypad.



- Awareness among users.

- Change to smartcard (smartcard is unforgeable).

2.3 Biometric

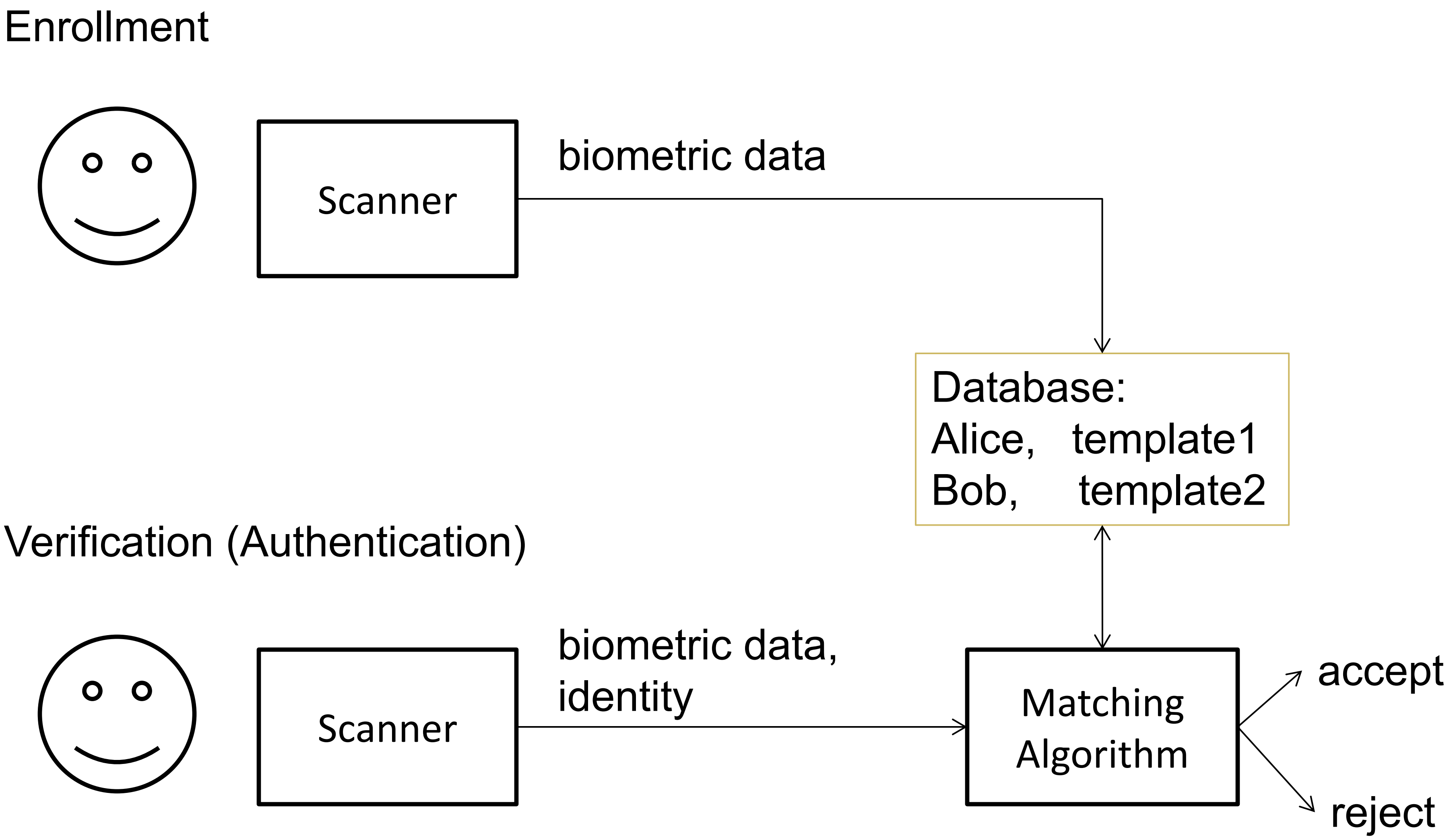
- Biometric uses unique physical characteristics of a person for authentication.

- During ***enrollment***, a ***template*** of a user's biometric data is captured and stored (same as bootstrapping in password system).

- During ***verification***, biometric data of the person-in-question is captured and compared with the template using a *matching algorithm.* The algorithm decides whether to accept or reject.

Biometric can be used for *identification* (identify the person from a database of many persons), or *verification* (verify whether the person is the claimed person). Here, we focus on verification.

Essentially, biometric data is the password.



Differences between Biometric and Password

Password	Biometric
Can be changed (revoked)	Can’t
Need to remember	Don’t have to
Zero non-matched rate	*Probability of error*
Users can pass the password to another person	Not possible

- Unlike password, there are inevitable noise in capturing the biometric data, leading to error in making the matching decision: FMR (False match rate) and FNMR (false non-match rate) .

$$\text{FMR} = \frac{\text{number of successful false matches (B)}}{\text{number of attempted false matches (B+D)}}$$

(false positive)

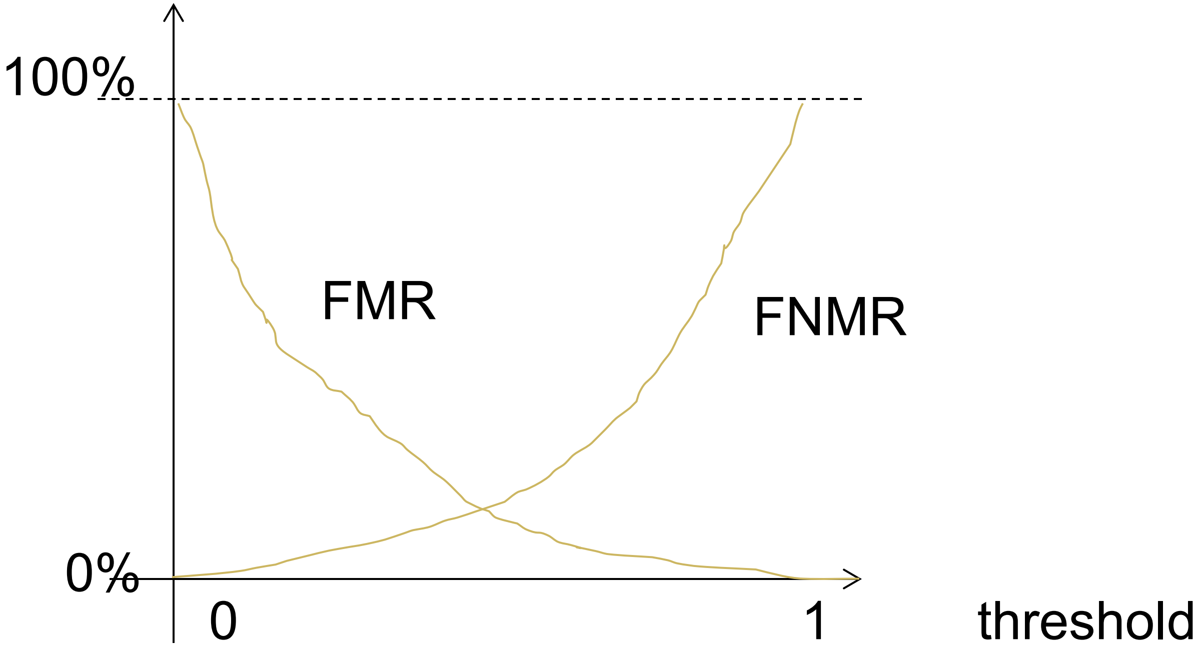
$$\text{FNMR} = \frac{\text{number of rejected genuine matches (C)}}{\text{number of attempted genuine matches (A+C)}}$$

(false negative)

	accept	reject
genuine attempt	A	C
false attempt	B	D

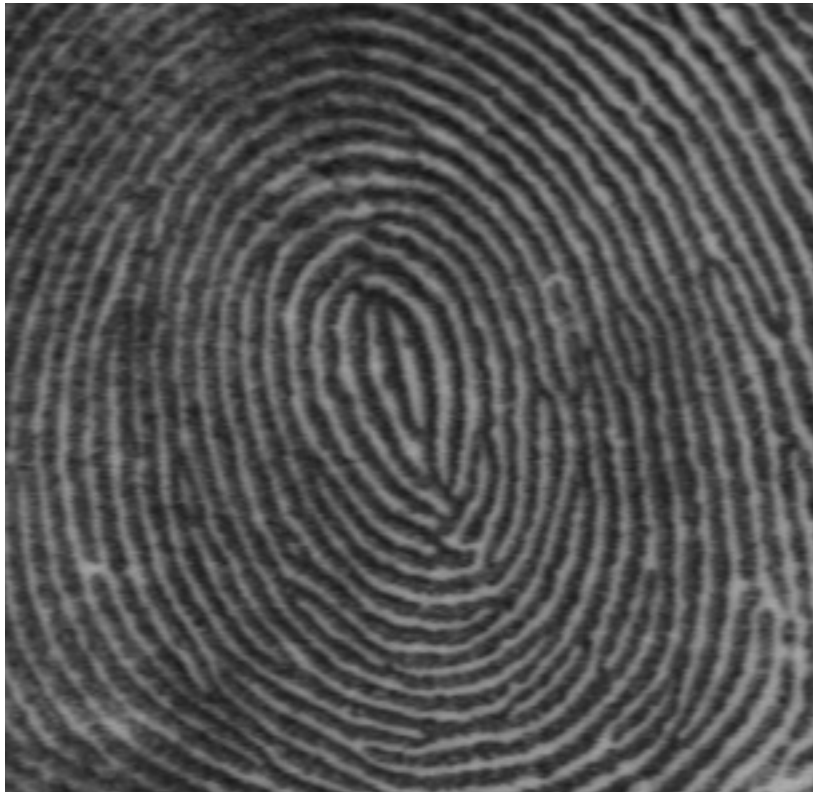
Optional:
Other definitions
Recall: A/(A+C)
Precision: A/(A+B)
False negative, False positive, True positive, True negative, F1 score, …
See https://en.wikipedia.org/wiki/Precision_and_recall for a list of definition on difference combination of these 4 numbers A, B, C, D.

The matching algorithm typically makes decision based on some adjustable threshold. By adjusting the threshold, the FMR and FNMR can be adjusted. (lower threshold => more relax in accepting, higher threshold => more stringent in accepting).

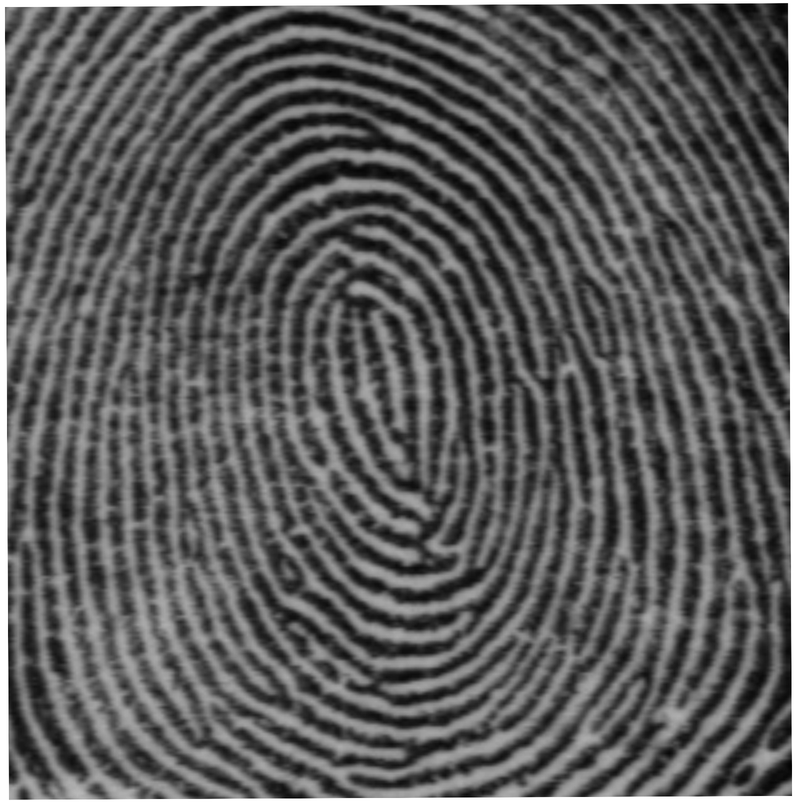


how to set the threshold? Depend on application.

Example on Fingerprint to illustrate the noise

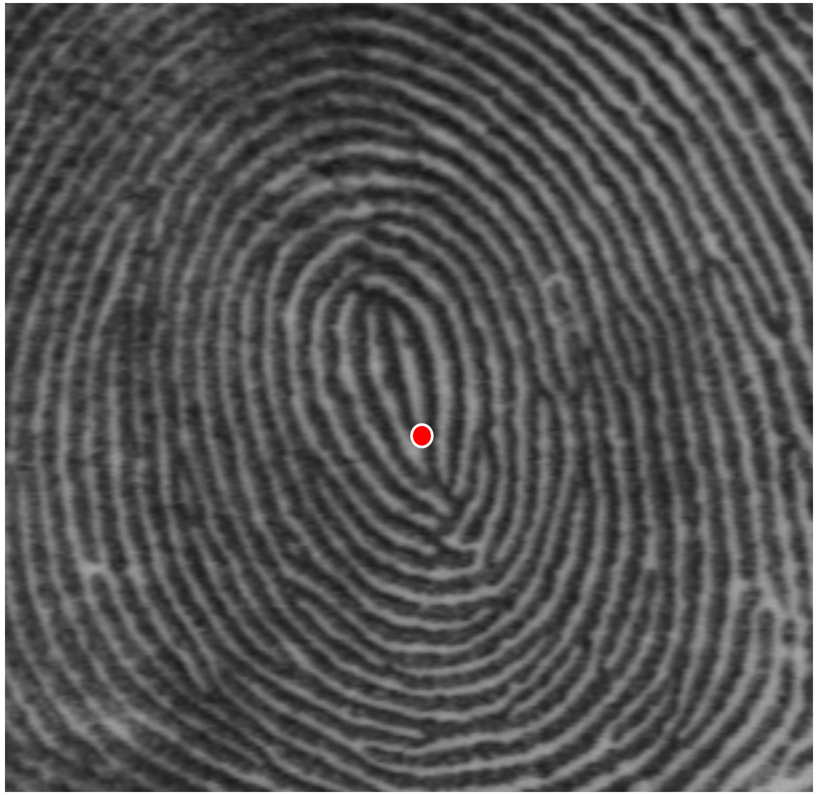


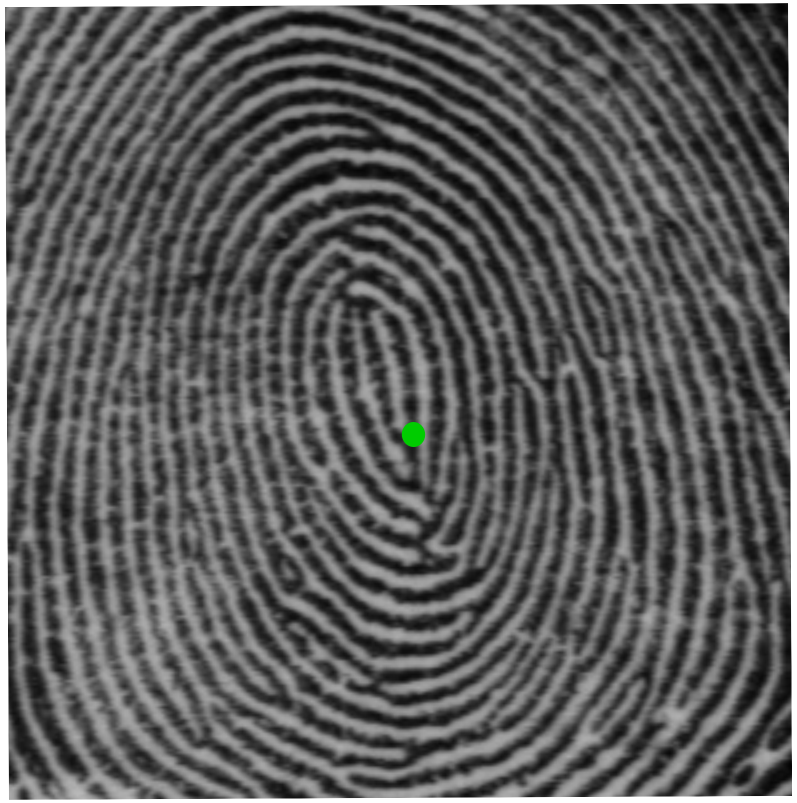
First scan of a
finger



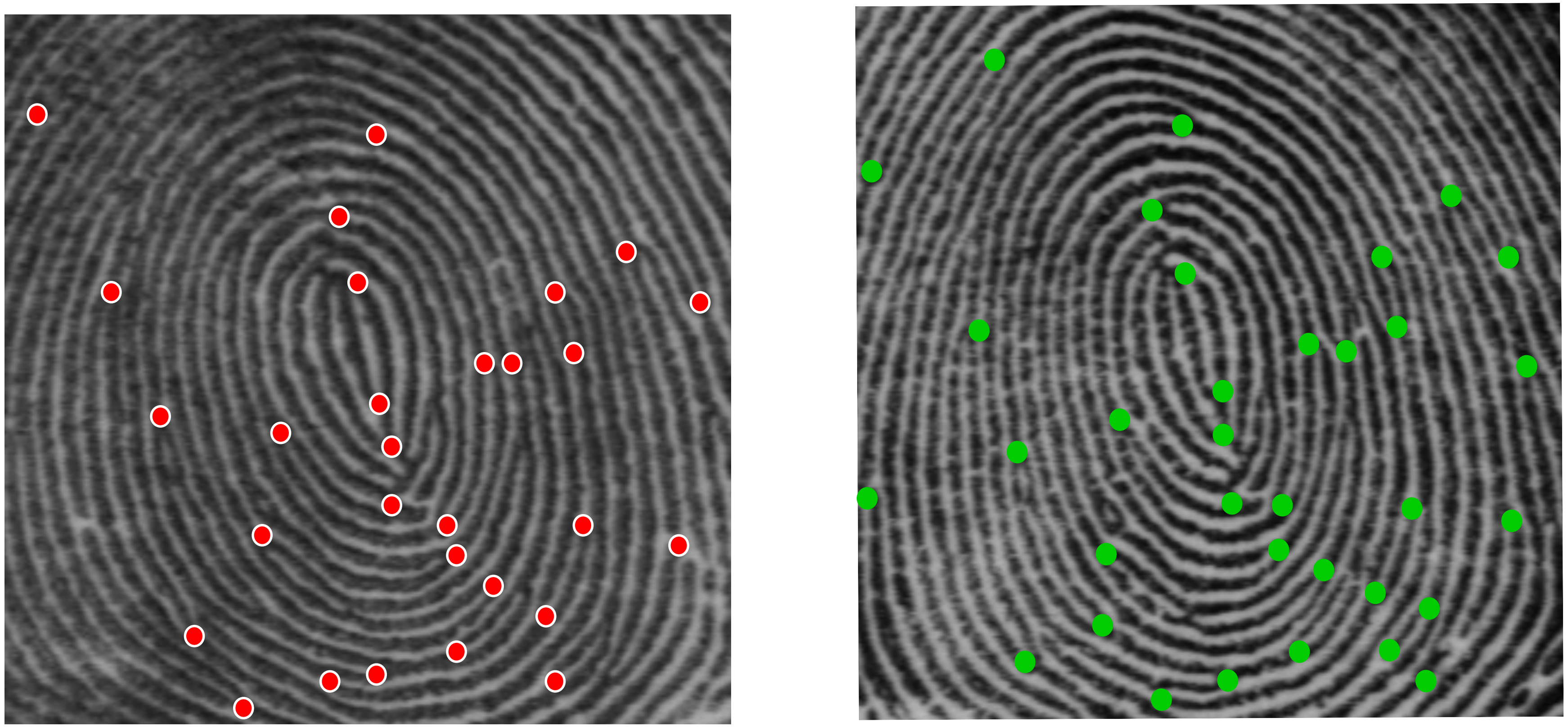
Another scan
of the same finger

Background: Fingerprint



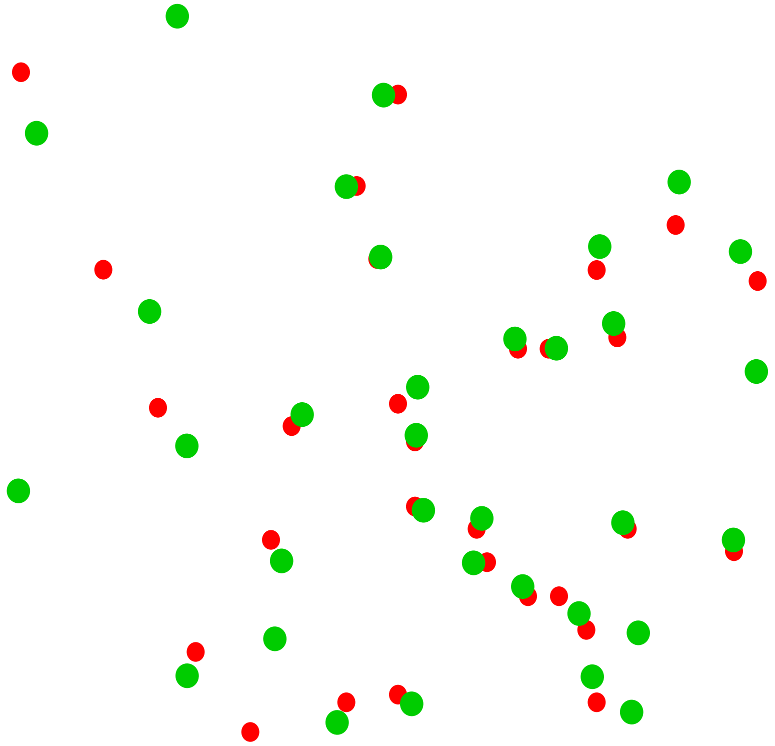


A feature point



The set of feature points
(known as *minutiae* for fingerprint).

The features points extracted from the two scans are similar but not exactly the same.



How good is fingerprint as a biometric?

Performance depends on the quality of the scanner.

EER can range from 0.5 to 5% depending on quality of scanners.

see result of Fingerprint Verification Competition

FVC2006 http://bias.csr.unibo.it/fvc2006/default.asp

Attack of biometric system

- Some biometric data could be easily spoofed as seen in movies. see http://www.wikihow.com/Fake-Fingerprints on how to make a fake fingerprint.

- A biometric system could include an addition mechanism on ***liveness detection.*** This mechanism verifies that the subject is indeed “live”, instead of spoofed materials, say a photograph.

- Example of liveness detection: temperature sensor in fingerprint scanner.

2.4 *n*-Factor Authentication (2FA) and multi-step verification

n-factor Authentication

Require at least two different authentication “factors.”
Example of factors:

1) Something you know:	Password, Pin.
2) Something you have:	Security token, smart card, mobile phone, ATM card.
3) Who you are:	Biometric.

It is called a 2-factor authentication if 2 factors are employed.

MAS (Monetary Authority of Singapore) expects all banks in Singapore to provide 2-factor authentication for e-banking.

[Gollmann] listed 2 additional factors (what you do, where you are). Most literatures only listed the above 3.

Something you have

Examples: ATM card, mobile phone, OTP token.

- ***One Time Password token.***

A hardware that generates one time password (i.e. password that can be used only once). Each token and the server share some secrets. There are two types:

1. **Time-based**: Based on the shared secret and current time interval, a password ***K*** is generated. Now, both server and the user has a common password ***K.***

2. **Sequence-based:** An event (for e.g. user pressing the button) triggers the change of the password.

Note: Not to be confused with "one-time pad"

Example of 2FA (1): Password + Mobile phone(SMS)

Registration:

User gives the server his mobile phone number and password.

Authentication:

(1) User sends password and username to server.

(2) Server verifies that the password is correct. Server sends a one-time-password (OTP) to the user through SMS.

(3) User receives the SMS and enters the OTP.

(4) Server verifies that the OTP is correct.

What you know: password.

What you have: The unforgeable SIM card in the phone.

Example of 2FA (2): Password + OTP Token

Registration:

The server issues a hardware OTP token to the user. The token contains a "secret key" ***k*** that the server knows. User registers a password.

Authentication:

(1) User "presses" the token. The token generates (can be time-based or sequence-based) and displays a one-time-password.

(2) User sends password, username, and OTP to server.

(3) Since the server has the "secret key", the server can also compute the OTP. Server verifies that the OTP and password are correct.

Some OTP includes a keypad for user to enter values. Can you give a scenario where OTP+keypad provide more security than solely OTP? (Give an attack that OTP+keypad can prevent but OTP)

Example of 2FA (2): Password + OTP Soft token.

Mobile phone can take the role of “hardware token”. This is also known as “Soft Token”.

Registration:

1. User installs the authentic Soft Token apps. During installation, some form of verification is carried out. After verification, a “secret key” ***k*** is established between the server and the mobile phone.
2. Separately, user registers a password with the server.

Authentication:

(1) The Soft token app establishes connection to the server, using the secret key ***k*** for authentication.

(2) User via another apps or browser, send request for a transaction T to the server. The apps/browser asked for user password. (in many cases, the password are stored in the app/browser, and the app/browser submits the password on behalf of the user.)

(3) The server contact the Soft-Token. The soft token app displays the transaction T, and ask the user, are you sure? If user click yes, send confirmation to server.

(4) After received confirmation from Soft-Token, carry out the transaction.

Likewise, password is what-you-know. Ownership of the software token (which essentially is the secret key k generated and managed by the apps) is what-you-have.

Example of 2FA (3): smartcard + fingerprint (Door access system)

Registration:

The server issues a smartcard to the user (note that the smartcard contains a secret key K). The user enrolls his/her fingerprint.

Authentication:

(1) User inserts smartcard to the reader. The reader obtains the user identity and verifies whether the smartcard is authentic.   If so, continue.

(2) User presents fingerprint to the reader. The reader performs matching to verify that it is authentic. If so, open door.

2-Step Verification

- Many online platforms use email account as the additional factor (e.g. to login, need a password and a link that is sent to the email account). Some may argue that since email account can be accessed using another password, hence basically it is a 2-password authentication method. Both are “what-you-know” and thus cannot be called “2-factor”. The argument is reasonable, nonetheless, many platforms still call it “2-factor”.

- Google calls theirs a 2-step verification. I guess this is to avoid the above confusion with the more narrowed definition of 2-factor authentication.

- To compromise 2-step verification, the attacker needs to obtain passwords of both accounts, which is more difficult. Hence 2-step is more secure than 1-step.

- (Terminology: **out-of-band**) In a 2-step verification, typically there are two communication channels. A main one (the web) and a separate channel (e.g. email) for additional authentication. The non-main channel is also called “out-of-band” channel.