Chapter 3: Authentication

Xianghang Mi

中国科学技术大学
University of Science and Technology of China

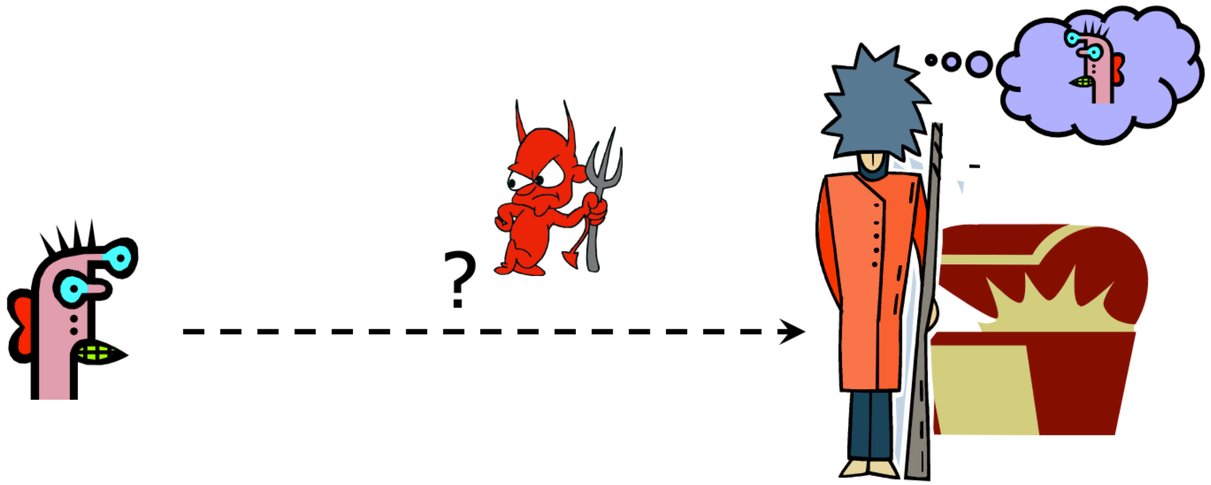
Outline

- Authentication Concepts
- Authentication Scenarios
- Authentication Primitives
- Authentication Protocols

Outline

- Authentication Concepts
- Authentication Scenarios
- Authentication Primitives
- Authentication Protocols

The Authentication Problem



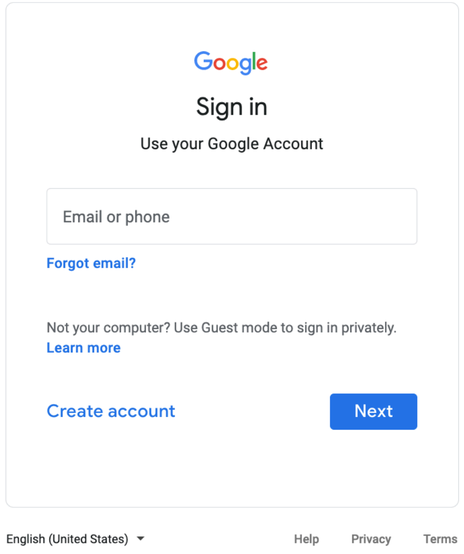
How do you prove to someone that
you are who you claim to be?

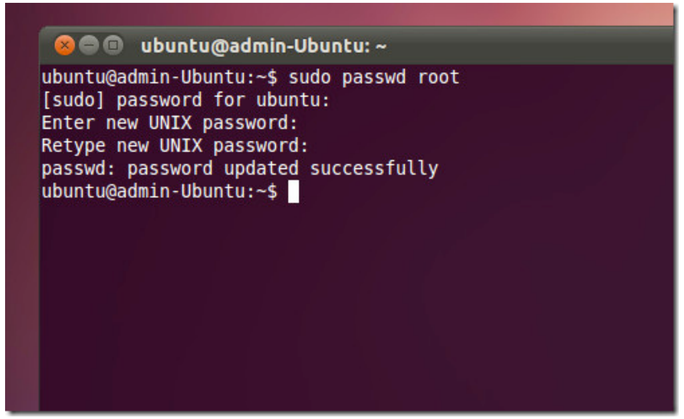
Any system with access control must solve this problem.

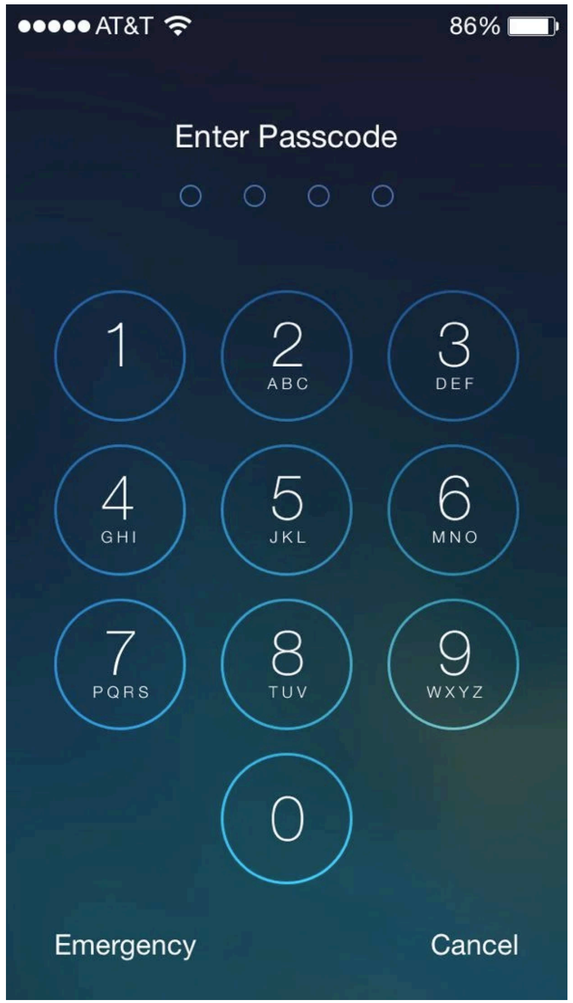
Authentication is the act of proving an identity assertion, such as the identity of a computer system user.

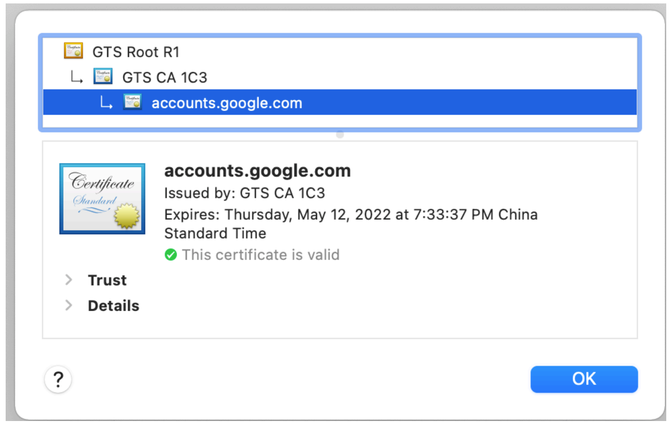
Outline

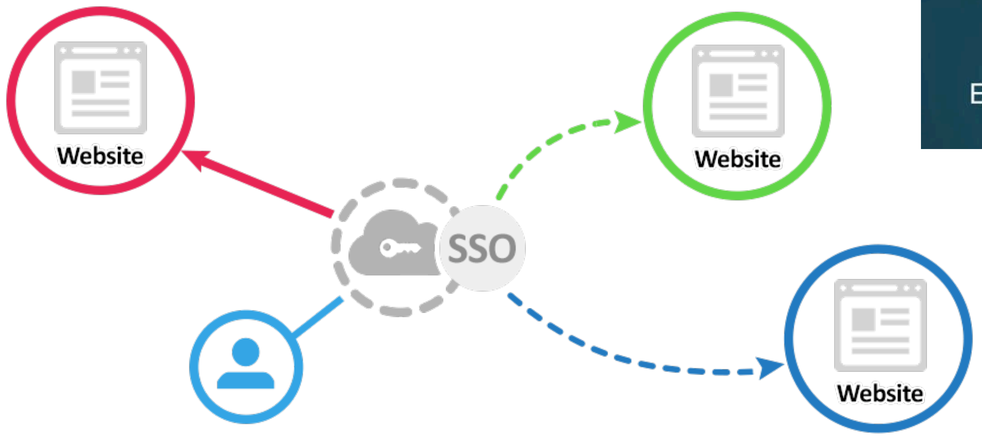
- Authentication Concepts
- Authentication Scenarios
- Authentication Primitives
- Authentication Protocols

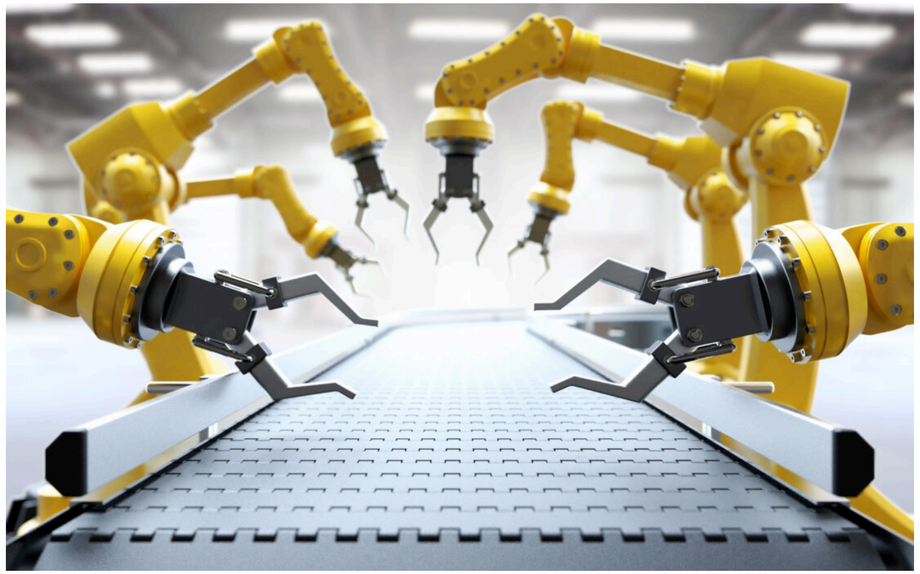

Google
Sign in
Use your Google Account
Email or phone
Forgot email?
Not your computer? Use Guest mode to sign in privately.
Learn more
Create account
Next
English (United States)
Help
Privacy
Terms


ubuntu@admin-Ubuntu: ~
ubuntu@admin-Ubuntu:~$ sudo passwd root
[sudo] password for ubuntu:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
ubuntu@admin-Ubuntu:~$


AT&T
86%
Enter Passcode
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0
Emergency
Cancel


GTS Root R1
GTS CA 1C3
accounts.google.com
accounts.google.com
Issued by: GTS CA 1C3
Expires: Thursday, May 12, 2022 at 7:33:37 PM China Standard Time
This certificate is valid
Trust
Details
OK


Website
Website
SSO
Website



Dynamics

How?

Outline

- Authentication Concepts
- Authentication Scenarios
- Authentication Primitives
- Authentication Protocols

How to

- Something the individual **knows**
  - Password, PIN, answers to prearranged questions, patterns, etc.
- Something the individual **possesses** (token)
  - Smartcard, electronic keycard, physical key,
- Something the individual **is** (physical/static biometrics)
  - Fingerprint, retina, face, brain wave

How to

- Something the individual **does** (behavioral/dynamic biometrics)
  - Voice pattern, handwriting, typing rhythm (keystroke), gait
- Where you are
  - IP, geolocation, etc.

0x19a3c400,
0x10c2a100,
0xPassword,
0x656ef500,
0xac121200,

Password Authentication

- Widely used line of defense against intruders
  - User provides **name/login** and **password**
  - System compares password with the one stored for that specified login
- A Long-existing authentication technique adopted well before the incubation of the cyberspace.

Attacks Targeting Password

- In 2012, 76% of network intrusions exploited weak or stolen credentials (username/password)
  - Source: Verizon Data Breach Investigations Report
- In Mitnick's "Art of Intrusion" 8 out of 9 exploits involve password stealing and/or cracking
  - First step after any successful intrusion: install sniffer or keylogger to steal more passwords
  - Second step: run cracking tools on password files

Attacks targeting Password

- Default password attack conducted by the Mirai botnet

Username	Password
666666	666666
888888	888888
admin	*(none)*
admin	1111
admin	1111111
admin	1234
admin	12345
admin	123456
admin	54321
admin	7ujMko0admin
admin	admin
admin	admin1234
admin	meinsm

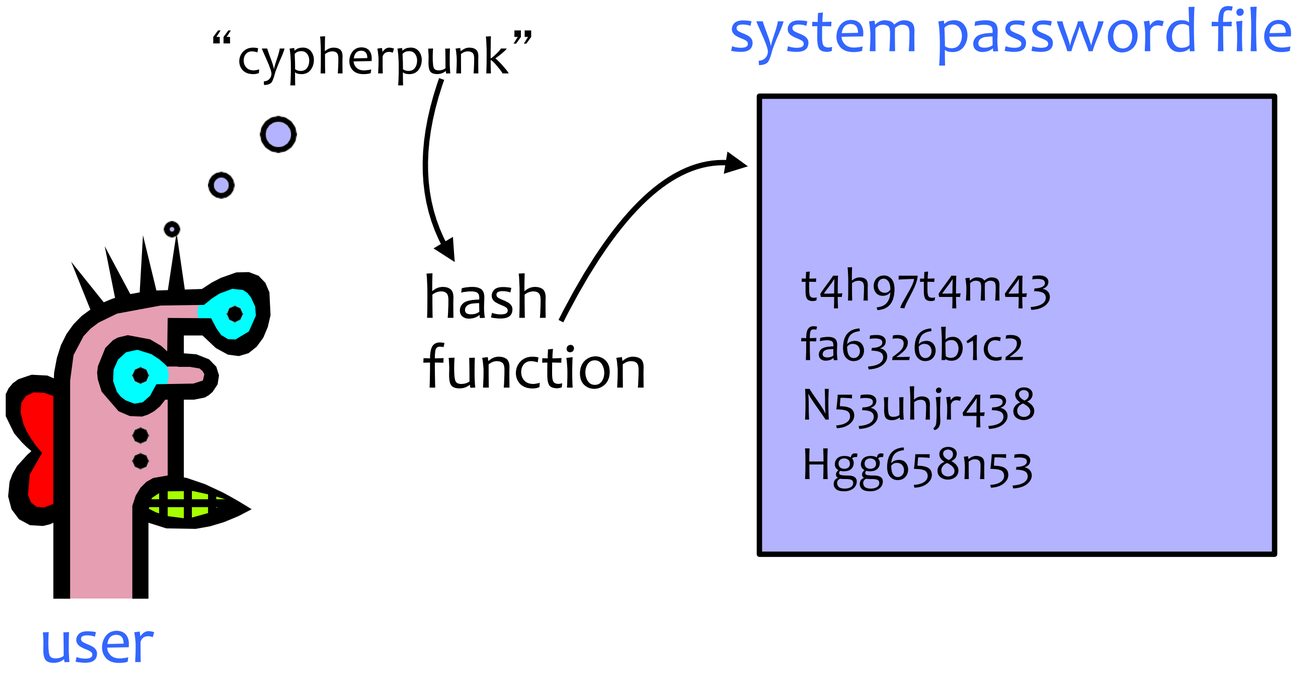
Over 500,000 IoT devices got hijacked into the Mirai botnet

Attacks targeting Password

- Offline dictionary attack
- Specific account attack
- Popular password attack
- Password guessing against single user
- Workstation hijacking
- Exploiting user mistakes
- Exploiting multiple password use
- Electronic monitoring

UNIX-Style Passwords

- How should we store passwords on a server?
  - In cleartext?
  - Encrypted?
  - Hashed?



Password Hashing

- Instead of user password, store H(password)
- When user enters password, compute its hash and compare with entry in password file
  - System does not store actual passwords!
  - System itself can't easily go from hash to password
    - Which would be possible if the passwords were encrypted
- Hash function H must have some properties
  - One-way: given H(password), hard to find password
    - No known algorithm better than trial and error

UNIX Password System

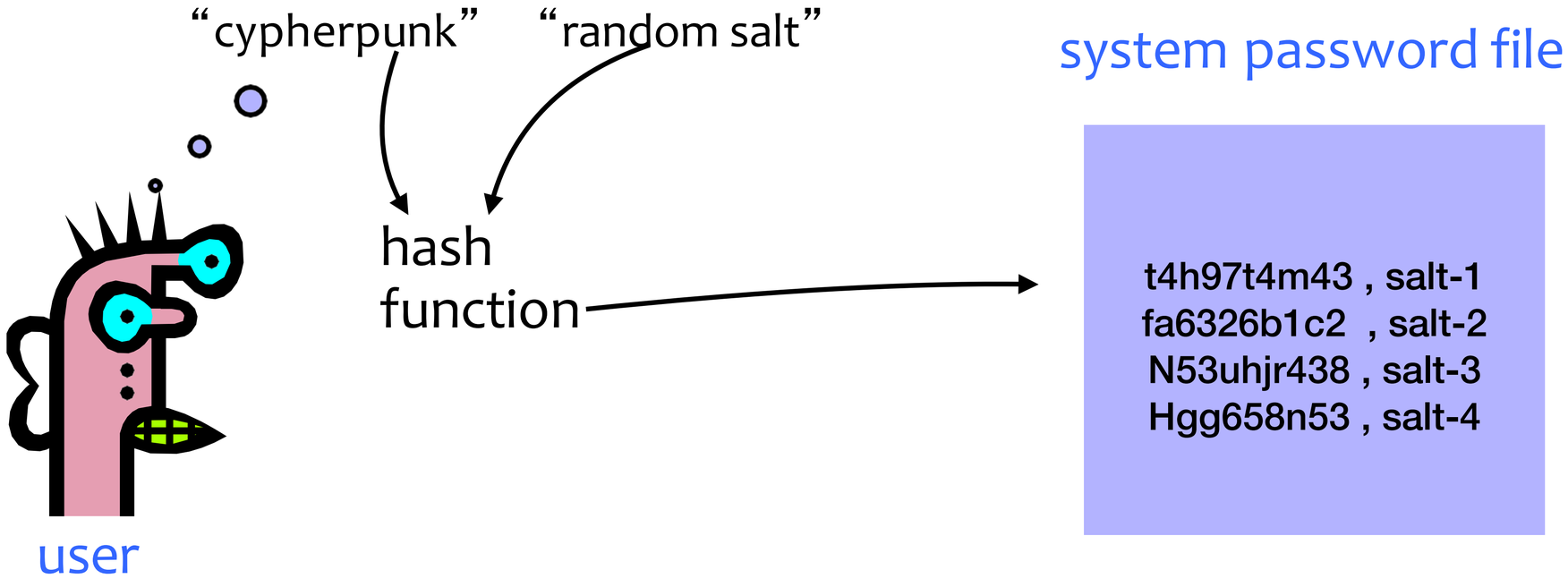
- Approach: Hash passwords
- Problem: passwords are not truly random
  - With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are 94^8
    6 quadrillion possible 8-character passwords ($\sim 2^{52}$)
  - **BUT:** Humans like to use dictionary words, human and pet names: 1 million common passwords

Dictionary Attack

- Dictionary attack is possible because many passwords come from a small dictionary
  - Attacker can pre-compute H(word) for every word in the dictionary - this only needs to be done once!
    - This is an offline attack
    - Once password file is obtained, cracking is instantaneous
  - Sophisticated password guessing tools are available
    - Take into account freq. of letters, password patterns, etc.

Salt + Hash

- How should we store passwords on a server?
  - Salt + Hash



Advantages of Salting

- Without salt, attacker can pre-compute hashes of all dictionary words once for <u>all</u> password entries
  - Same hash function on all UNIX machines
  - Identical passwords hash to identical values; one table of hash values can be used for all password files
- With salt, attacker must compute hashes of all dictionary words once for <u>each</u> password entry
  - With 12-bit random salt, same password can hash to 2^{12} different hash values
  - Attacker must try all dictionary words **for each salt value** in the password file
- Pepper: Secret salt (not stored in password file)

Other Password Security Risks

- Keystroke loggers
  - Hardware
  - Software (spyware)
- Shoulder surfing
- Same password at multiple sites
- Broken implementations
  - Example: TENEX timing attack
- Social engineering

Default Passwords

- Examples from Mitnick's "Art of Intrusion"
  - U.S. District Courthouse server: "public" / "public"
  - NY Times employee database: pwd = last 4 SSN digits
- Mirai IoT botnet
  - Weak and default passwords on routers and other devices

Weak Passwords

- RockYou hack
  - "Social gaming" company
  - Database with 32 million user passwords from partner social networks
  - Passwords stored in the clear
  - December 2009: entire database hacked using an SQL injection attack and posted on the Internet
  - One of many such examples!

Weak Passwords

- RockYou hack

Password Popularity – Top 20

Rank	Password	Number of Users with Password (absolute)
1	123456	290731
2	12345	79078
3	123456789	76790
4	Password	61958
5	iloveyou	51622
6	princess	35231
7	rockyou	22588
8	1234567	21726
9	12345678	20553
10	abc123	17542

Rank	Password	Number of Users with Password (absolute)
11	Nicole	17168
12	Daniel	16409
13	babygirl	16094
14	monkey	15294
15	Jessica	15162
16	Lovely	14950
17	michael	14898
18	Ashley	14329
19	654321	13984
20	Qwerty	13856

Password Policies

- Old recommendation:
  - 7 or 8 characters, at least 3 out of {digits, upper-case, lower-case, non-alphanumeric}, no dictionary words, change every 4 months, password may not be similar to previous 12 passwords

[Inglesant and Sasse, "The True Cost of Unusable Password Policies"]

Image from http://www.interactivetools.com/staff/dave/damons_office/

Password Policies

- Old recommendation:
  - 7 or 8 characters, at least 3 out of {digits, upper-case, lower-case, non-alphanumeric}, no dictionary words, change every 4 months, password may not be similar to previous 12 passwords
- **But** results in frustrated users and <u>less</u> security
  - Burdens of devising, learning, forgetting passwords
  - Users construct passwords insecurely, write them down
    - Can't use their favorite password construction techniques (small changes to old passwords, etc.)
  - Heavy password re-use across systems
  - (Password managers can help)

New NIST Guidelines ©

- Remove requirement to periodically change passwords
- Screen for commonly used passwords
- Allow copy-paste into password fields
- Allow but don't require arbitrary special characters
- Etc.

https://pages.nist.gov/800-63-3/sp800-63b.html

Recovering Passwords

Palin E-Mail Hacker Says It Was Easy

By Kim Zetter September 18, 2008 | 10:05 am | Categories: Elections, Hacks and Cracks

A p
obt
priv
sup
rev
too
Rep

after the password recovery was reenabled, it took seriously 45 mins on wikipedia and google to find the info, Birthday? 15 seconds on wikipedia, zip code? well she had always been from wasilla, and it only has 2 zip codes (thanks online postal service!)

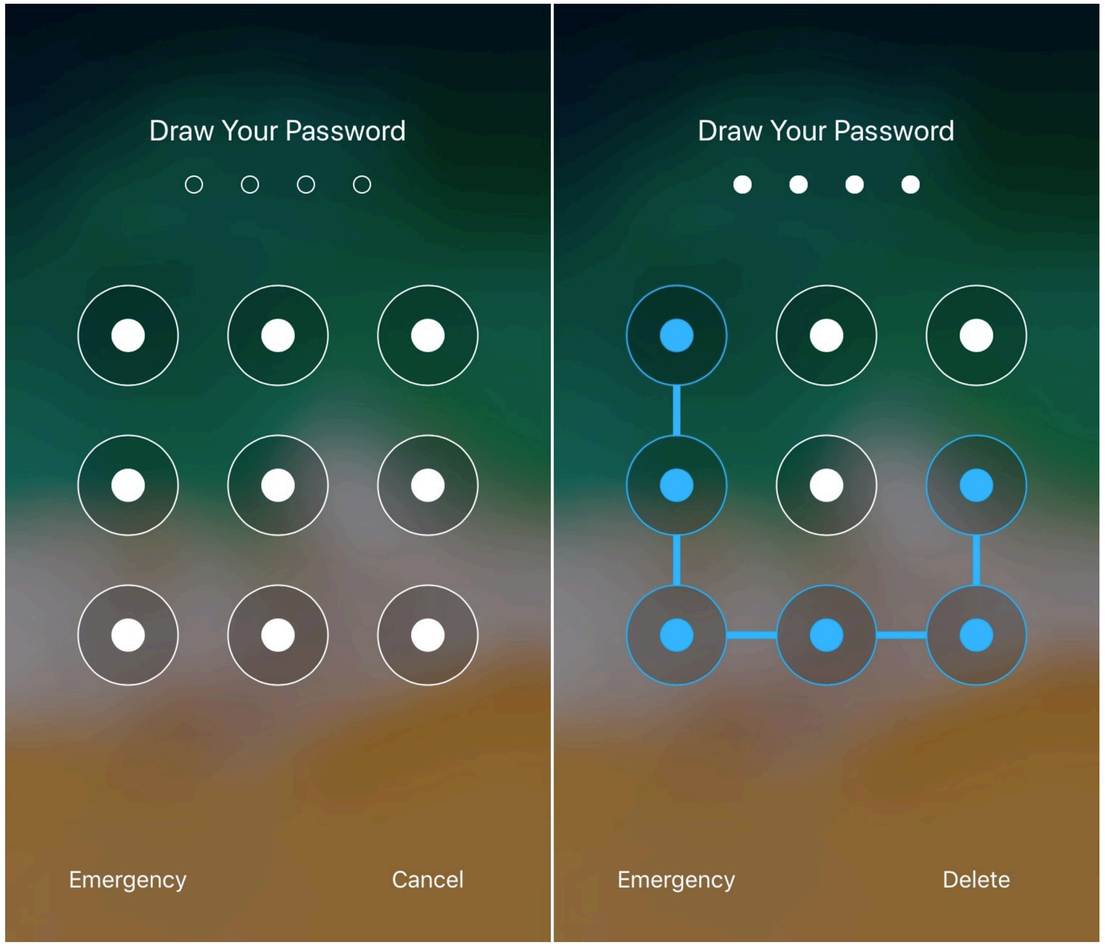
the second was somewhat harder, the question was "where did you meet your spouse?" did some research, and apparently she had eloped with mister palin after college, if youll look on some of the screenshits that I took and other fellow anon have so graciously put on photobucket you will see the google search for "palin eloped" or some such in one of the tabs.

I found out later though more research that they met at high school, so I did variations of that, high, high school, eventually hit on "Wasilla high" I promptly changed the password to popcorn and took a cold shower…

Improving(?) Passwords

- Add biometrics
  - For example, keystroke dynamics or voiceprint
- Graphical passwords
  - Goal: easier to remember? no need to write down?
- Password managers
  - Examples: LastPass, KeePass, built into browsers
  - Can have security vulnerabilities
- Two-factor authentication
  - Leverage phone (or other device) for authentication

Improving(?) Passwords



- Pattern password
- Any vulnerabilities?

Authentication Tokens

- Authentication leveraging what you possess
  - Password: what you know
- Techniques
  - One-Time Password (OTP)
  - Challenge-Response Pattern

RSA SecureID

YubiKey

Biometrics

- Authentication leveraging who you are
  - Password: what you know
  - Authentication Token: what you possess

Border control, security, mobile payment, time & attendance, fraud detection, civil ID

Biometrics: Understand the Application

Requirements: Embedded; high accuracy, throughput & usability; robust; fair; secure; ROI

https://www.bioenabletech.com/biometrics-atm https://www.bayometric.com/automobile-industry-adopting-automotive-biometrics

Outline

- Introduction
  - What is Biometrics?
  - Why Biometrics?
- Past
- Present
- Future

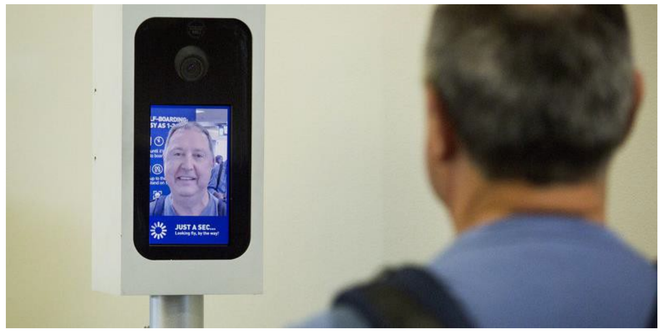
Biometrics

- ***"Automated recognition of individuals based on their behavioral and biological characteristics"*** *ISO/IEC JTC1 2382-37:2012*

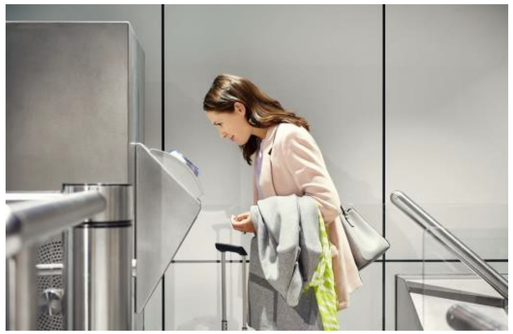
- ***Biometrics: bios (life) + metron (measure)*** *Morris,1875*

Most Popular Biometric Traits

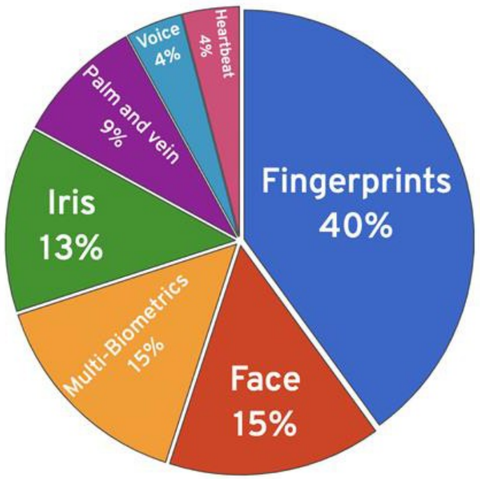
Incheon, South Korea: Smart Entry



Australia: SmartGate



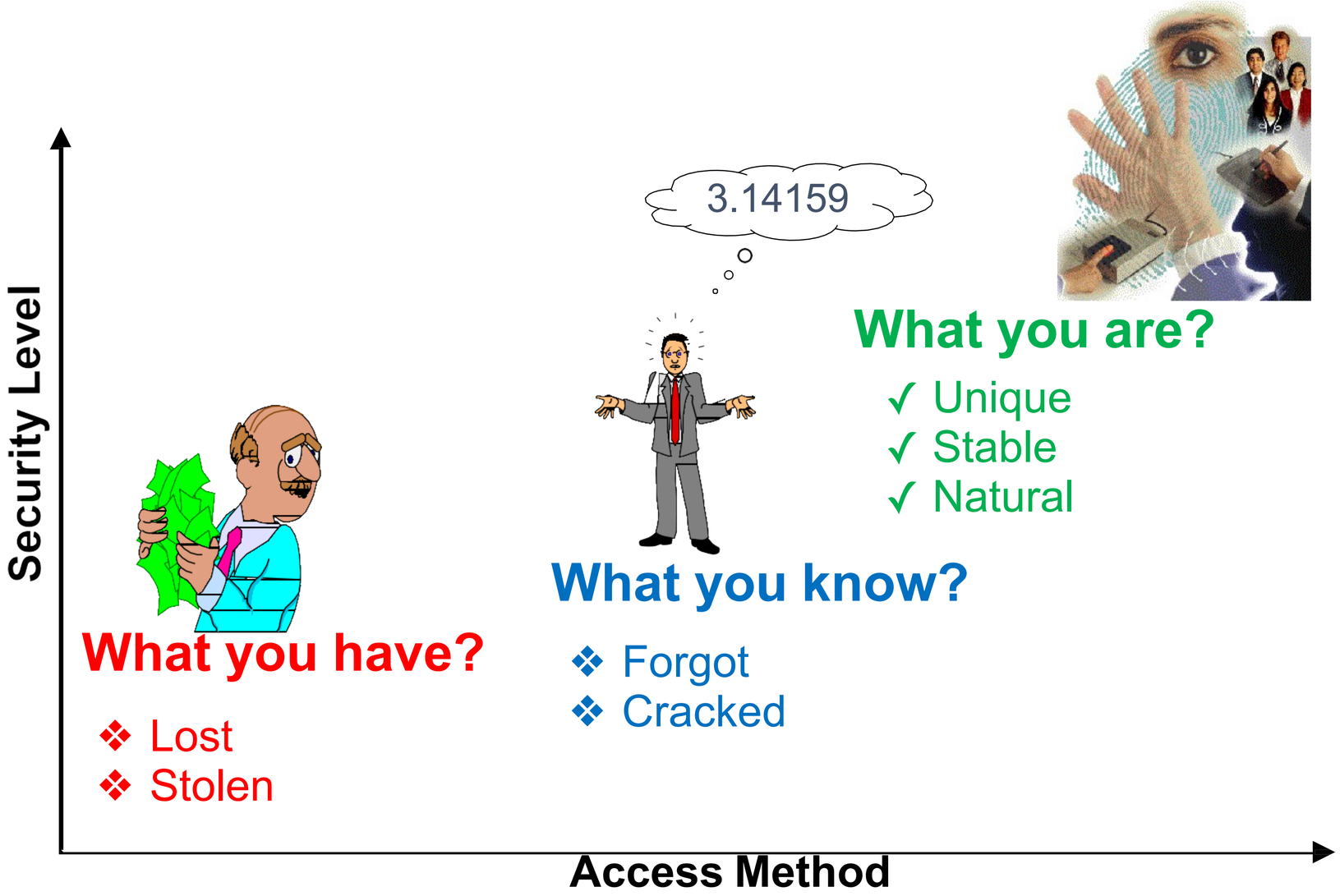
Amsterdam: Privium border passage



Uniqueness, persistence, legacy data, accuracy, fast search (1:N), NIST evaluations

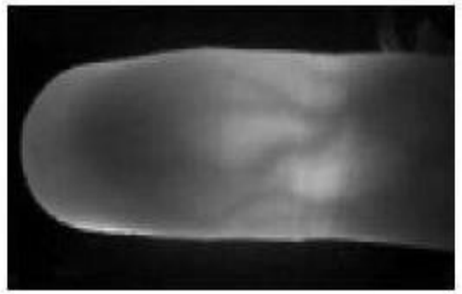
http://www.homestaykorea.com/?document_srl=73667&mid=bbs_koreainfo_news
https://tottnews.com/tag/smart-gates/
https://www.idemia.com/news/multi-biometrics-future-border-control-2016-04-21

Why Biometrics?



Multi-factor Authentication

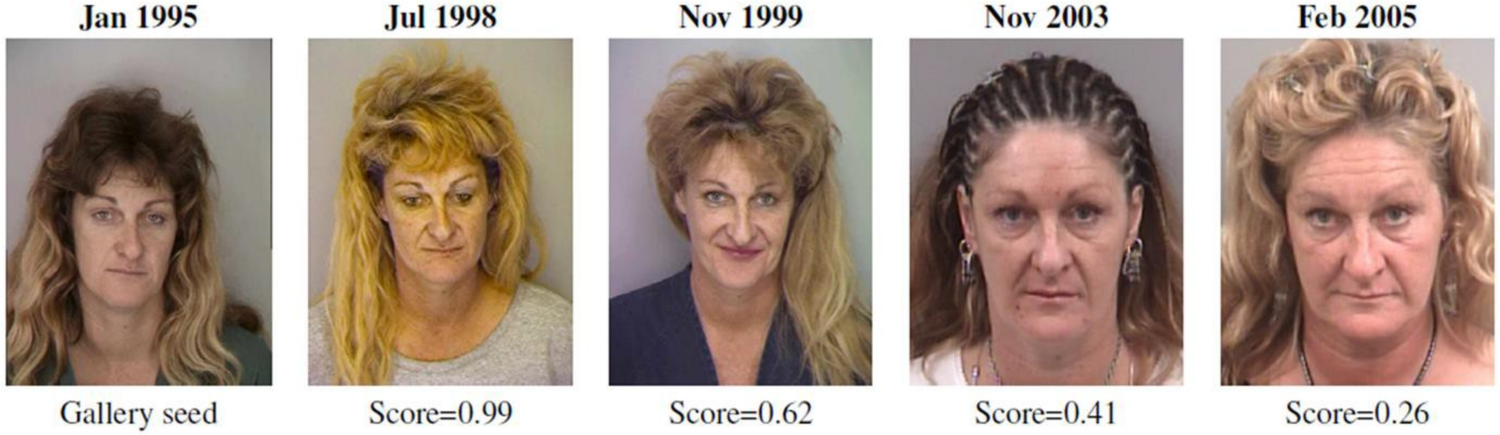
Finger Vein recognition



Higher security: ATM card + PIN + Finger vein

Fundamental Premise

- **Individuality:** Do different individuals have different biometric features? Does the same person always has the same features?
- **Permanence**: How does recognition accuracy change with time gap between enrolled and query samples?
- **Low intra-person variability and low inter-person similarity**



http://www.theguardian.com/theguardian/2010/dec/05/barack-obama-doppelganger-ilham-anas

Basic Issue: Compute Pair-wise Similarity

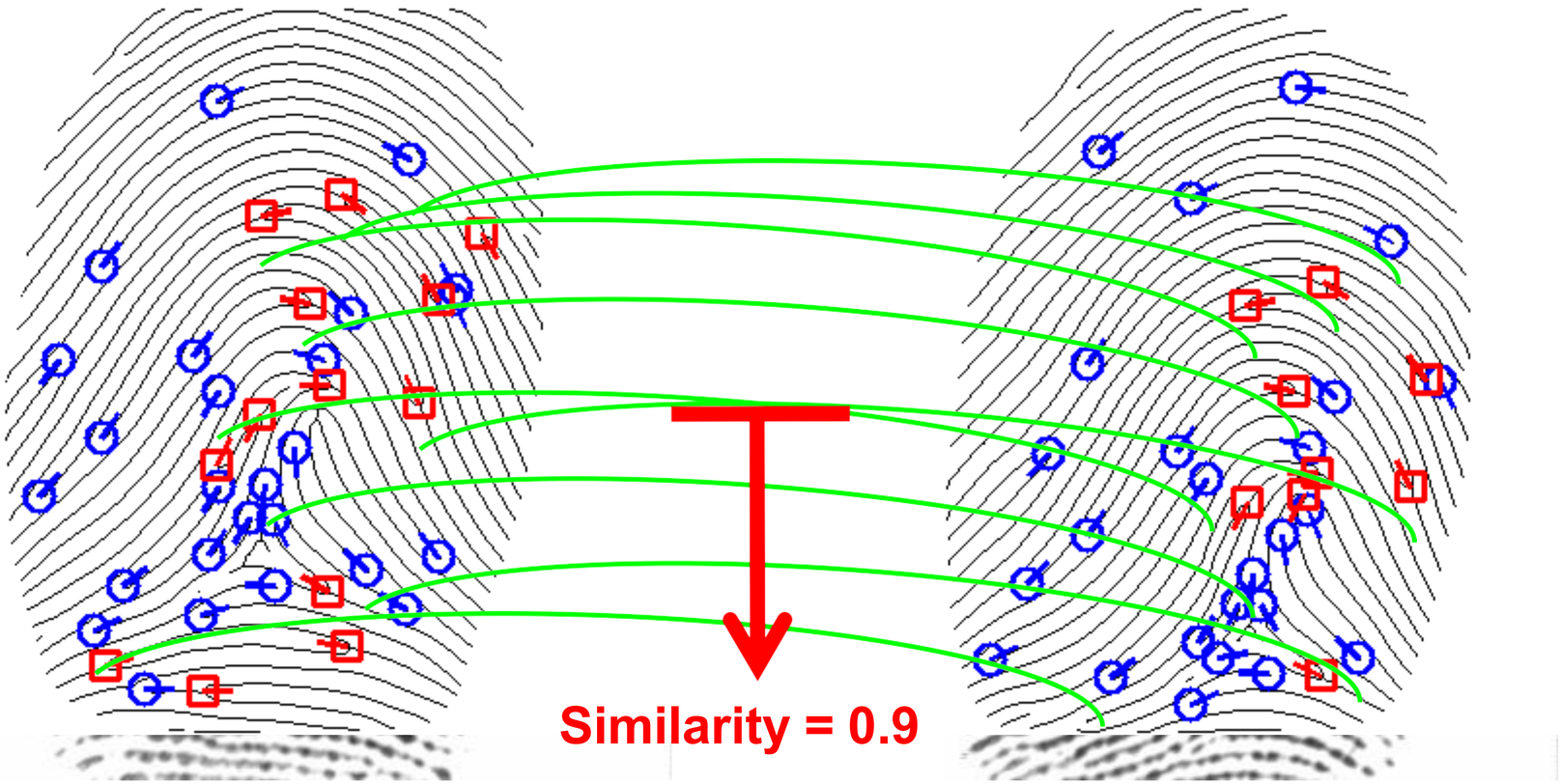
Query image

0.69 (Arcface)

Enrolled image

Representation (features) and similarity measure

Pair-wise Fingerprint Similarity



Query Print

Enrolled Fingerprint

- **Different #minutiae in query and enrolled images**
- **Unreliable minutiae in poor quality images**

Search (1:N Comparison)

Probe

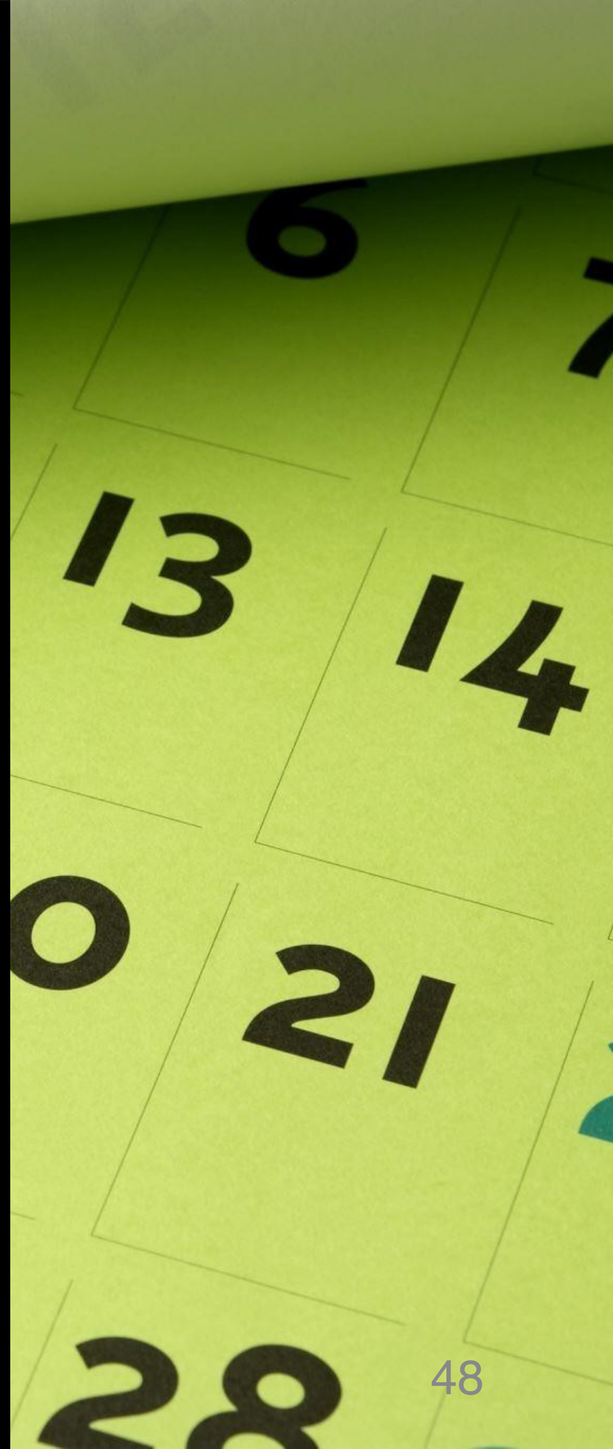
Who is he?

Gallery of size N

83% Similarity

Closed-set vs. Open-set

Past



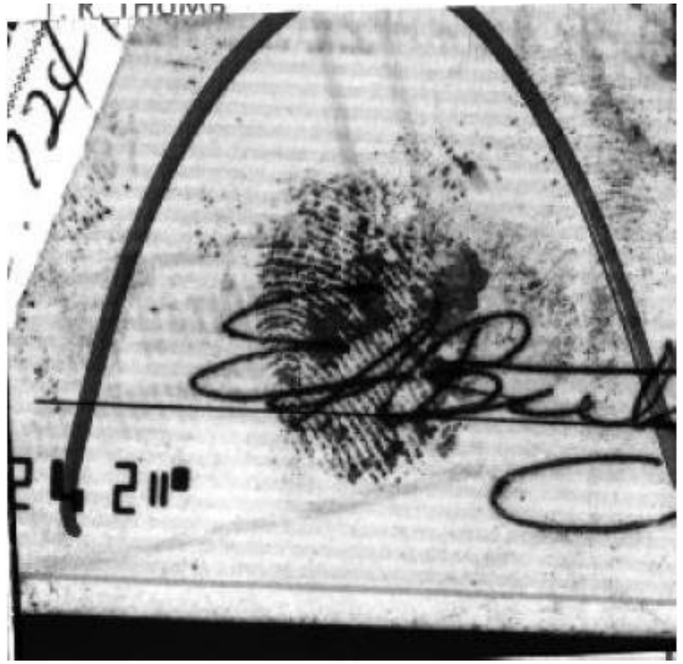
Habitual Criminal Act (1855)

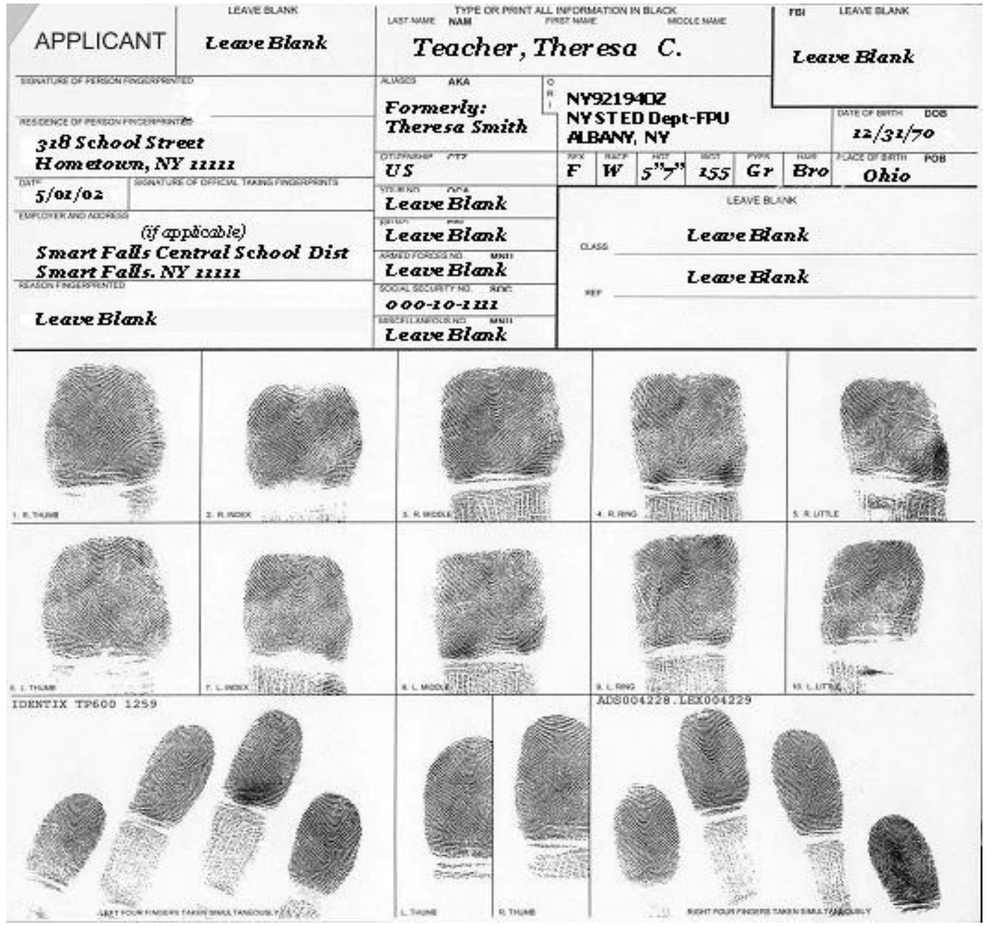
"What is wanted is a means of classifying the records of habitual criminal, such that as soon as the particulars of the personality of any prisoner (*whether description, measurements, marks, or photographs*) are received, it may be possible to ascertain readily, and with certainty, whether his case is in the register, and if so, who he is."

Scotland Yard (1905)

The headquarters of the Metropolitan Police, the territorial police force responsible for policing the 32 boroughs of London.

FBI (1924)





APPLICANT

LEAVE BLANK: Leave Blank

TYPE OR PRINT ALL INFORMATION IN BLACK

LAST NAME NAM / FIRST NAME / MIDDLE NAME: Teacher, Theresa C.

FBI LEAVE BLANK: Leave Blank

SIGNATURE OF PERSON FINGERPRINTED

RESIDENCE OF PERSON FINGERPRINTED: 318 School Street Hometown, NY 11111

DATE: 5/01/02

SIGNATURE OF OFFICIAL TAKING FINGERPRINTS

EMPLOYER AND ADDRESS: (if applicable) Smart Falls Central School Dist Smart Falls, NY 11111

REASON FINGERPRINTED: Leave Blank

ALIASES AKA: Formerly: Theresa Smith

ORI: NY9219402 NYSTED Dept-FPU ALBANY, NY

DATE OF BIRTH DOB: 12/31/70

CITIZENSHIP CTZ: US

SEX F | RACE W | HGT 5"7" | WGT 155 | EYES Gr | HAIR Bro | PLACE OF BIRTH POB: Ohio

SMT NO. SCA: Leave Blank

FBI NO.: Leave Blank

ARMED FORCES NO. MNU: Leave Blank

SOCIAL SECURITY NO. SOC: 000-10-1111

MISC FILE/REF NO. MNU: Leave Blank

LEAVE BLANK

CLASS: Leave Blank

REF: Leave Blank

1. R. THUMB 2. R. INDEX 3. R. MIDDLE 4. R. RING 5. R. LITTLE

6. L. THUMB 7. L. INDEX 8. L. MIDDLE 9. L. RING 10. L. LITTLE

IDENTIX TP600 1259

ADS004228.LEX004229

LEFT FOUR FINGERS TAKEN SIMULTANEOUSLY L. THUMB R. THUMB RIGHT FOUR FINGERS TAKEN SIMULTANEOUSLY

US Congress authorizes DOJ to collect fingerprints and arrest information

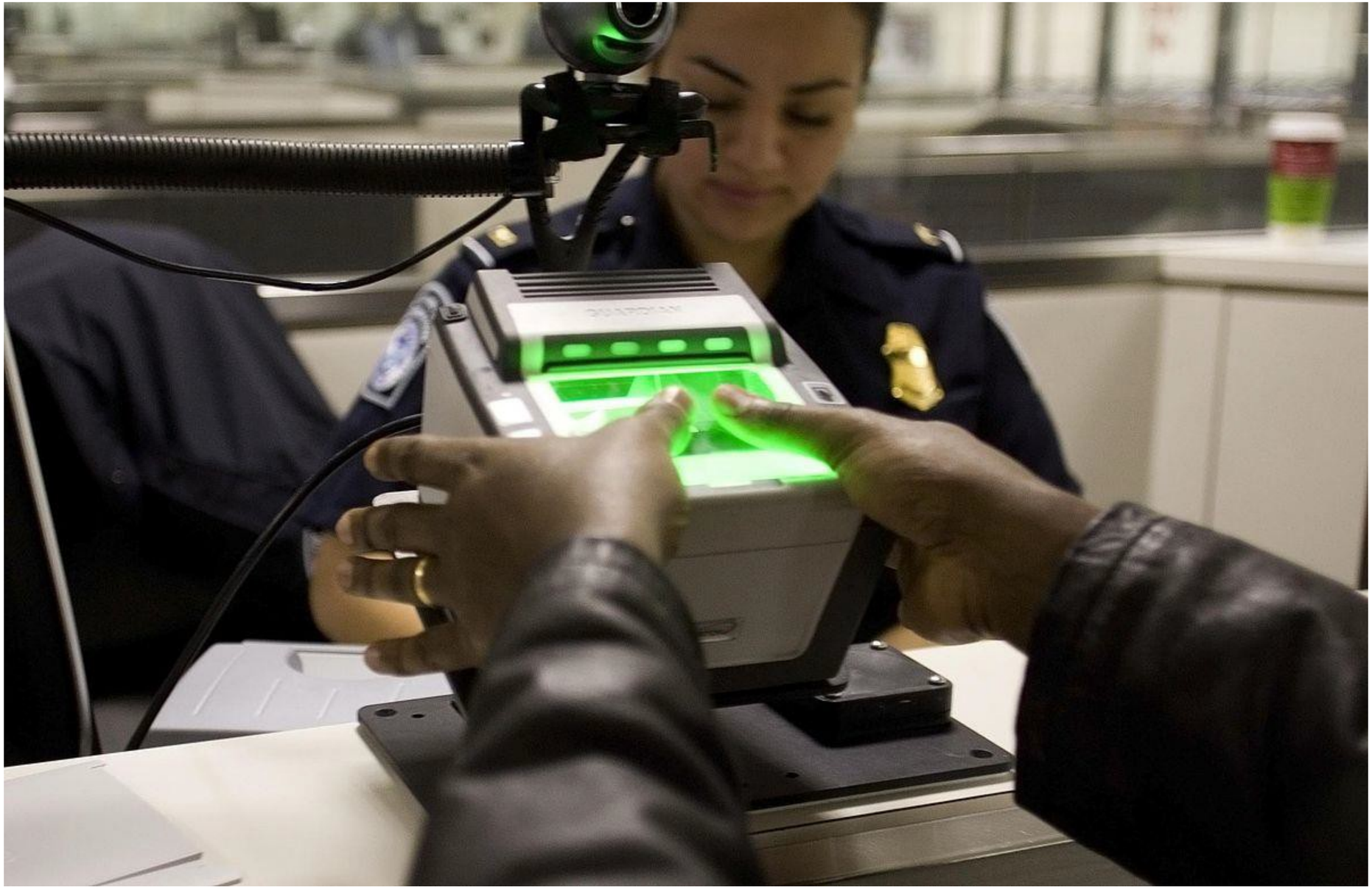
Identimate (1972)

First commercial use of biometrics

Present

9/11 Terrorist Attacks (2001)

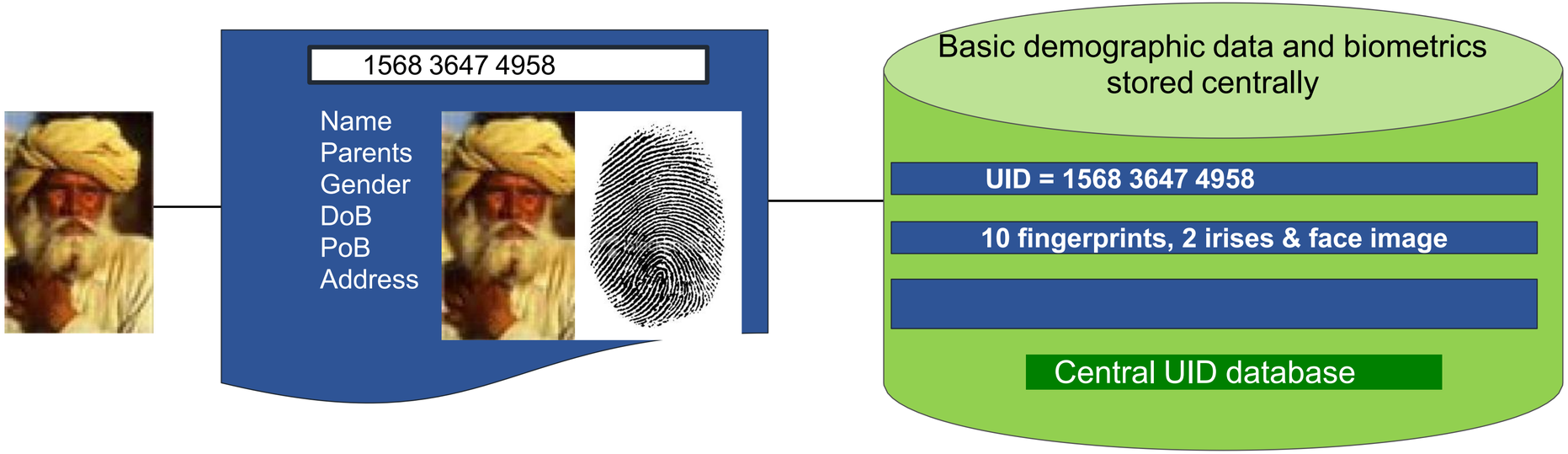
US-VISIT (2003)



Walt Disney Parks (2005)

Linking ticket to visitor's fingerprint

Aadhaar (2008)



"To empower residents of India with a unique identity and a digital platform to authenticate anytime, anywhere."

De-duplication (1: N search) based on ten fingers, 2 irises and face (N = 1.3 billion)

https://uidai.gov.in/

Authentication

https://uidai.gov.in/aadhaar_dashboard/auth_trend.php

Two-factor authentication: ~40 M authentications/day

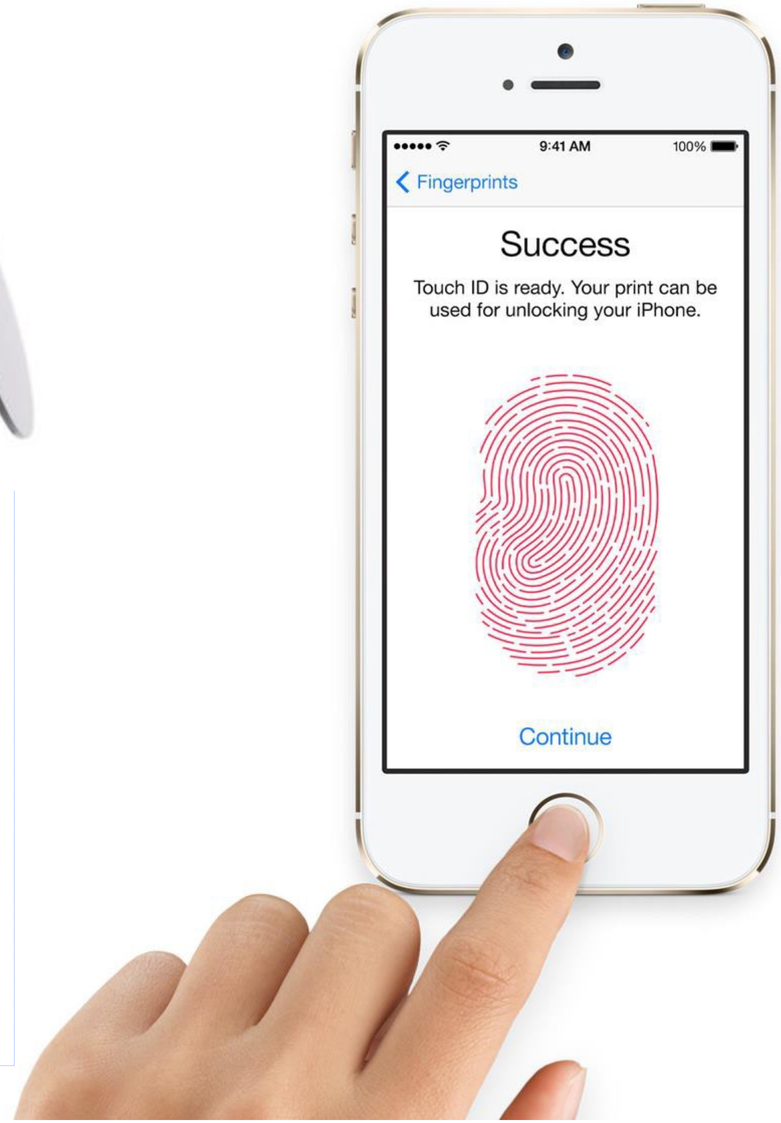
Apple Touch ID (2013)

Tactile Switch

Capacitive Single-Touch Sensor

Stainless Steel Detection Ring

Laser-Cut Sapphire Crystal



Apple Face ID (2017)

Face replaced fingerprint to increase the screen size

In-display Fingerprint Sensor (2018)

Entry/Exit system at Airports (2019)

Exit (leaving the country)

Entry (arriving from overseas)

https://www.cnn.com/travel/article/airports-facial-recognition/index.html (8th October 2019)

Amazon One (2020)

https://www.theverge.com/2020/10/1/21496673/amazon-one-palm-reading-vein-recognition-payments-identity-verification

Authentication Accuracy

Same Person?

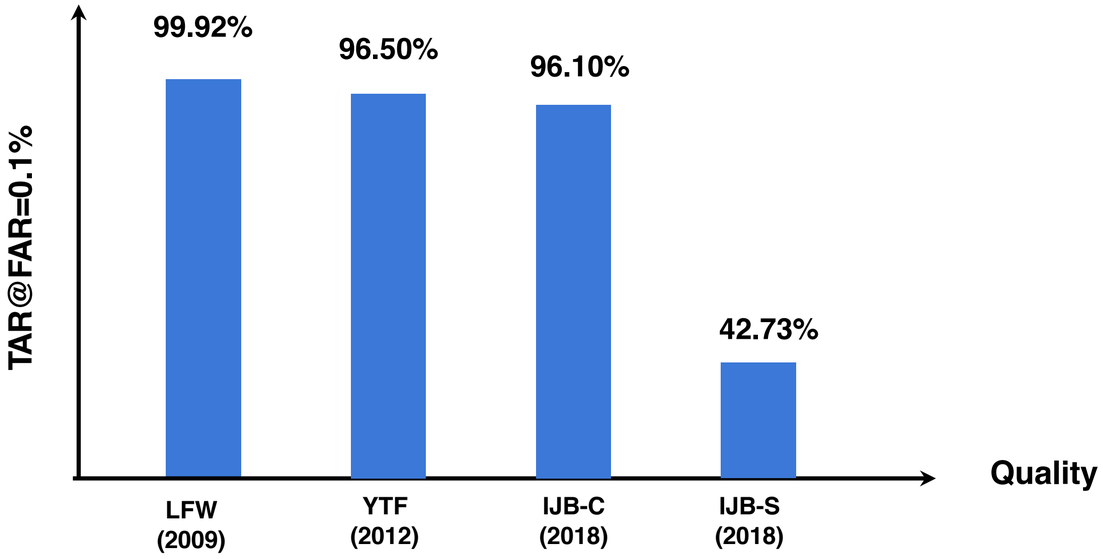
Fingerprint: TAR = 99.96% @ FAR = 0.01% (FVC-ongoing)

Iris: TAR = 99.82% @ FAR = 0.01% (NIST IREX II)

Face: TAR = 99.7% @ FAR = 0.1% (NIST FRVT 2010)

Image Quality vs. Accuracy

LFW (2009) YTF (2012) NIST IJB-C (2018) NIST IJB-S (2018)



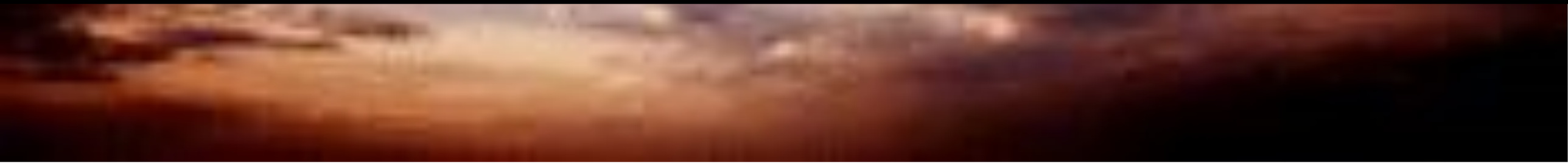
Identification (1:N) Accuracy

Probe

Gallery

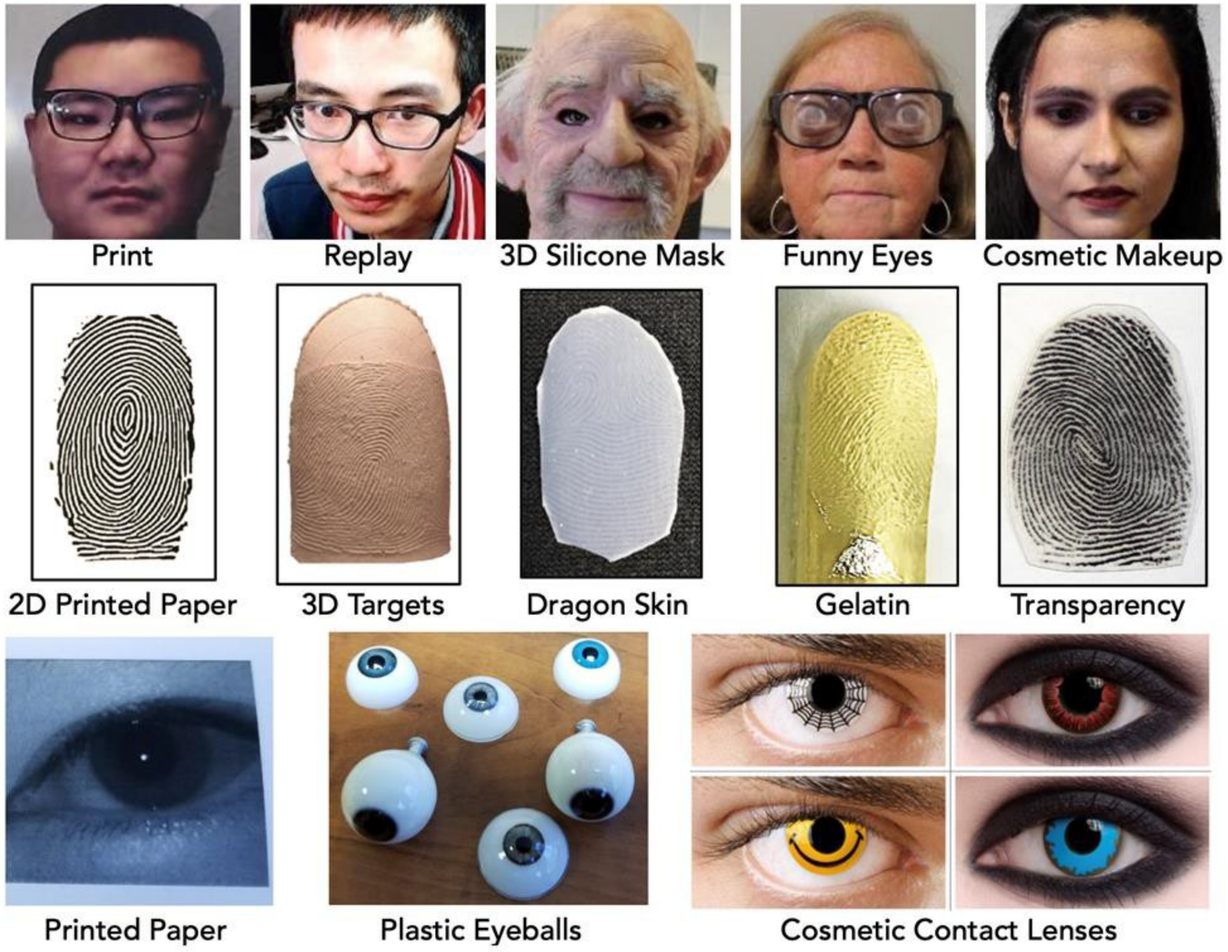
- **Face:** FNIR = 0.058 @ FPIR = 0.001 (12M gallery)
- **Fingerprint:** (Ten fingers): FNIR = 0.001 @ FPIR = 0.001 (5M gallery) Single finger: FNIR = 0.019 @ FPIR = 0.001 (100K gallery)
- **Iris:** (Both eyes): FNIR = 0.0059 @ FPIR = 0.001 (500K gallery)

1. NIST FRVT 1:N Identification: *https://pages.nist.gov/frvt/html/frvt1N.html*
2. NIST FpVTE: *https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.8034.pdf*
3. NIST IREX 10 Identification Track: *https://pages.nist.gov/IREX10/*



Spoofs (Minority Report, 2002)

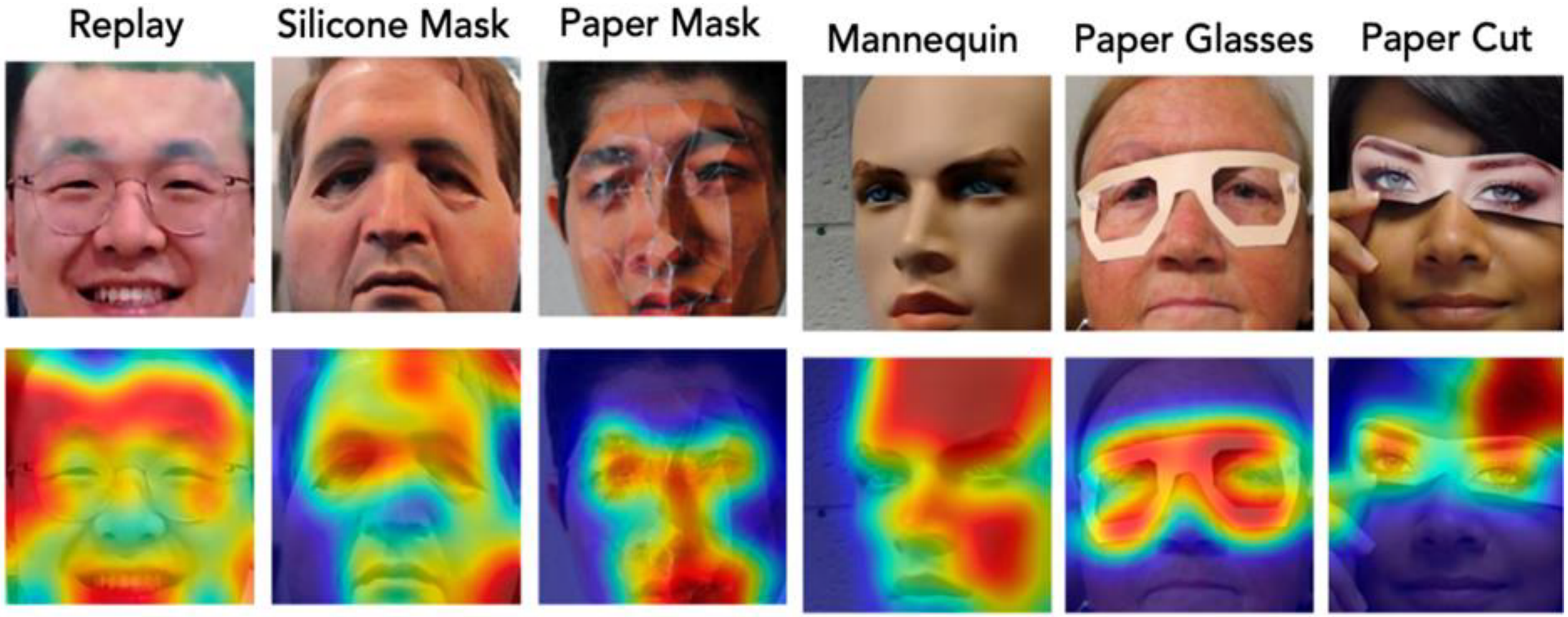
Physical Spoofs (Presentation Attacks)



Generalizaility to new spoofs, sensors & data collections

1. D. Deb and A. K. Jain, "Look Locally Infer Globally: A Generalizable Face Anti-Spoofing Approach," in IEEE TIFS, vol. 16, pp. 1143-1157, 2021, doi: 10.1109/TIFS.2020.3029879.
2. T. Chugh and A. K. Jain, “Fingerprint presentation attack detection: Generalization and efficiency,” in IEEE ICB, pp. 1–8, 2019.
3. S. Hoffman, R. Sharma, and A. Ross, “Convolutional neural networks for iris presentation attack detection: Toward cross-dataset and cross-sensor generalization,” in CVPRW, pp. 1620– 1628, 2018.

Explainability (Visualization)

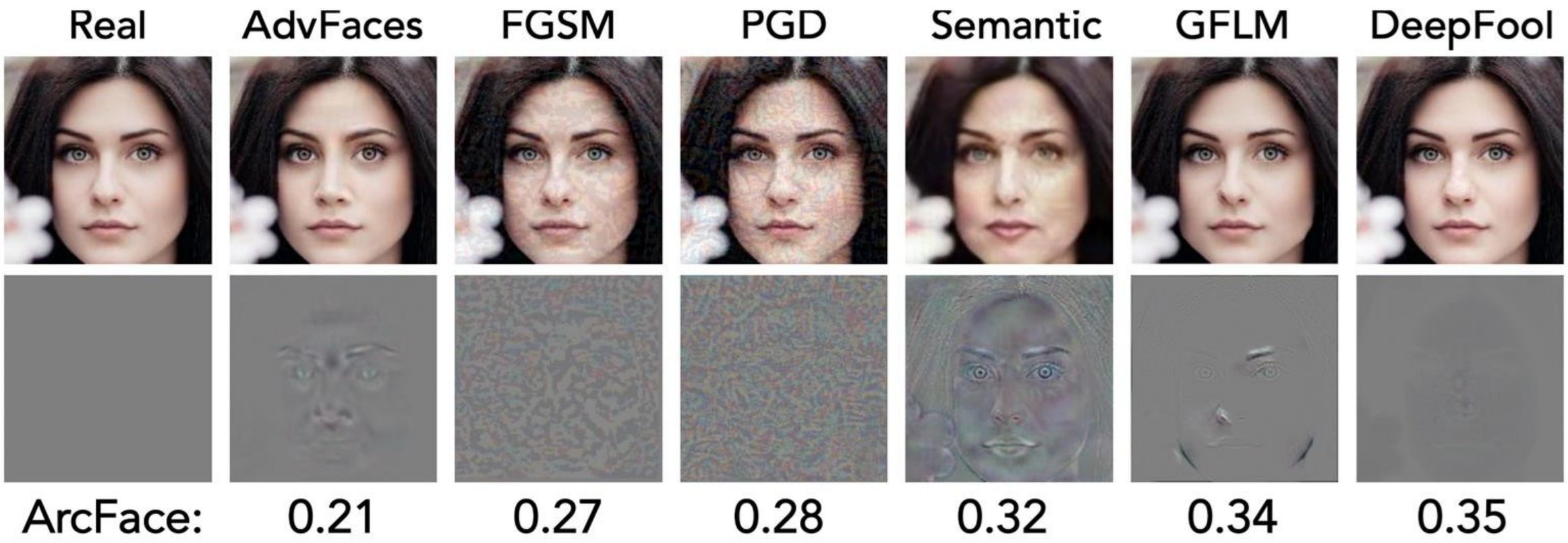


(a) Face Spoof Regions

Blue regions indicate bona fide regions while red regions denote spoofs

D. Deb and A. K. Jain, "Look locally infer globally: A generalizable face anti-spoofing approach," IEEE TIFS, vol. 16, 2020

Digital (Adversarial) Attacks



- **Top Row:** Adversarial faces synthesized via 6 adversarial attacks.
- **Bottom Row:** Corresponding adversarial perturbations (gray indicates no change from the input). Notice the diversity in the perturbations. ArcFace match scores between adversarial image and the unaltered gallery image are given below each image. A score above 0.36 indicates that two faces are of the same subject.

[1] D. Deb, J. Zhang and A. K. Jain, "AdvFaces: Adversarial Face Synthesis," 2020 IEEE IJCB 2020, pp. 1-10, doi: 10.1109/IJCB48548.2020.9304898.

Wrongfully Accused by an Algorithm

(a) Robert Williams

MICHIGAN STATE POLICE

INVESTIGATIVE LEAD REPORT

LAW ENFORCEMENT SENSITIVE

THIS DOCUMENT IS NOT A POSITIVE IDENTIFICATION. IT IS AN INVESTIGATIVE LEAD ONLY AND IS NOT PROBABLE CAUSE TO ARREST. FURTHER INVESTIGATION IS NEEDED TO DEVELOP PROBABLE CAUSE TO ARREST.

BID DIA Identifier: BID-39641-19	Requester: CA Yager, Rathe
Date Searched: 03/11/2019	Requesting Agency: Detroit Police Department
Digital Image Examiner: Jennifer Coulson	Case Number: 1810050167 File Class/Crime Type: 3000

Probe Image	Investigative Lead

(b) Investigative Lead Report

Face recognition wrongfully identified (a) Robert William when the CCTV frame in (b) was searched against 49 M gallery; Forensic experts did not follow the standard operating practice of manual examination of the candidate list

New York Times, "Wrongfully Accused by an Algorithm." https://www.nytimes.com/2020/06/24/technology/facial-recognition-arrest.html, 2020.

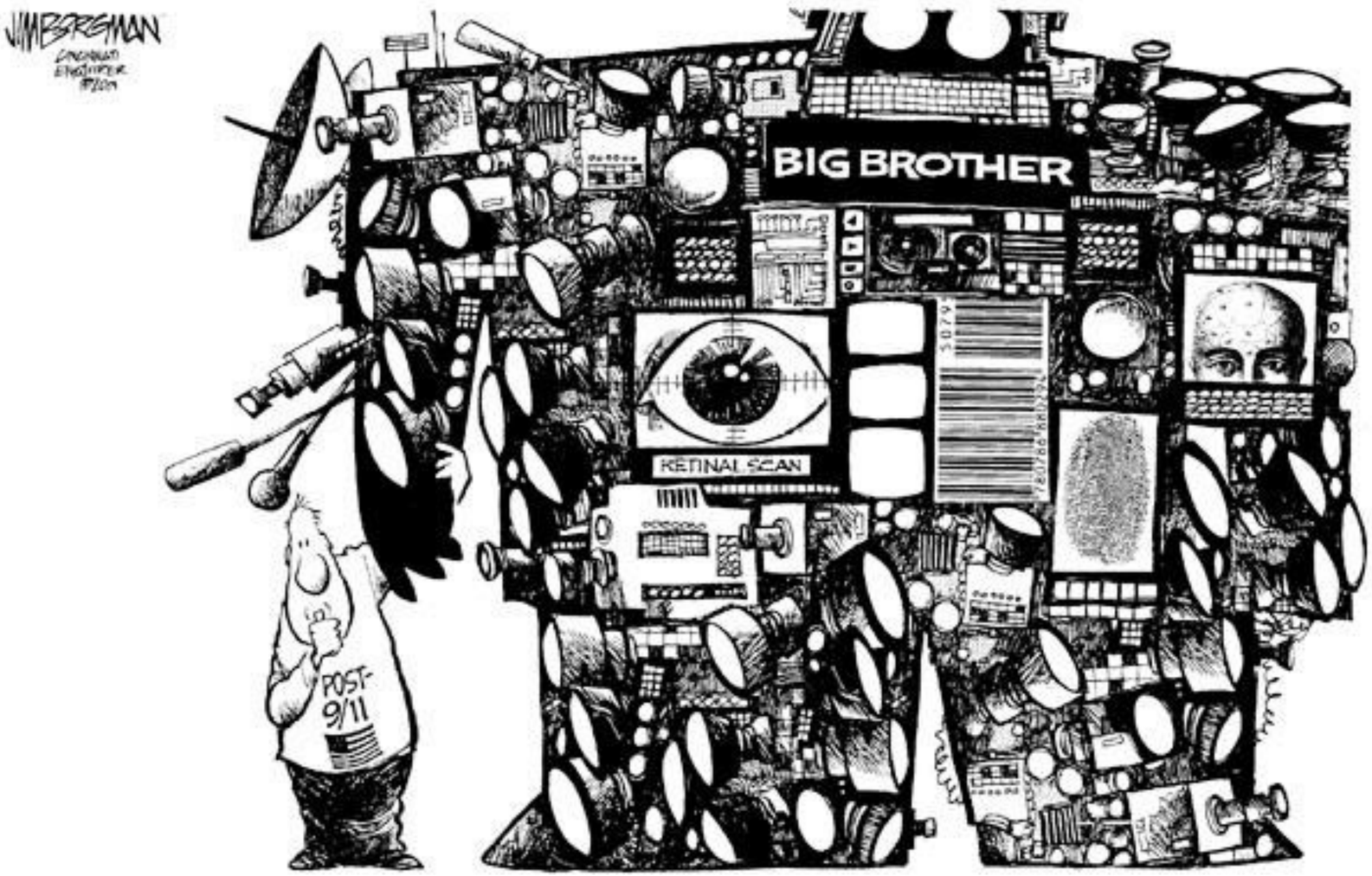
Demographic Bias

Race /Ethnicity	Sample Images	Verification Accuracy (%)
East Asian		93.72
Black		94.67
South Asian		93.98
Caucasian		96.18

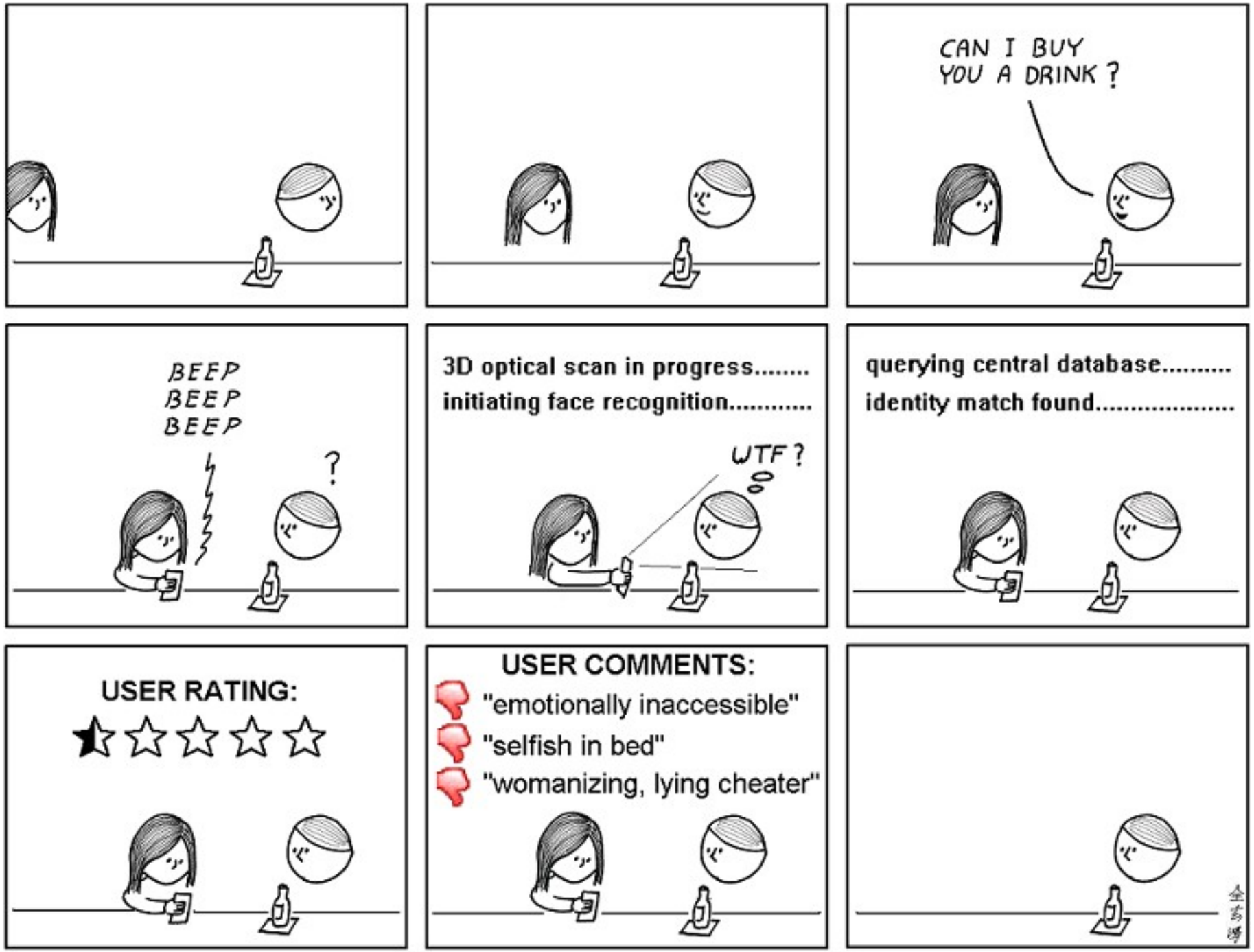
Face Verification performance by ArcFace [1] on each race/ethnicity cohort in RFW dataset [2]

1. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive an- gular margin loss for deep face recognition,” in CVPR, pp. 4690– 4699, 2019.
2. M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, “Racial faces in the wild: Reducing racial bias by information maximization adaptation network,” in ICCV, 2019.

Privacy

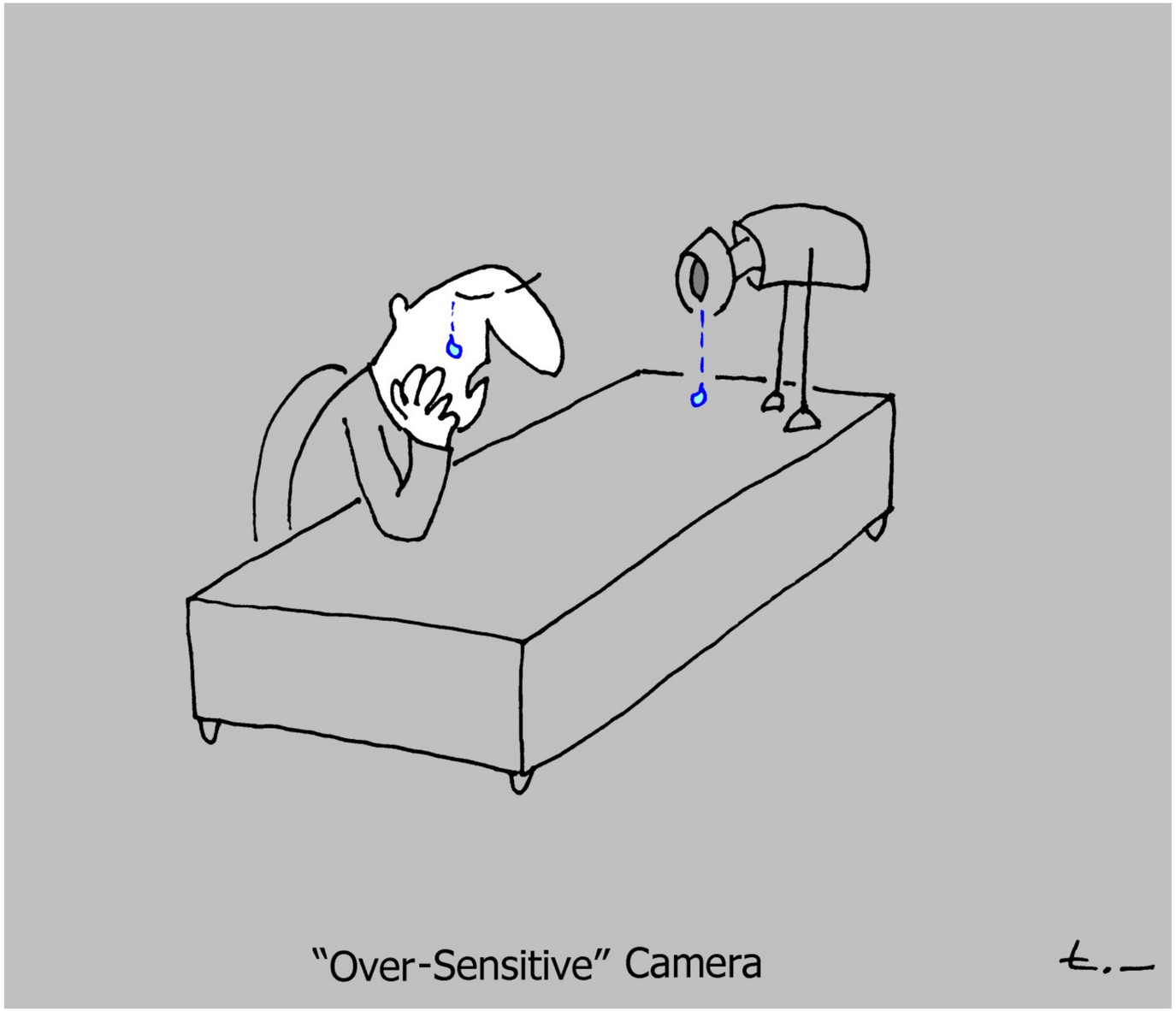


User consent, data retention and sharing policy, function creep


CAN I BUY YOU A DRINK ?
BEEP BEEP BEEP
?
3D optical scan in progress.....
initiating face recognition.....
WTF ?
querying central database.....
identity match found.....
USER RATING:
USER COMMENTS:
"emotionally inaccessible"
"selfish in bed"
"womanizing, lying cheater"

* PAT BEND

Biometrics and Emotion



Face biometrics should not be used to implement evaluations of employees based on emotion recognition, The Council of Europe has warned.

Targeted Ads

Digital billboard in London's Piccadilly Circus uses recognition technology to display targeted advertisements based on the make of passing cars, and the gender and age of pedestrians.

https://www.dezeen.com/2017/11/10/piccadilly-circus-digital-billboard-screen-targeted-advertisements-algorithm-news-technology/

Biometrics For Social Good

Face image and corresponding thumb-print (1,900 ppi) of a 1-week old infant

J. J. Engelsma, D. Deb, K. Cao, A. Bhatnagar, P. S. Sudhish, and A. K. Jain, "Infant-id: Fingerprints for global good," IEEE PAMI, 2021

Take Home Message

- **Biometrics: Ensures a *person is who he claims to be* and *not who he denies to be***
- **Drivers: Applications (Security, fraud, civil-ID, payment,…)**
- **Enablers: Advances in recog. algorithms, processor, memory & sensor (fingerprint technology embedded in a mobile costs ~$1)**
- **Challenge*: Recognize anyone, anywhere, anytime in real time***
- ***Concerns: Privacy, bias, impersonation,…….***
- ***Recourse: What if biometric system makes a mistake?***

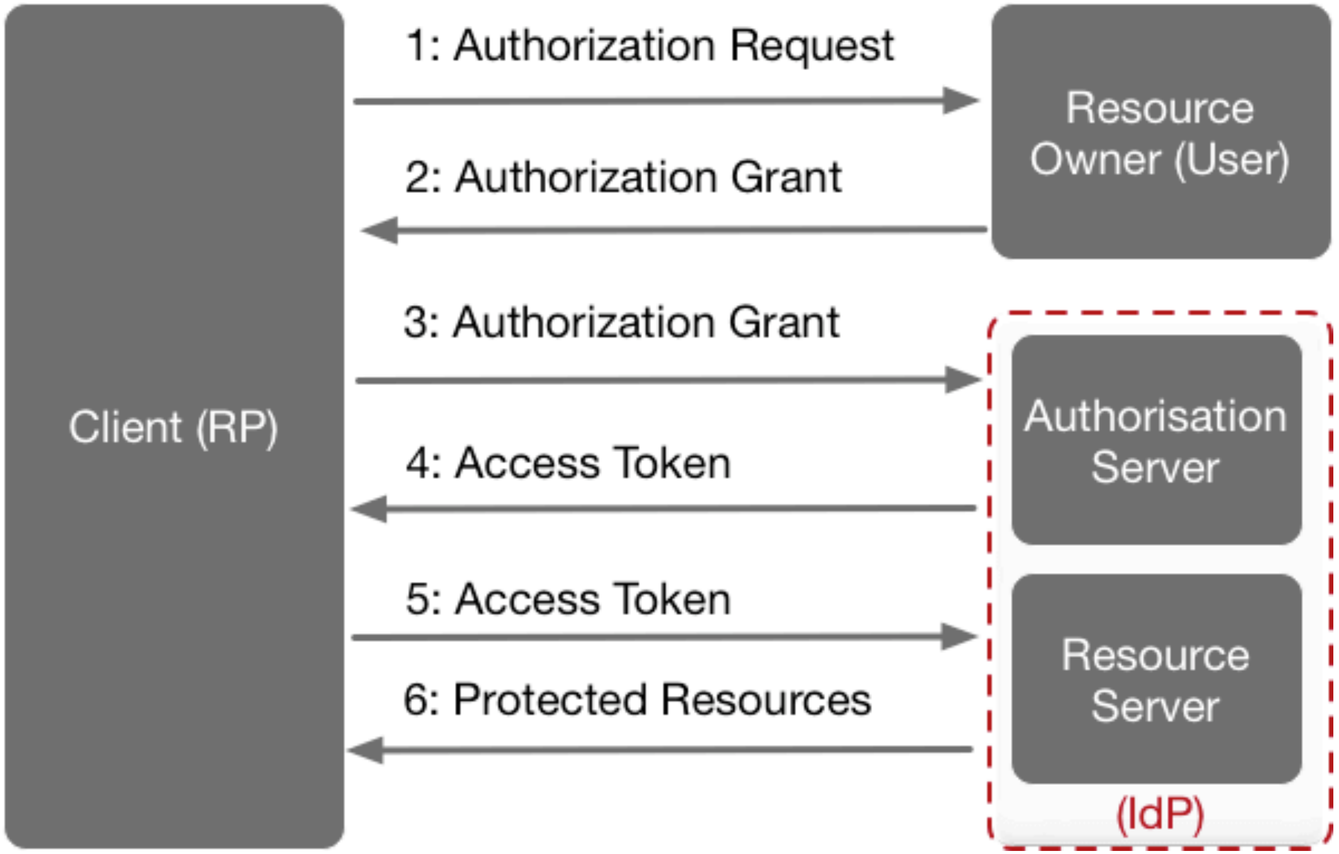
Outline

- Authentication Concepts
- Authentication Scenarios
- Authentication Primitives
- Authentication Protocols

OAuth/OpenID

- **OAuth is an open standard for access delegation**
- It allows Internet users (**resource owners**) to grant websites/apps (**clients**) the fine-grained access to their information on another website (**resource server**), without revealing the account password
  - Grant a Wechat app-in-app to access the username/profile image of your WeChat account
  - Login a third-party website using your Google account, i.e., allowing a third-party website to access your Google account information without knowing your account credentials

OAuth/OpenID



Public Key Infrastructure

Public Key Infrastructure

- In public key cryptography, we have assumed that everyone knows everyone else's public key. However, this is not easy to achieve.
  - How does Alice know that the public key she received is really Bob's public key?

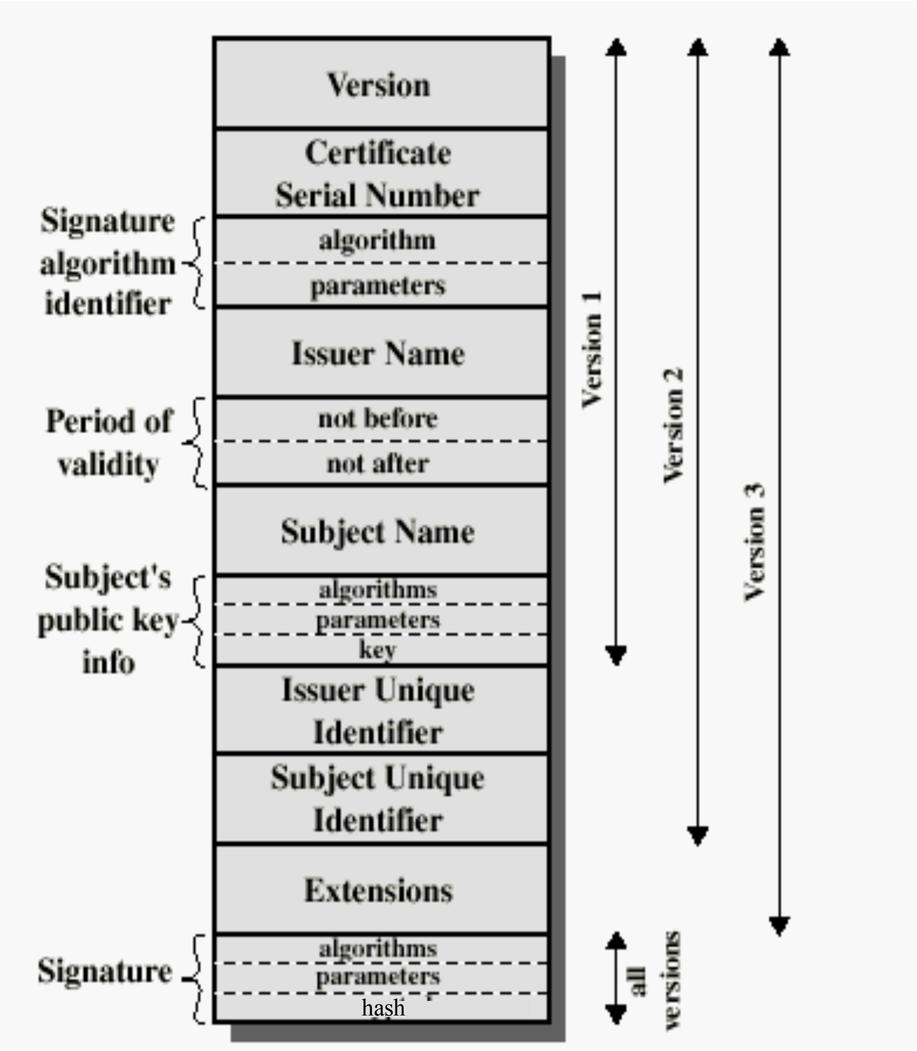
- A public key infrastructure (PKI) consists of all necessary components to securely distribute public keys.

- Basic idea: use certificate to distribute public keys. We need
  - Entities for creating certificates
  - Methods for evaluating certificates
  - Methods for revoking certificates

X.509 Authentication Service

- Internet standard (1988-2000)
- Specifies certificate format
  - X.509 certificates are used in IPsec and SSL/TLS
- Specifies certificate directory service
  - For retrieving other users' CA-certified public keys
- Specifies a set of authentication protocols
  - For proving identity using public-key signatures
- Does <u>not</u> specify crypto algorithms
  - Can use it with any digital signature scheme and hash function, but hashing is required before signing

X.509 Certificate



Important Things in X.509 Certificate

- Country: C=CN
- Organization: O=Nanjing University
- Organization Unit: OU=Department of Computer Science and Technology
- Common Name: CN=Alex Liu

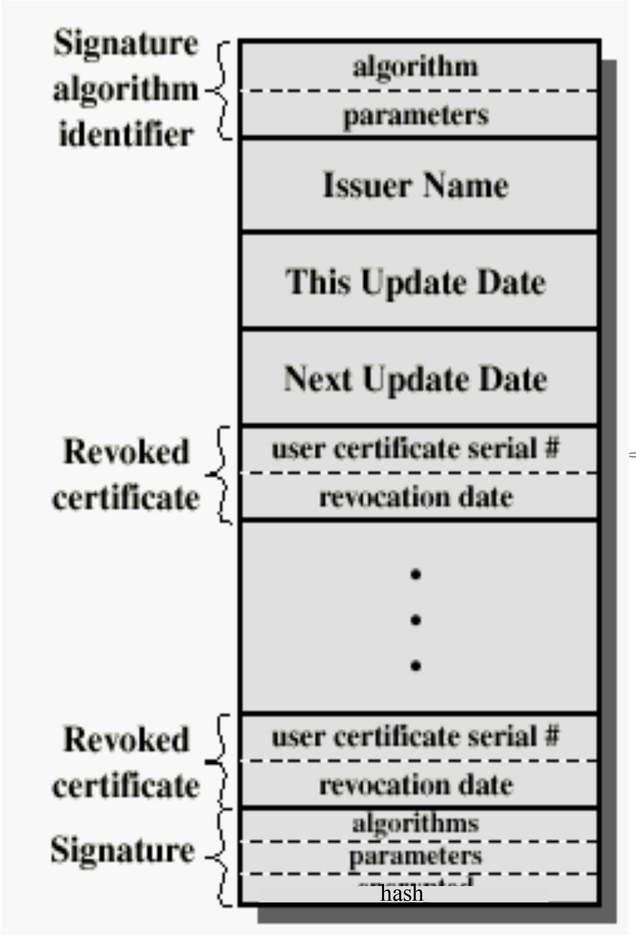
Certificate Revocation

- Revocation is very important
- Many valid reasons to revoke a certificate
  - Private key corresponding to the certified public key has been compromised
  - User stopped paying his certification fee to this CA and CA no longer wishes to certify him
  - CA's certificate has been compromised!
- Expiration is a form of revocation, too
  - Many deployed systems don't bother with revocation
  - Re-issuance of certificates is a big revenue source for certificate authorities

Certificate Revocation Mechanisms

- Online revocation service
  - When a certificate is presented, recipient goes to a special online service to verify whether it is still valid
    - Like a merchant dialing up the credit card processor
- Certificate revocation list (CRL)
  - CA periodically issues a signed list of revoked certificates
    - In 1970s, credit card companies used to issue thick books of canceled credit card numbers
  - Can issue a “delta CRL” containing only updates

X.509 Certificate Revocation List



Because certificate serial numbers must be unique within each CA, this is enough to identify the certificate

CRL Problems

- CRLs don't work in reality
  - Blacklist approach was abandoned by credit card vendors 20 years ago because it didn't work properly
- CRLs mirror credit card blacklist problems
  - Not issued frequently enough to be effective against an attacker
  - Expensive to distribute
  - Vulnerable to simple DOS attacks
    - Attacker can prevent revocation by blocking CRL delivery
- CRLs add further problems of their own
  - Can contain retroactive invalidity dates
  - CRL issued right now can indicate that a cert was invalid last week
    - Checking that something was valid at time *t isn't sufficient* to establish validity
    - Back-dated CRL can appear at any point in the future
  - Destroys the entire concept of nonrepudiation

CRL Problems (continued)

- Revoking self-signed certificates is hairy
  - Cert revokes itself
  - Applications may
    - Accept the CRL as valid and revoke the certificate
    - Reject the CRL as invalid since it was signed with a revoked certificate
    - Crash
- CRL Distribution Problems
  - CRLs have a fixed validity period
    - Valid from *issue date to expiry date*
  - At *expiry date, all relying parties connect to the CA to fetch the* new CRL
    - Massive peak loads when a CRL expires (DDOS attack)
  - Issuing CRLs to provide timely revocation exacerbates the problem
  - 10M clients download a 1MB CRL issued once a minute =~150GB/s traffic
  - Even per-minute CRLs aren't timely enough for high-value transactions with interest calculated by the minute

CRL Alternative: Online Certification Status Protocol

- OCSP is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate.
- Described in RFC 2560.
- The "request/response" nature of these messages leads to OCSP servers being termed OCSP responders.
- An OCSP responder may return a signed response signifying that the certificate specified in the request is 'good', 'revoked' or 'unknown'. If it cannot process the request, it may return an error code.
- The OCSP request format supports additional extensions. This enables extensive customization to a particular PKI scheme.
- How does OCSP prevents replay attacks?
  - OCSP can be resistant to replay attacks, where a signed, 'good' response is captured by a malicious intermediary and replayed to the client at a later date after the subject certificate may have been revoked. OCSP overcomes this by allowing a nonce to be included in the request that must be included in the corresponding response.

OCSP Example

- 1. Alice and Bob have public key certificates issued by Ivan, the Certificate Authority (CA).
- 2. Alice wishes to perform a transaction with Bob and sends him her public key certificate.
- 3. Bob, concerned that Alice's private key may have been compromised, creates an 'OCSP request' that contains Alice's certificate serial number and sends it to Ivan.
- 4. Ivan's OCSP responder looks up the revocation status of Alice's certificate (using the certificate serial number Bob informed) in his own CA database. If Alice's private key had been compromised, this is the only trusted location at which the fact would be recorded.
- 5. Ivan's OCSP responder confirms that Alice's certificate is still OK, and returns a signed, successful 'OCSP response' to Bob.
- 6. Bob cryptographically verifies the signed response (He has Ivan's public key on-hand – Ivan is a trusted responder) and ensures that it was produced recently.
- 7. Bob completes the transaction with Alice.

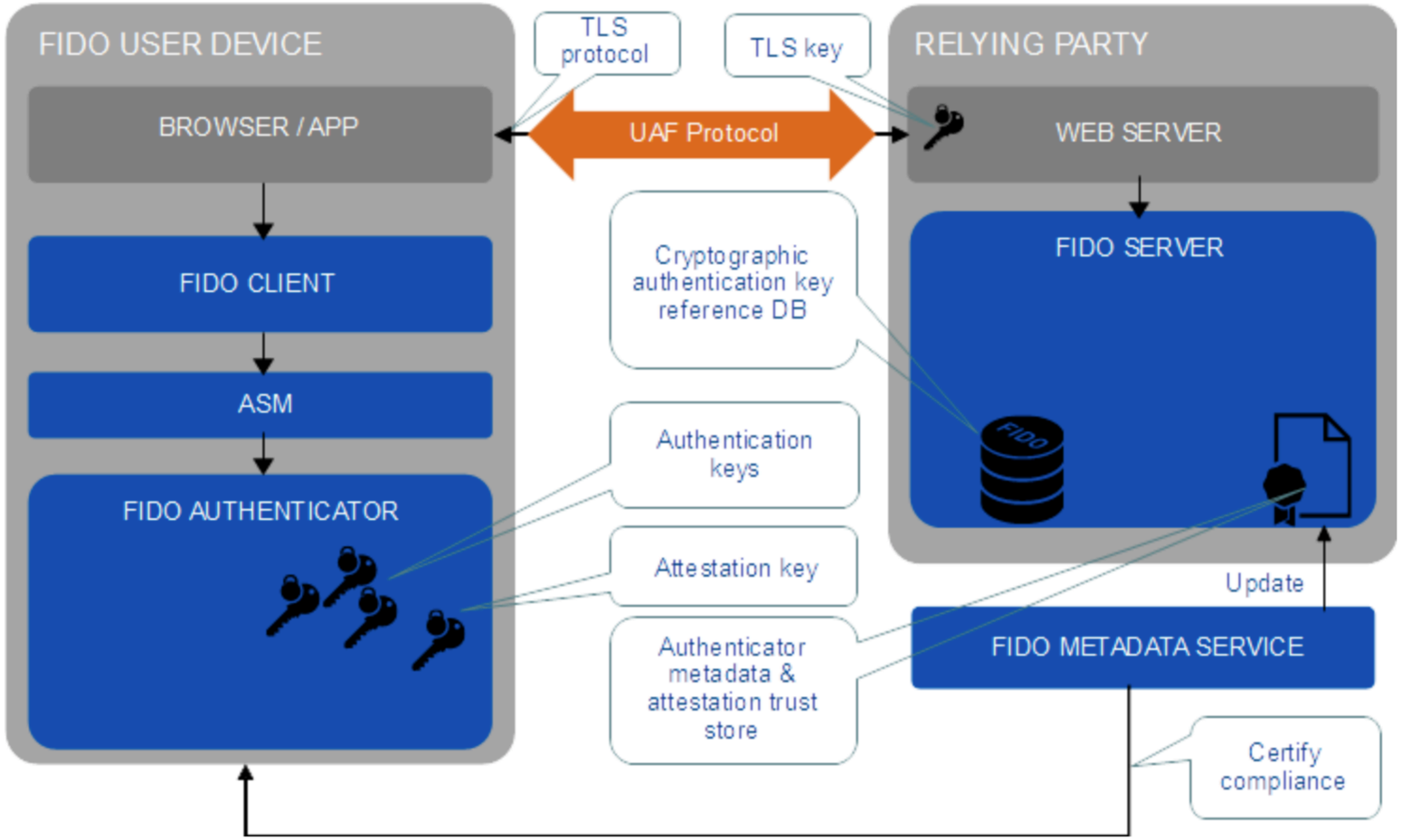
Revocation/Status Checking in the Real World

- In practice, revocation checking is turned off in user software
  - Because it slows everything down a lot
- CRLs are useful in special-case situations where there exists a statutory or contractual obligation to use them
  - Relying party needs to be able to claim CRL use for due diligence purposes or to avoid liability
- CA key compromise
  - Sun handled revocation of their CA key via posts to mailing lists and newsgroups

FIDO UAF

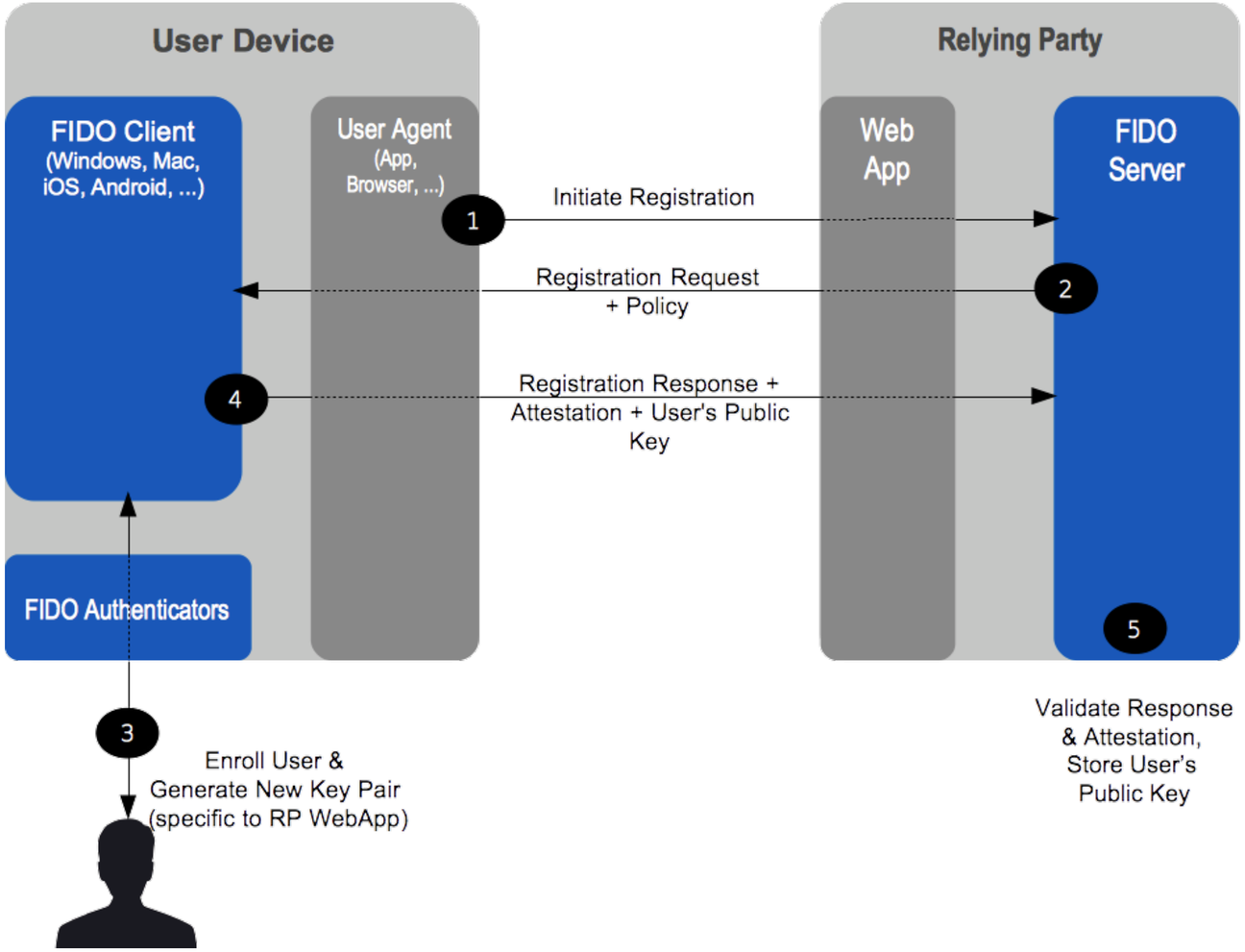
- A strong authentication framework
- Enables online services and websites, to transparently leverage native security features of end-user computing devices, for strong user authentication.
- More details can be found in FIDO UAF's official specifications
- A further development is FIDO2 (webauthn), aiming to get rid of password:
  - Use of Rivest-Shamir-Adelman (RSA) encryption public and private key pairs for the digital signature scheme
  - Use of platform authenticators (such as the Trusted Platform Module (TPM) on a desktop/laptop computer, or a Secure Element built into a mobile phone) to generate and protect private keys
  - Use of external (or roaming) authenticators such as smart cards, Subscriber Identity Module (SIM) cards, or USB-based cryptographic hardware with HID and BLE transport support

FIDO UAF



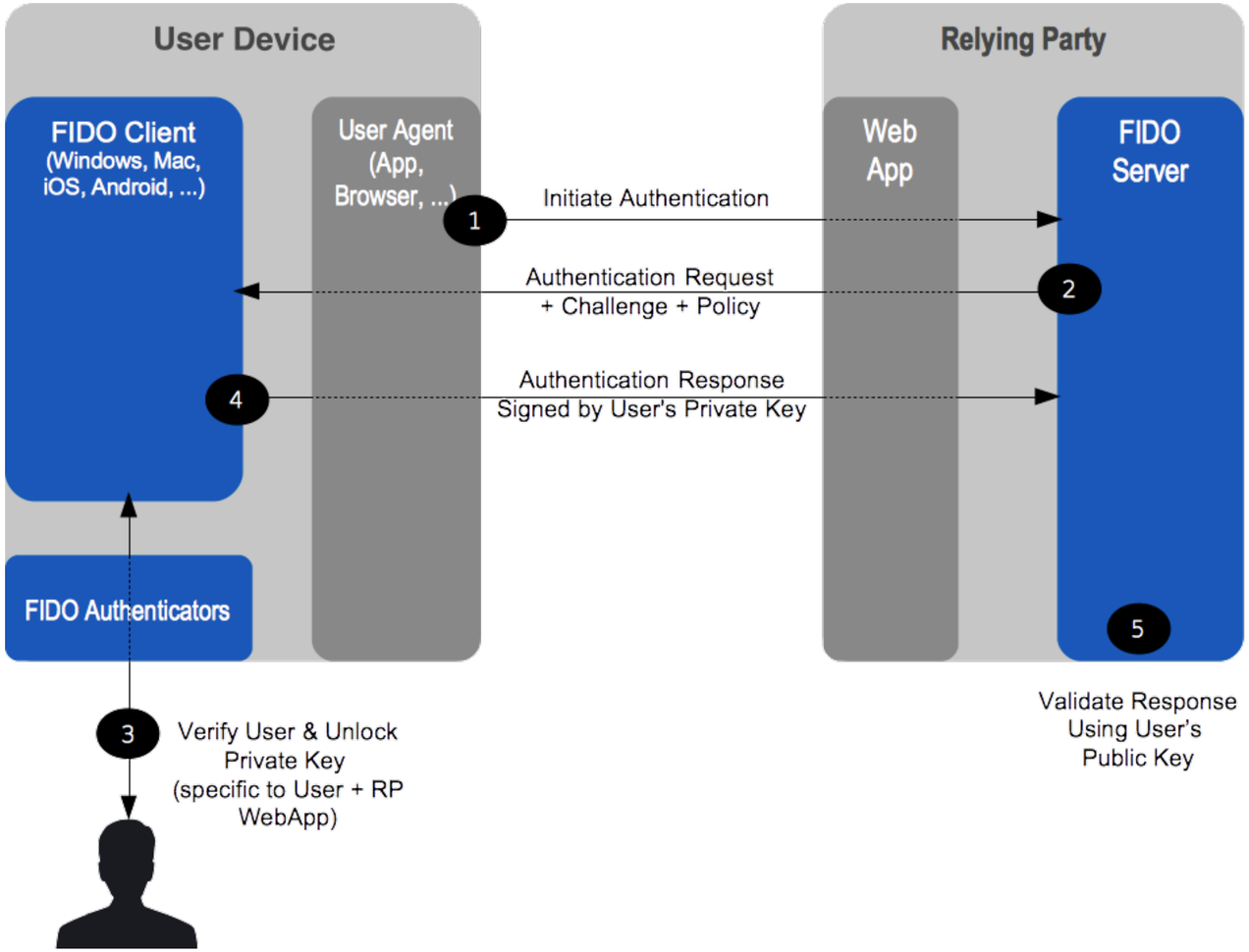
FIDO UAF Framework

FIDO UAF



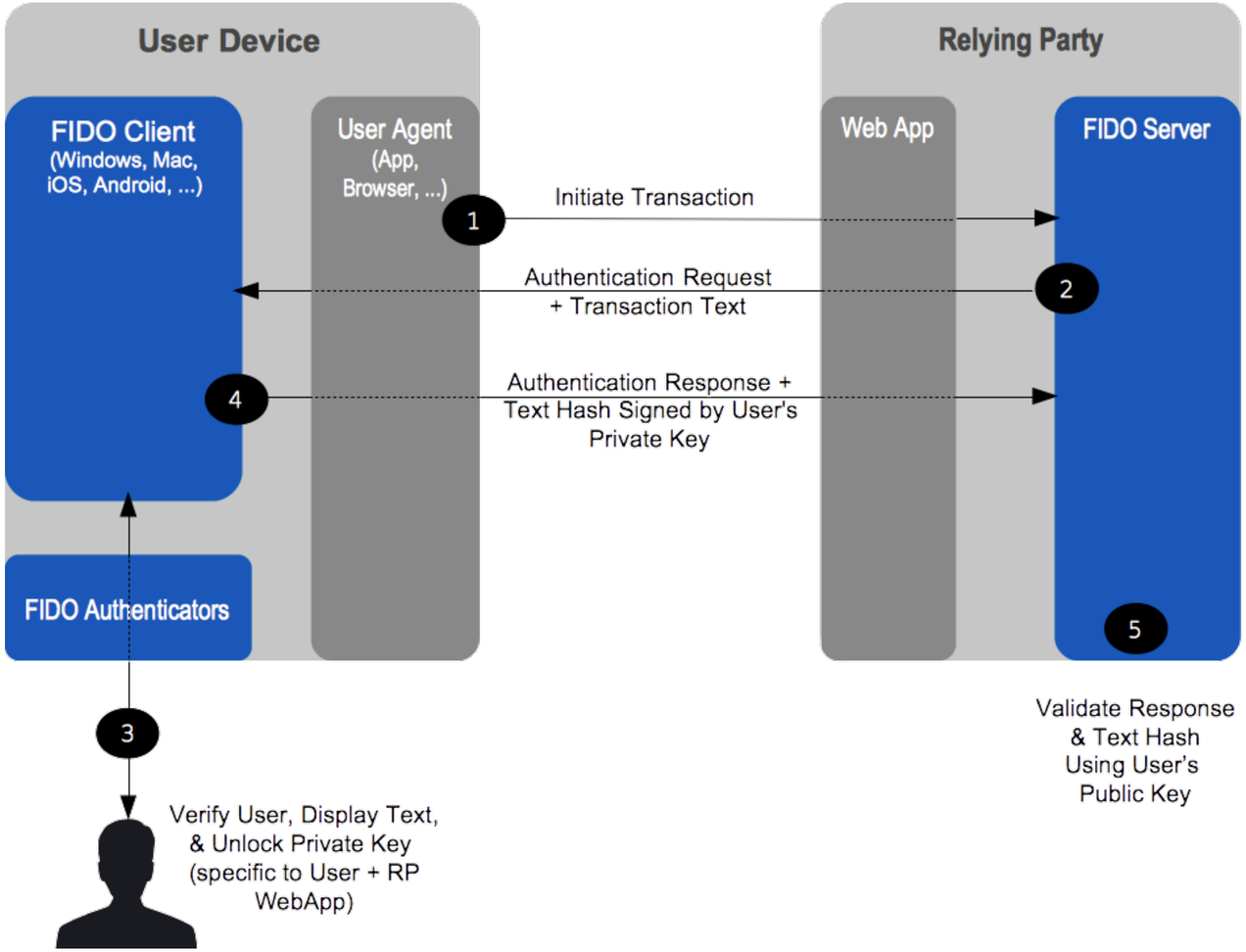
User Registration Process

FIDO UAF



User Authentication Process

FIDO UAF



User Transaction Confirmation Process

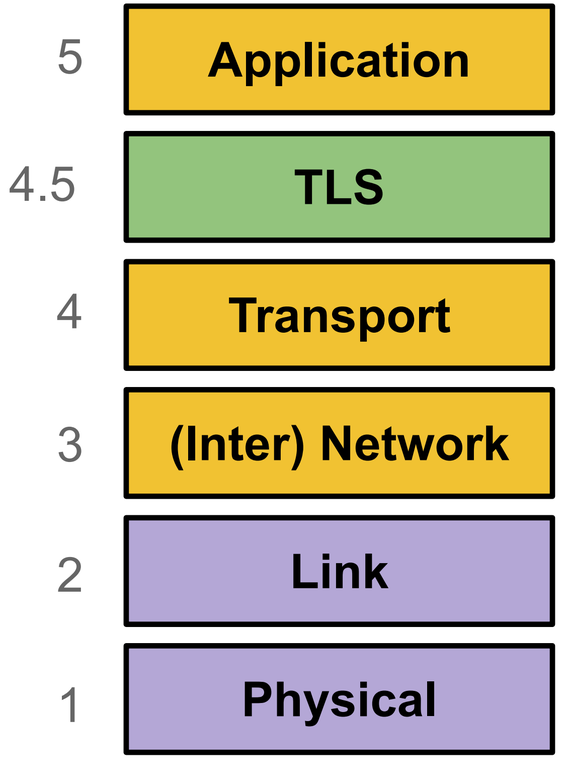
TLS (Transport Layer Security)

Xianghang Mi

An Overview of TLS

TLS Overview: The Definition

TLS (**Transport Layer Security**): a protocol for creating a secure communication channel over the insecure Internet

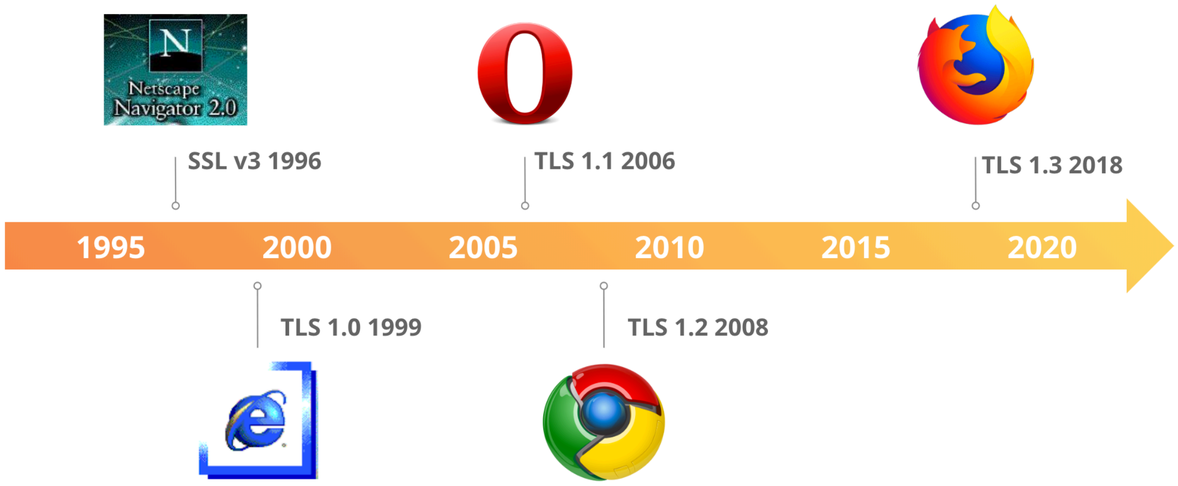


TLS Overview: Security Goals

- Goals of TLS
  - **Confidentiality**: Ensure that attackers cannot read your traffic
  - **Integrity**: Ensure that attackers cannot tamper with your traffic
    - Prevent **replay attacks**
      - The attacker records encrypted traffic and then replays it to the server
      - Example: Replaying a packet that sends “Pay $10 to Mallory”
  - **Authenticity**: Make sure you’re talking to the legitimate server
    - Defend against an attacker impersonating the server
  - **Forward Secrecy**: If an attacker records a connection now and compromises secret values later, they cannot compromise the recorded connection

TLS Overview: The History

- TLS 1.3
  - Defined in Aug 2018
  - Improvements in performance and security compared to TLS 1.2

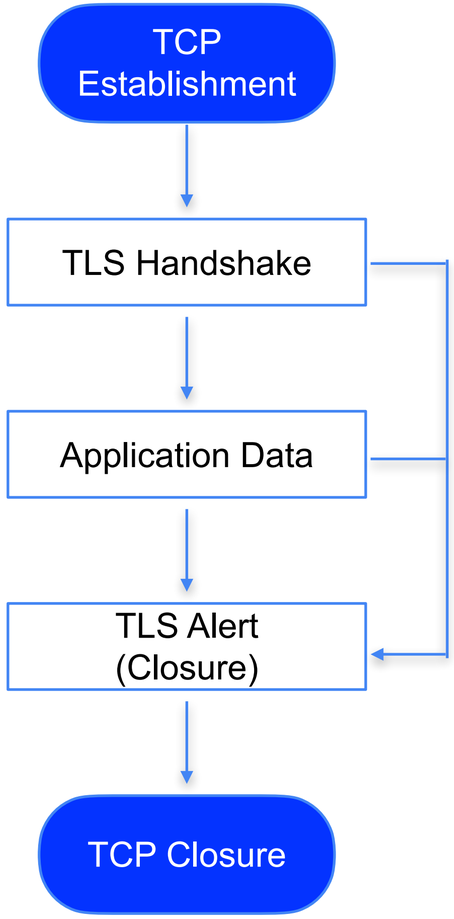


https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/

Protocol	Published	Status
TLS 1.3	2018	In use
TLS 1.2	2008	In use
TLS 1.1	2006	Deprecated in 2021
TLS 1.0	1999	Deprecated in 2021
SSL 3.0	1996	Deprecated in 2015
SSL 2.0	1995	Deprecated in 2011
SSL 1.0	Unpublished	Unpublished

TLS Overview: The Workflow

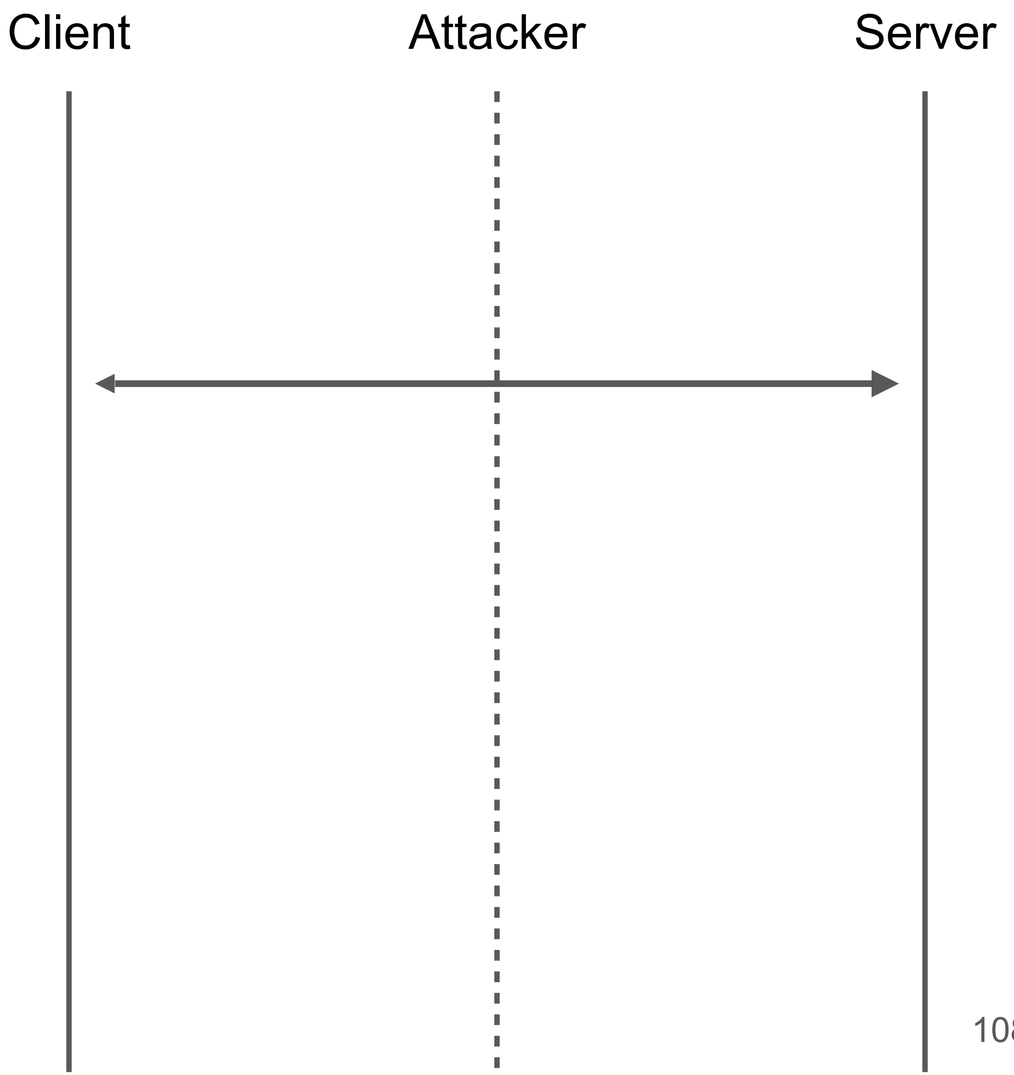
- A handshake protocol
  - Negotiate a cipher suite
  - Agree a master secret
  - Verify identities
- A record protocol
  - Transmit application data, handshake messages, and alert messages
- An alert protocol
  - Indicate errors or closure



Diving Into TLS

TLS: Threat Model

- The Client (Alice)
  - It can be a browser, a mobile app, etc
- The Server (Bob)
  - E.g., a web server
  - It typically has a CA-issued certificate
- The MITM attacker (Mallory)
  - It can be any intermediate hops, e.g., a router, or a WIFI hotspot
  - It may record, intercept, tamper, or replay the communication between Alice and Bob

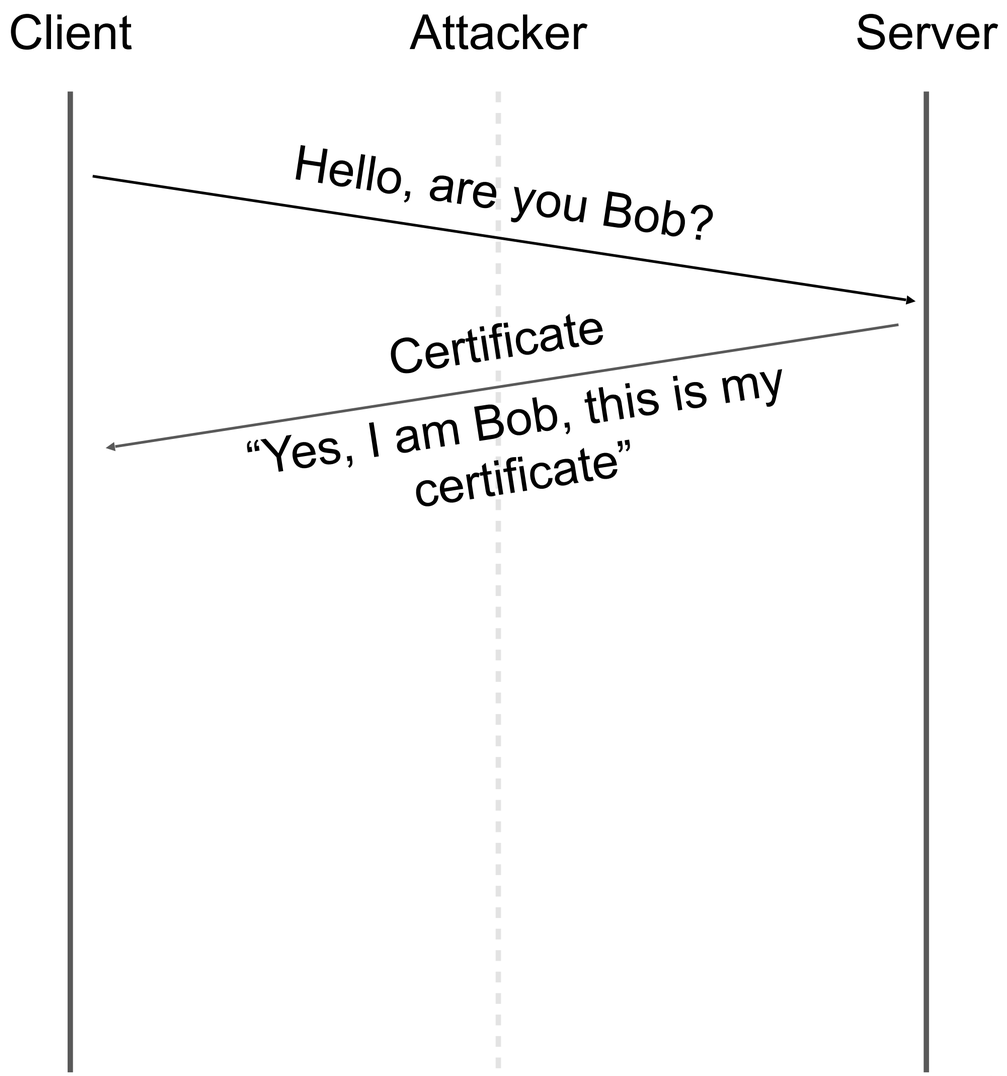


TLS Overview: Security Goals

- Goals of TLS
  - **Confidentiality**: Ensure that attackers cannot read your traffic
  - **Integrity**: Ensure that attackers cannot tamper with your traffic
    - Prevent **replay attacks**
      - The attacker records encrypted traffic and then replays it to the server
      - Example: Replaying a packet that sends “Pay $10 to Mallory”
  - **Authenticity**: Make sure you’re talking to the legitimate server
    - Defend against an attacker impersonating the server
  - **Forward Secrecy**: If an attacker records a connection now and compromises secret values later, they cannot compromise the recorded connection

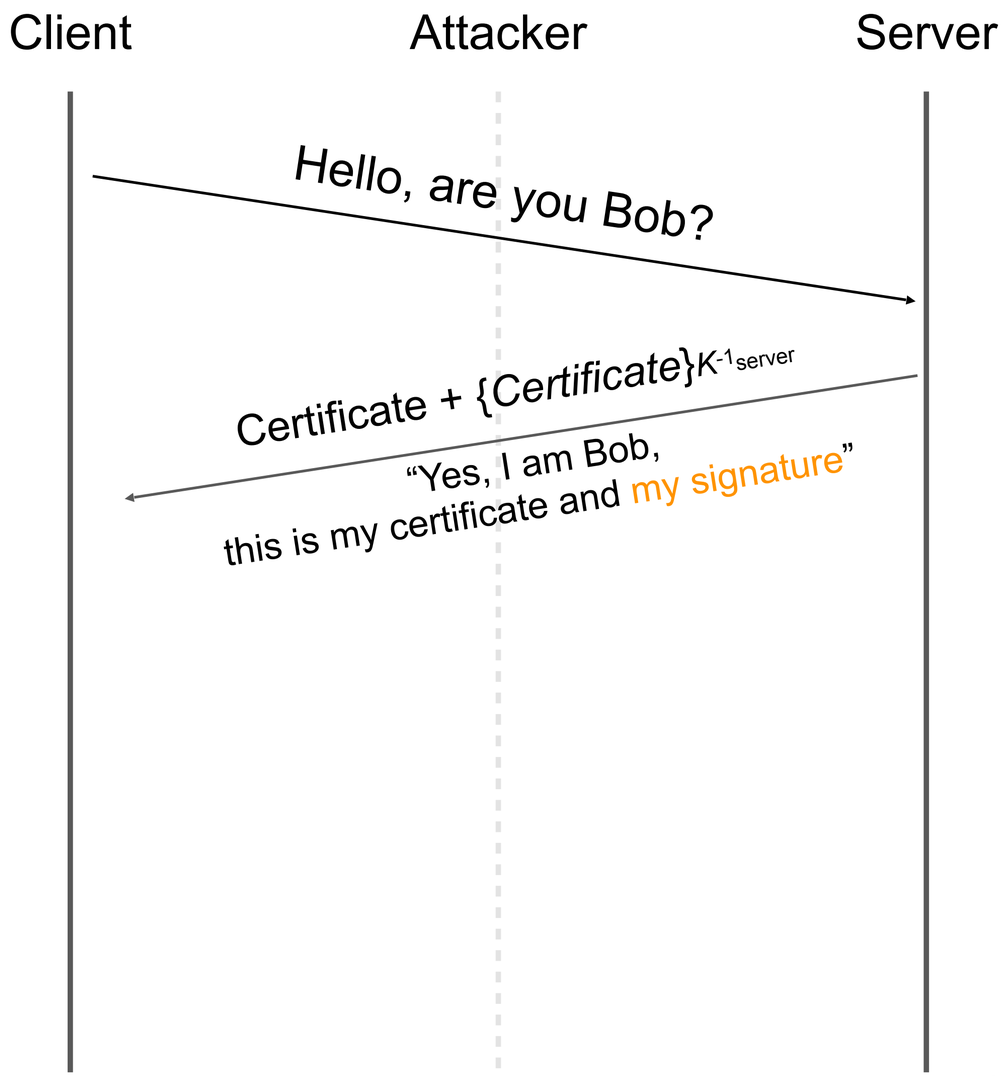
TLS Task I: Authenticating the Server

- The server sends its certificate
  - Recall certificates: The server's identity and public key, signed by a trusted certificate authority
- The client validates the certificate
  - Verify the signature in the certificate as generated using the private key of the CA
- The client now knows the server's public key
  - The client is not yet sure that they are talking to the legitimate server (not an impersonator)
  - Recall: Certificates are public. Anyone can provide a certificate for anybody
  - How to address this issue?



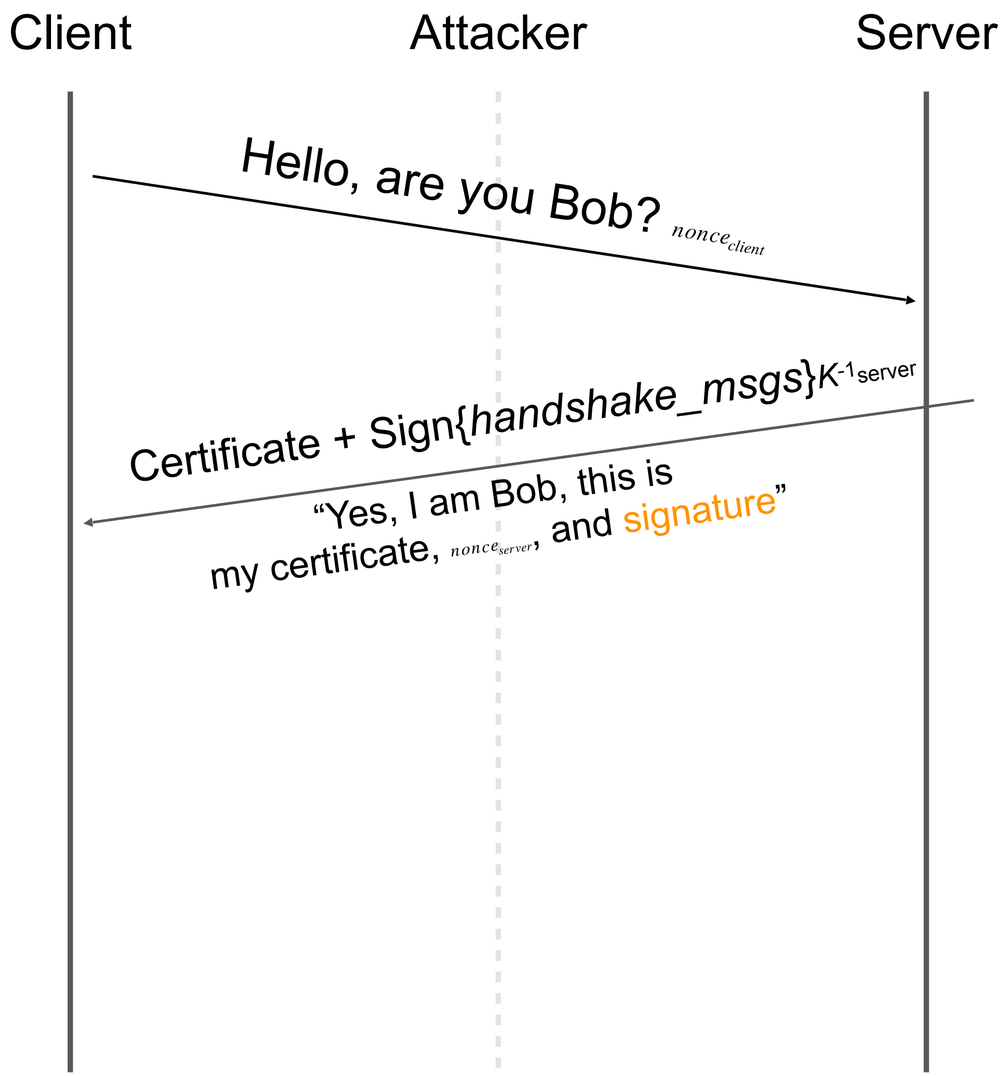
TLS Task I: Authenticating the Server

- What if the server signs its certificate and returns the signature along with the certificate?
  - In addition to the certificate, the server responds with also the signature over the certificate
  - The attacker can record and replay this signature



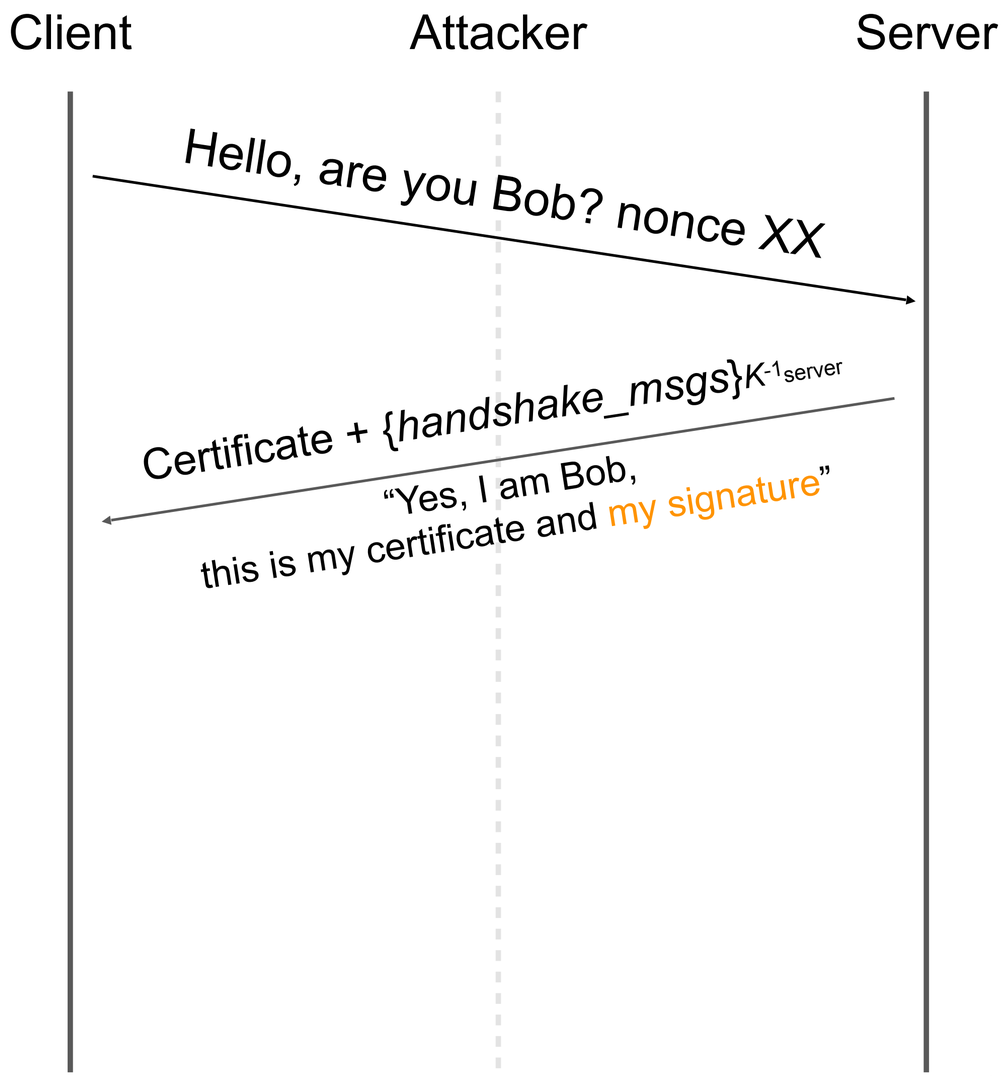
TLS Task I: Authenticating the Server

- What if a challenge-response protocol?
  - The client embeds in each hello message with a random nonce
  - The server generates a signature for all the handshake messages up to this point
    - The random nonce is part of the msgs
  - Is the replay attack still doable?
    - No!



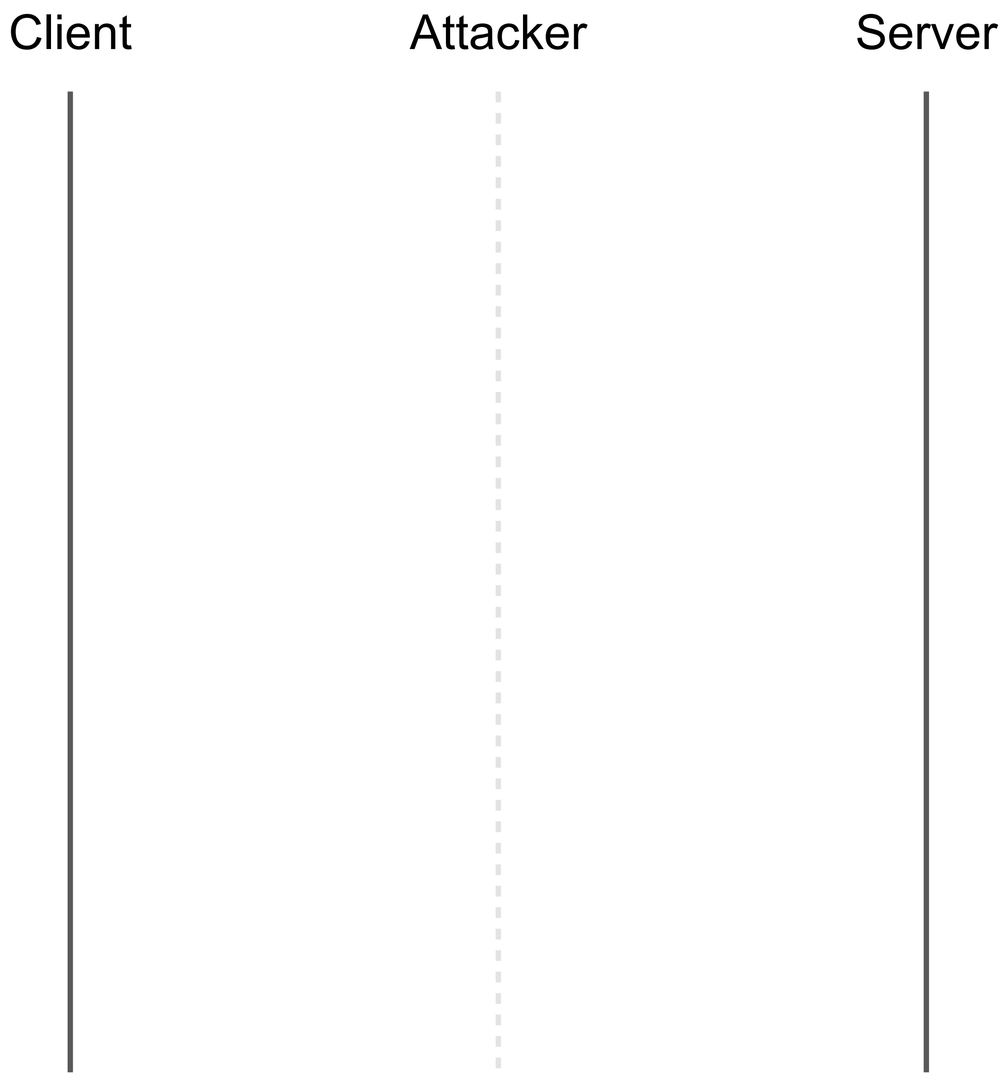
TLS Task I: Authenticating the Server

- How to authenticate the client?
  - The same way should the client has a certificate
- What if neither side has a certificate?
  - TLS also supports authentication through pre-shared key (PSK)
    - The PSK can be either out-of-band shared or negotiated in previous TLS connections
    - It is a corner case with security risks introduced



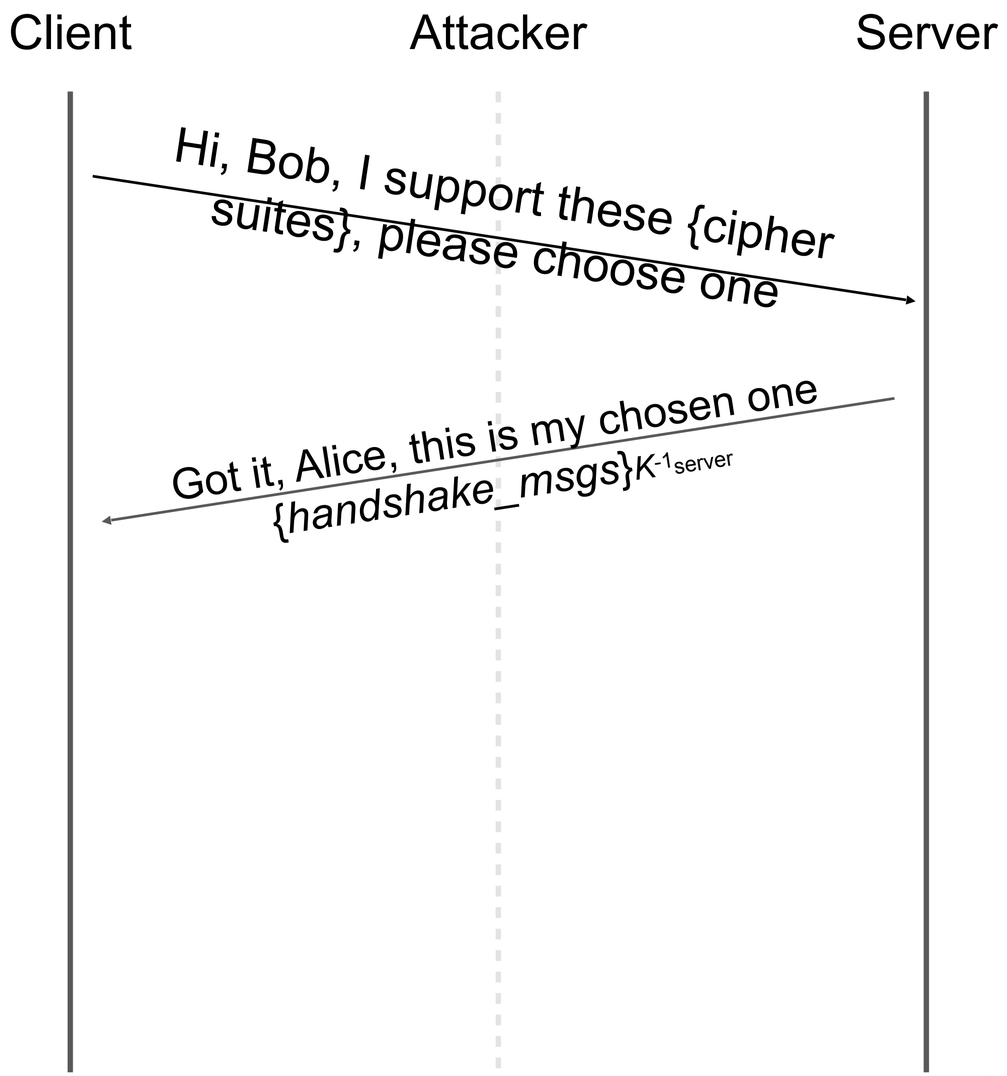
TLS Task 2: Negotiating the Cipher Suite

- Given the server authenticated, how to fulfill confidentiality and integrity for future data transmission?
- Authenticated encryption (AE) schemes
  - MAC-then-encrypt
    - It is used in TLS 1.2, however, recall its security risks and limitations
  - Encrypt-then-MAC
  - Authenticated encryption with additional data (**AEAD**)
    - TLS 1.3 supports only AEAD
- Hash algorithms are also needed to be decided for future key derivation
- We need to negotiate **the cipher suite (AE algorithm + hash algorithm)**



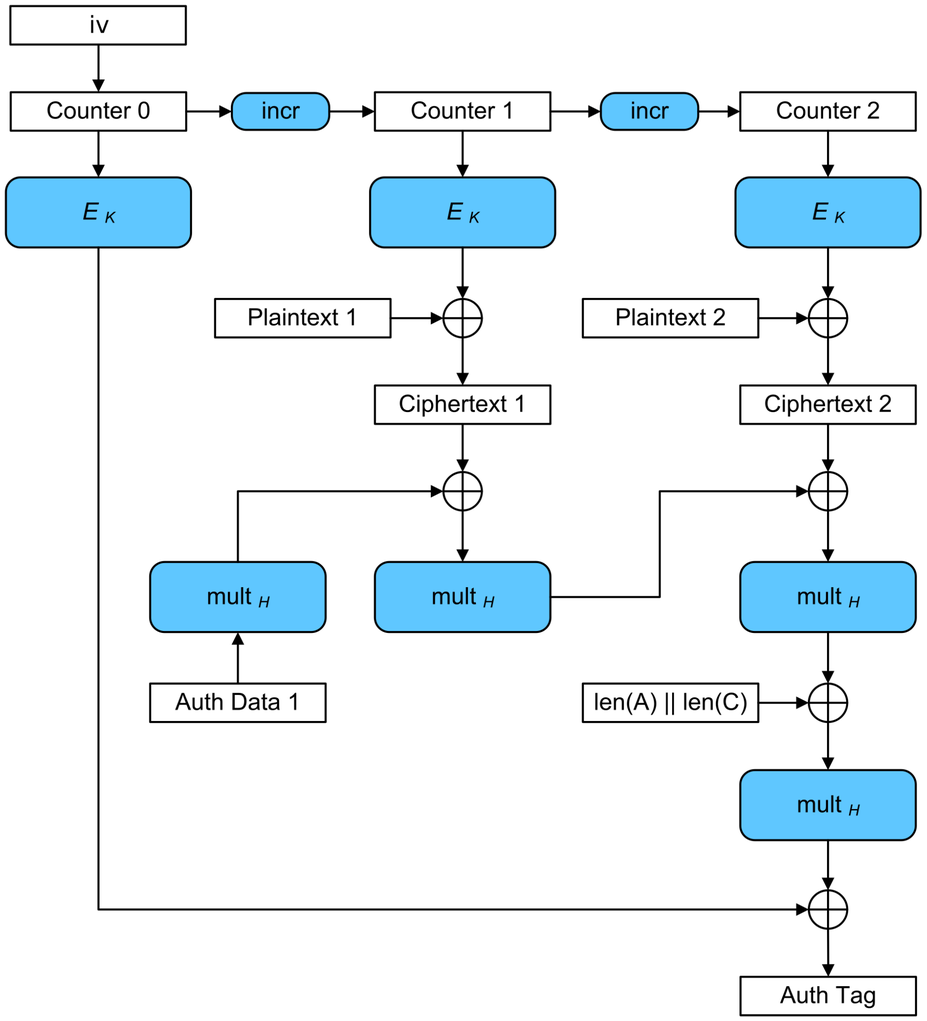
TLS Task 2: Negotiating the Cipher Suite

- Giving too many options leads to the wrong option being chosen
  - TLS 1.2 adopts MAC-then-encrypt along with 37 different cipher suites
  - 17 are vulnerable to attacks, e.g., Padding oracle attacks
- Instead, TLS 1.3 supports ONLY 5 cipher suites known to be secure
  - 5 AEAD algorithms, e.g., AEAD_CHACHA20_POLY1305, AEAD_AES_256_GCM
  - Two hash algorithms: SHA256 and SHA384
- What if the attacker manipulates the plaintext handshake messages, e.g., change the selected cipher suite from one to another
  - The handshake messages by this point is signed, and its integrity can be verified



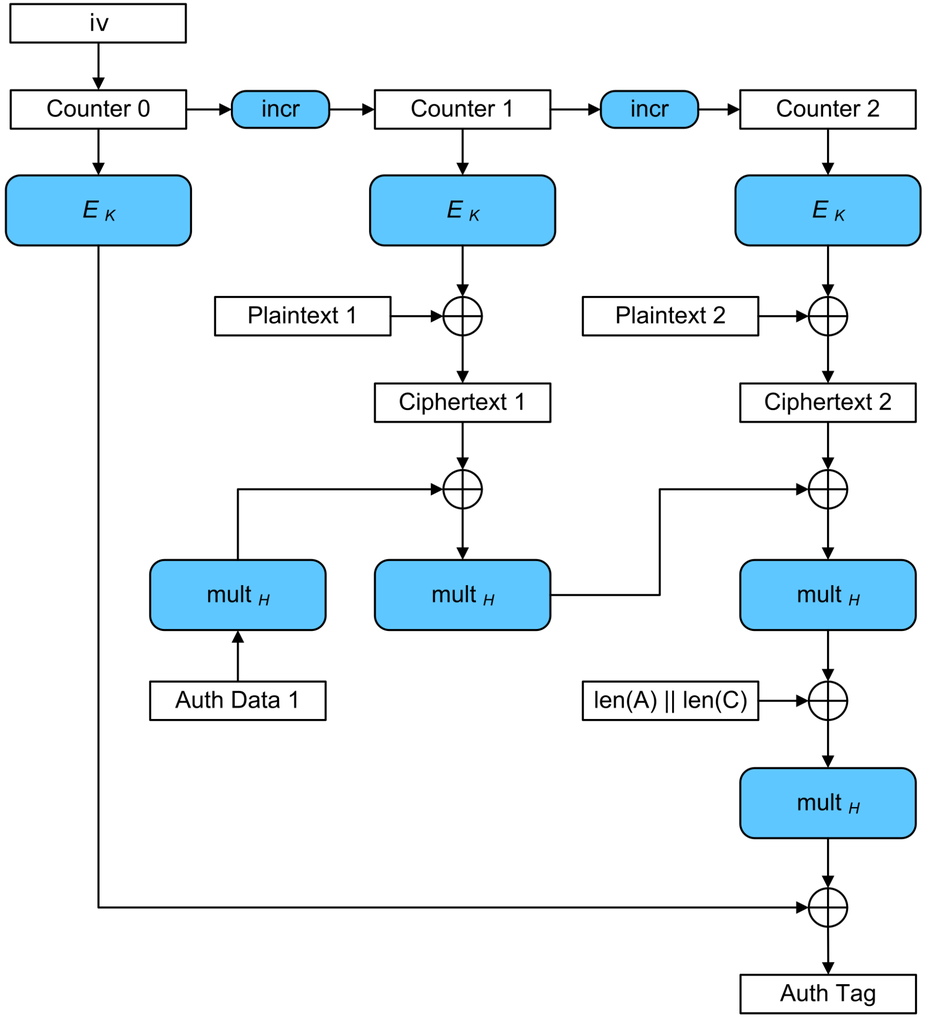
AEAD Example: Galois Counter Mode (GCM)

- **Galois Counter Mode** (**GCM**): An AEAD block cipher mode of operation
- **Input**: the key, the plaintext, and the additional data
- **Output**: the cipher text, the auth tag, and the additional data
- E_K is standard block cipher encryption
- mult_H is 128-bit multiplication over a special field (Galois multiplication)
  - Don't worry about the math



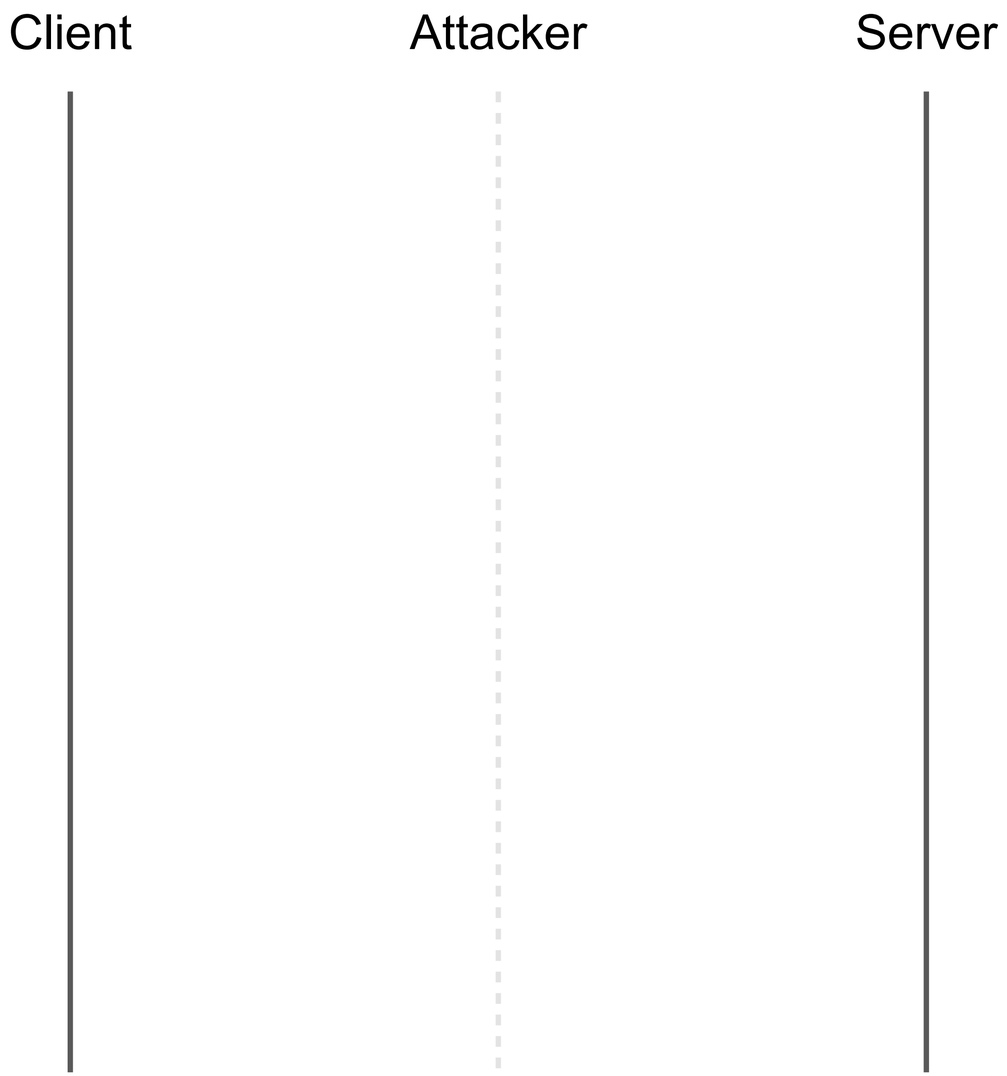
AEAD Example: Galois Counter Mode (GCM)

- Very fast mode of operation
  - Fully parallel encryption
  - Galois multiplication isn't parallelizable, but it's very fast
- Drawbacks
  - IV reuse leads to loss of confidentiality, integrity, and authentication
  - This wouldn't happen if you used AES-CTR and HMAC-SHA256
  - Implementing Galois implementation is difficult and easy to screw up
- **Takeaway**: GCM provides integrity and confidentiality, but if you misuse it, it's even worse than CTR mode



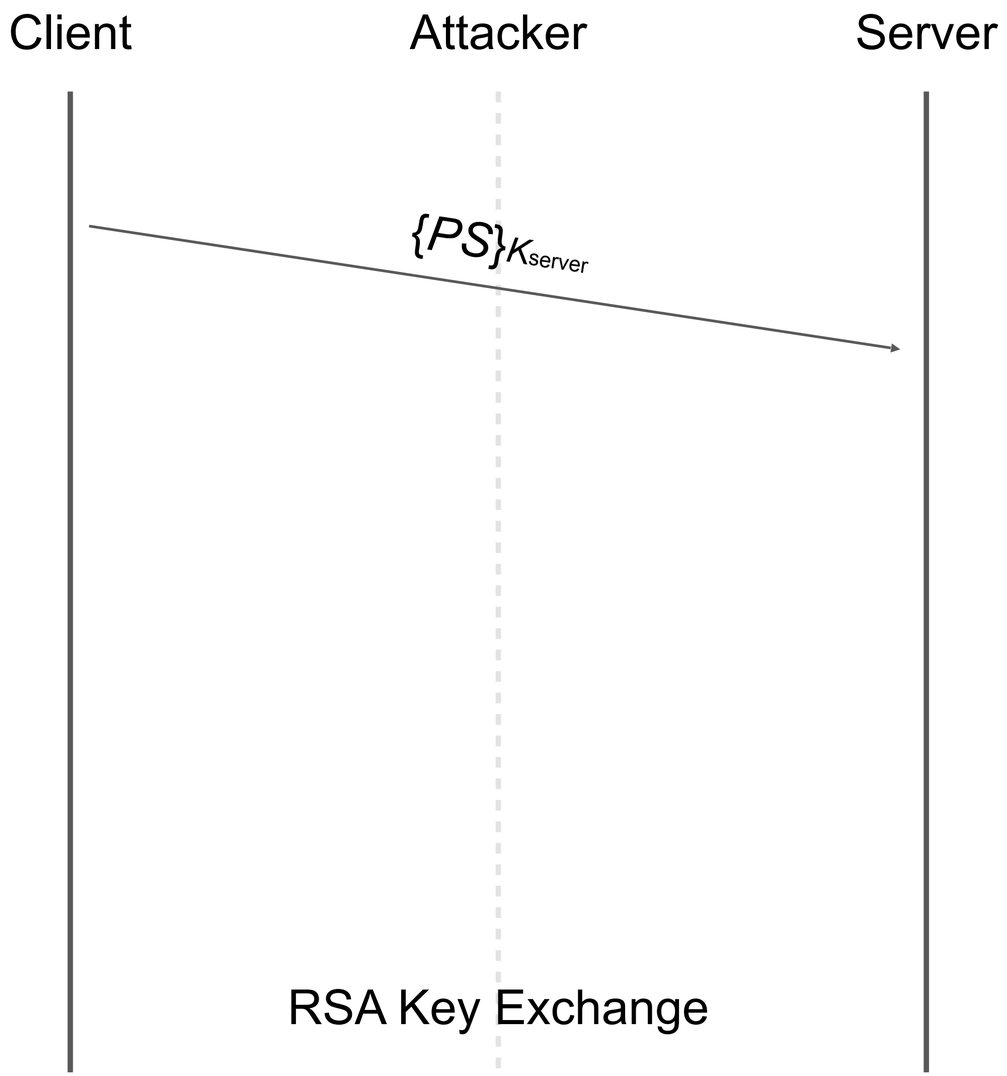
TLS Task 3: Exchanging Key Materials

- Given the cipher suite negotiated, we need to have key materials exchanged safely
- How to achieve it?
  - RSA key exchange
  - (elliptic curve) Diffie-Hellman key exchange (DHE)



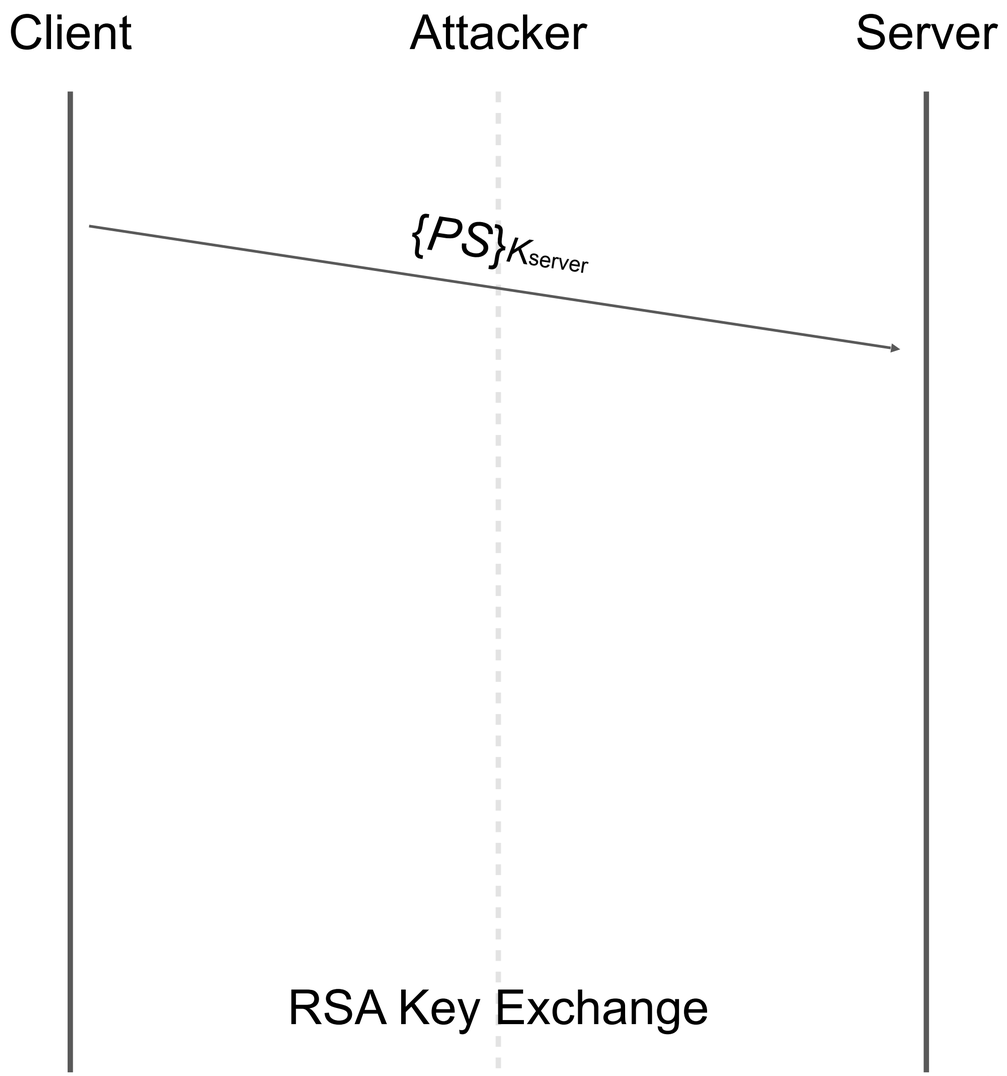
TLS Task 3: Exchanging Key Materials (RSA)

- The client randomly generates a **premaster secret** (**PS**)
- The client encrypts *PS* with the server's public key and sends it to the server
  - The client knows the server's public key from the certificate
- The server decrypts the premaster secret
- The client and server now share a secret
  - Recall RSA encryption: Nobody except the legitimate server can decrypt the premaster secret
  - Proves that the server owns the private key (otherwise, it could not decrypt *PS*)
- Any issues?



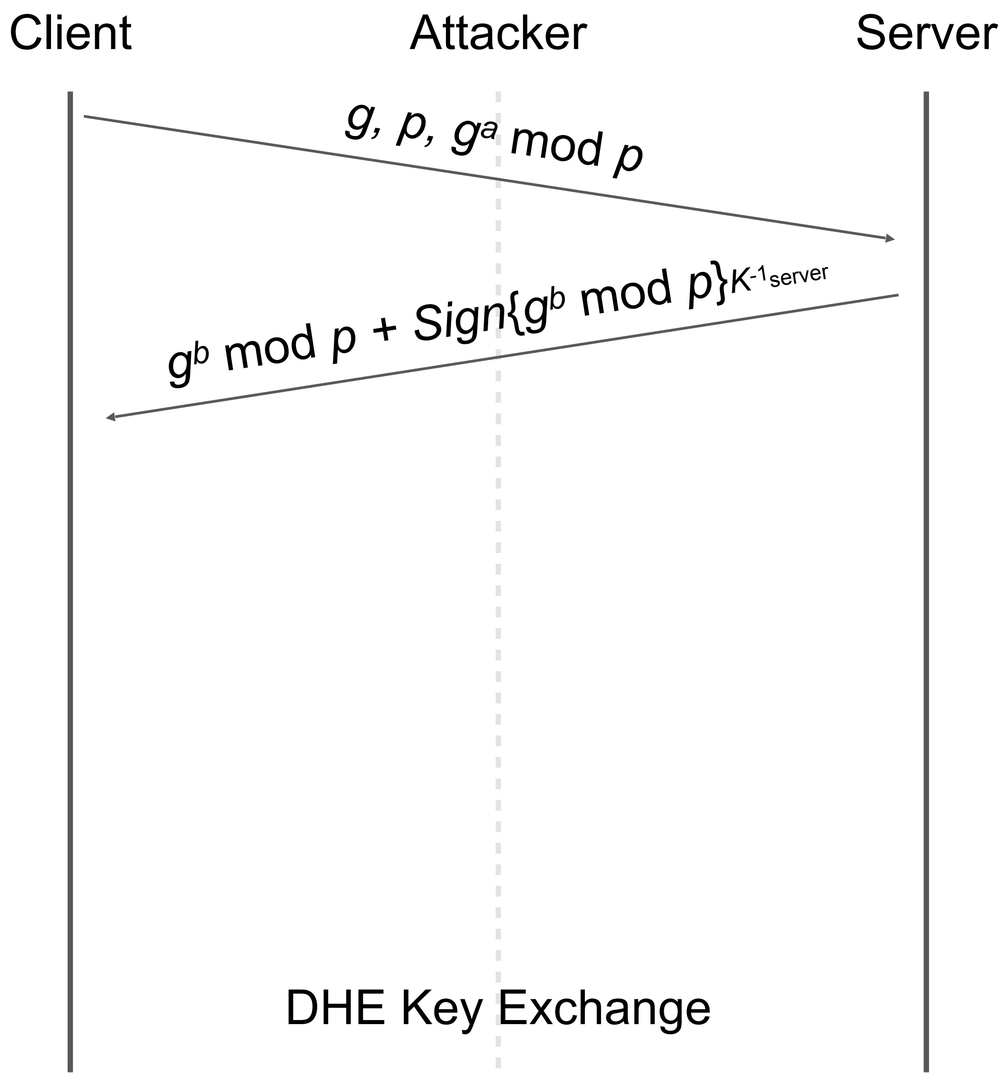
TLS Task 3: Exchanging Key Materials (RSA)

- Does it guarantee forward secrecy?
  - If the attacker compromises the server's private key at a future time point?
  - $\{PS\}_{K_{server}}$ recorded at this point will be available to the attacker
- RSA key exchange doesn't guarantee forward secrecy
- Thus, it is deprecated in TLS 1.3



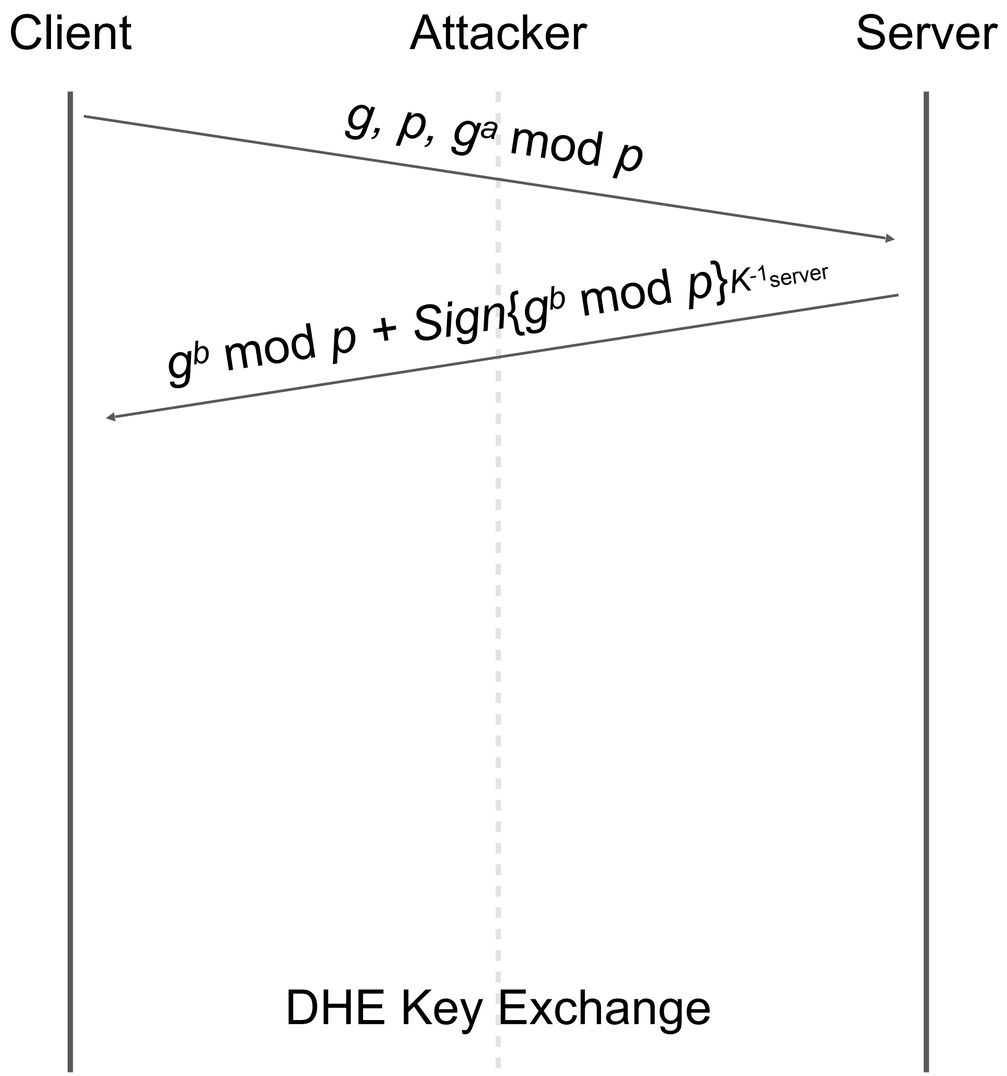
TLS Task 3: Exchanging Key Materials (DHE)

- The client generates a secret *a* and computes g^a mod *p*
- *The Client send generator g, prime p,* g^a mod *p (the DHE key share), to the server*
- The server generates a secret *b* and computes g^b mod *p*
- The server signs g^b mod *p* with its private key and sends its DHE key share and signature



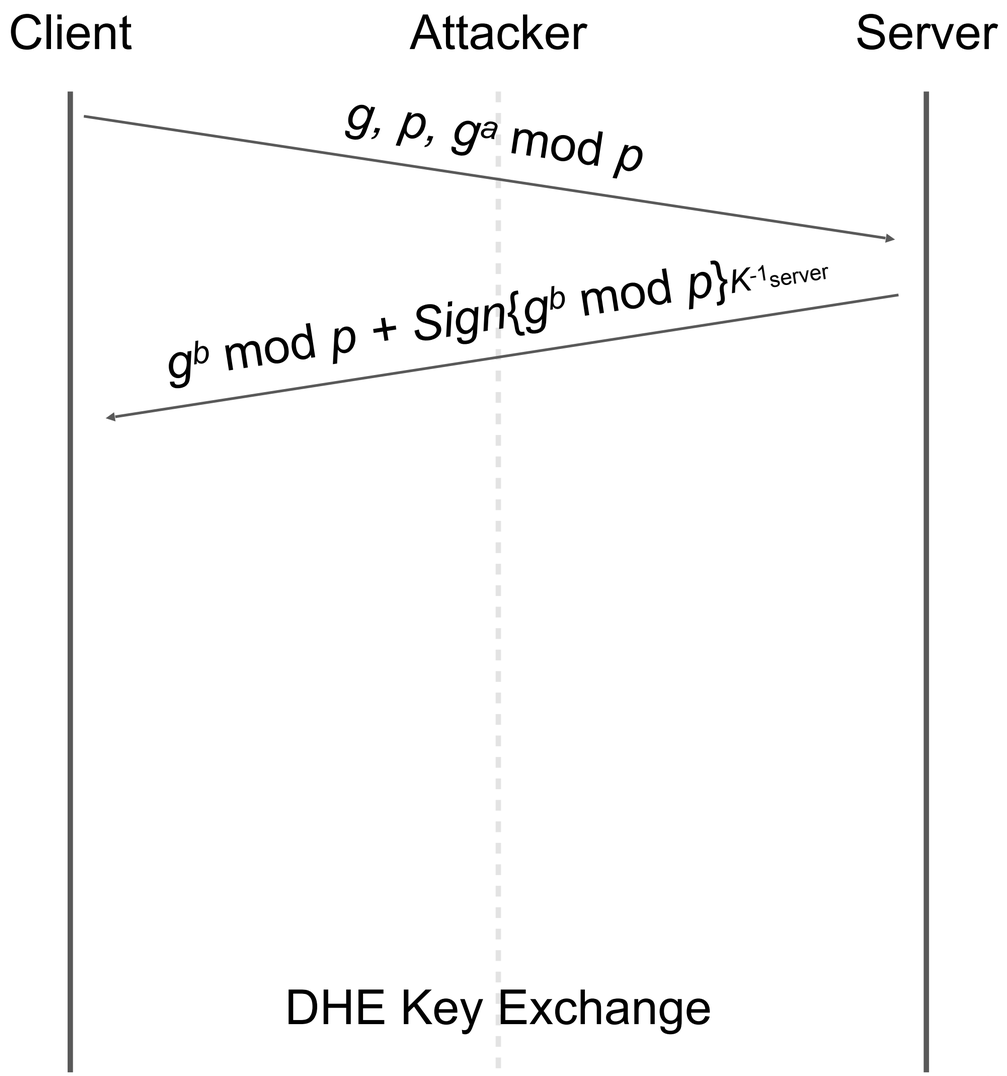
TLS Task 3: Exchanging Key Materials (DHE)

- The client and server now share a **premaster secret**: g^{ab} mod p
  - Recall Diffie-Hellman: an attacker cannot compute g^{ab} mod p, *thanks to the hardness of discrete logarithm problem*
- Does DHE guarantee forward secrecy?
  - Yes as long as the resulting PS is removed once the TLS session is closed
- Any issues?



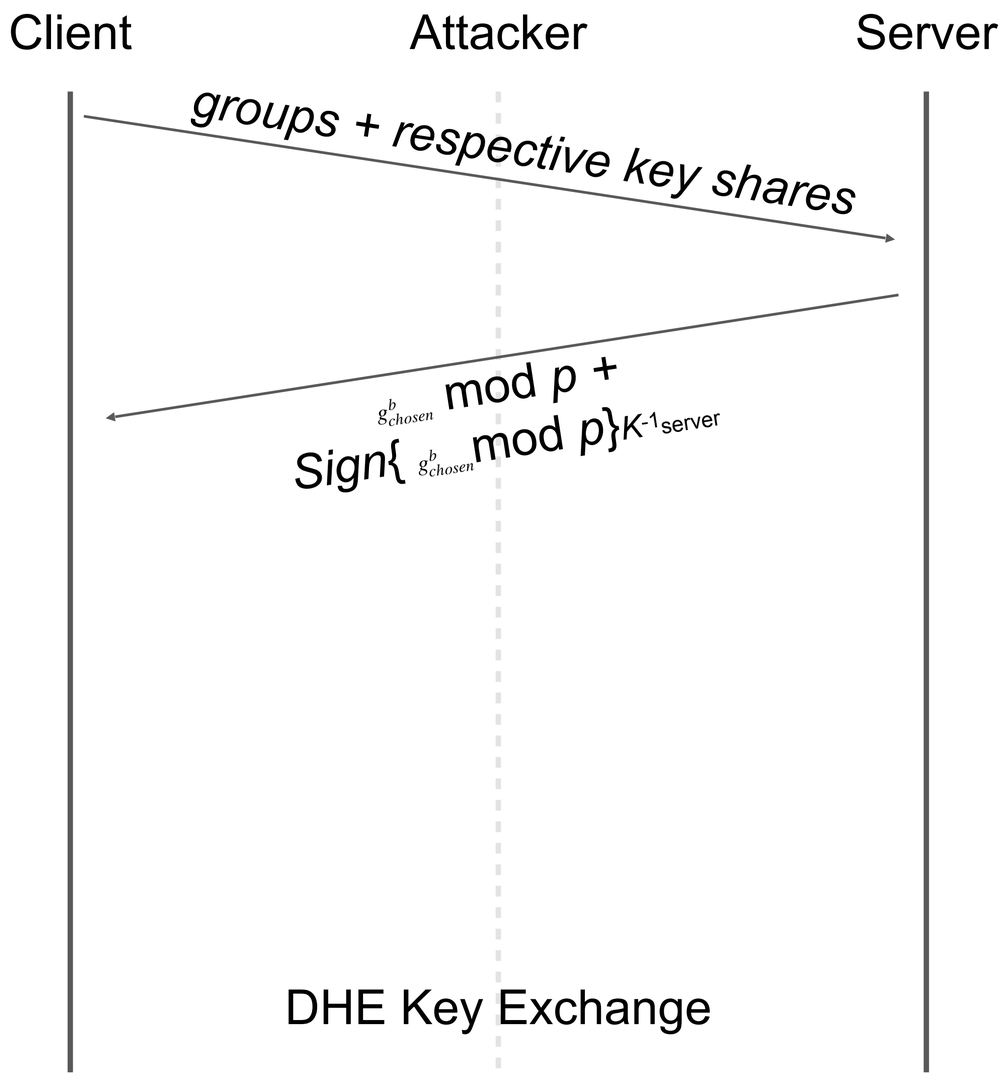
TLS Task 3: Exchanging Key Materials (DHE)

- There are a huge number of group generators g and prime p to select from
  - finite field groups
  - elliptic curve groups
- Many TLS 1.2 servers could be tricked into using small numbers for Diffie-Hellman, and thus breaking the security of DHE
  - The LogJam and WeakDH attacks
- Instead, TLS 1.3 supports ONLY a small set of 10 groups that are considered as secure
  - E.g., x25519 and ffdhe8192



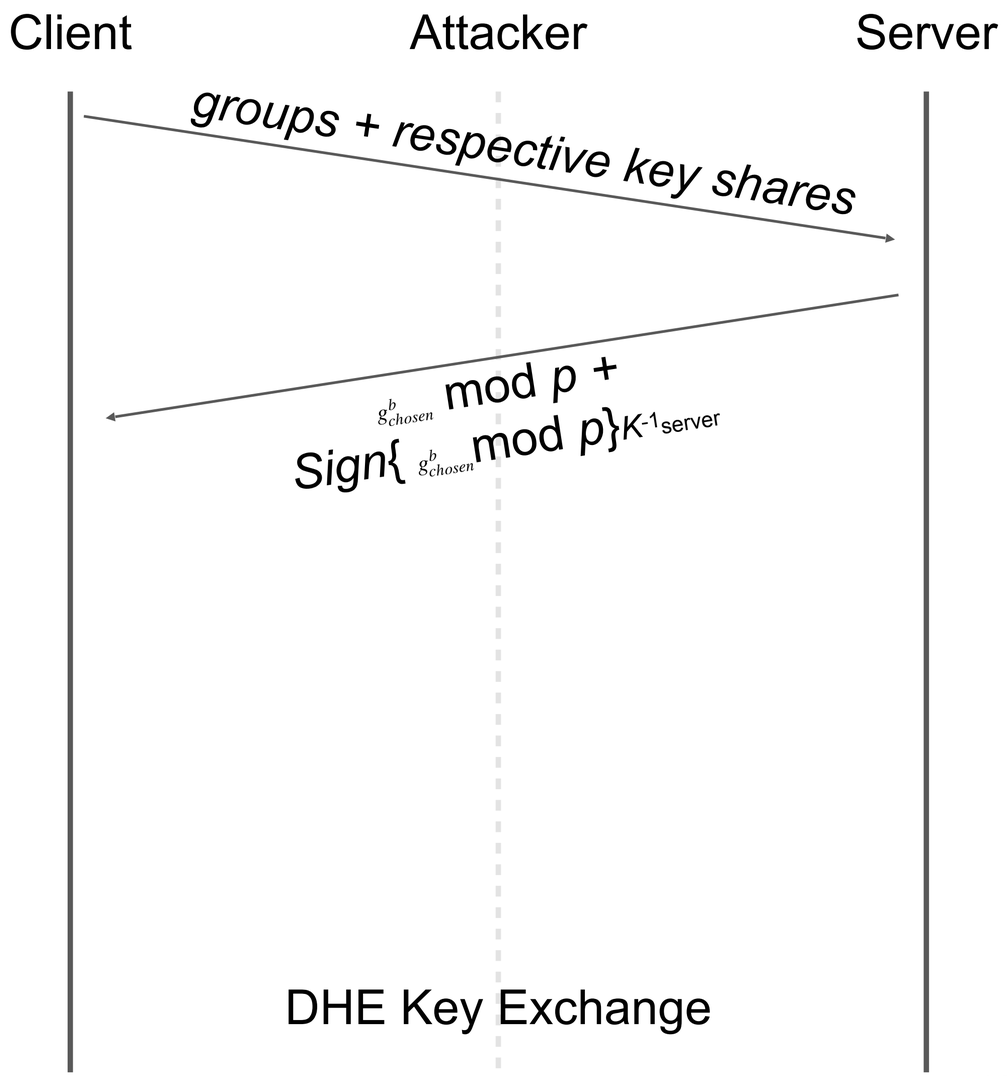
TLS Task 3: Exchanging Key Materials (DHE)

- How can the client know the server support a given group?
  - Not known in advance
  - However, we only have 10 groups
- Therefore, the client first sends all the groups it prefers along with the respective key share entries
- The server chooses one from the provided groups, and responds with the chosen group along with its key share



TLS Task 4: Deriving Secrets and Keys

- Separate keys and nonces (IVs) for different traffic scenarios:
  - *AEAD for* **client-to-server handshake** message
    - *Key{ client, handshake}*
    - *IV{ client, handshake}*
    - *Key{ client, finished}*
  - *AEAD for* **client-to-server application** message
    - *Key{ client, application}*
    - *IV{ client, application}*
  - *AEAD for* **server-to-client handshake** message
    - *Key{ server, handshake}*
    - *IV{server, handshake}*
    - *Key{ server, finished}*
  - *AEAD for* **server-to-client** application message
    - *Key{ server, application}*
    - *IV{ server, application}*



TLS Task 4: Deriving Secrets and Keys

- How?
  - Bootstrapping from PS
  - HKDF
    - Generate the master secret and the handshake secrete through HKDF-Extract
    - Generate intermediate secrets through HKDF-Expand,
      - E.g., client-side handshake traffic secret

```
(EC)DHE -> HKDF-Extract = Handshake Secret
           |
           +-----> Derive-Secret(., "c hs traffic",
           |                     ClientHello...ServerHello)
           |                     = client_handshake_traffic_secret
           |
           +-----> Derive-Secret(., "s hs traffic",
           |                     ClientHello...ServerHello)
           |                     = server_handshake_traffic_secret
           v
     Derive-Secret(., "derived", "")
           |
           v
0 -> HKDF-Extract = Master Secret
           |
           +-----> Derive-Secret(., "c ap traffic",
           |                     ClientHello...server Finished)
           |                     = client_application_traffic_secret_0
           |
           +-----> Derive-Secret(., "s ap traffic",
           |                     ClientHello...server Finished)
           |                     = server_application_traffic_secret_0
           |
           +-----> Derive-Secret(., "exp master",
           |                     ClientHello...server Finished)
           |                     = exporter_master_secret
           |
           +-----> Derive-Secret(., "res master",
                                 ClientHello...client Finished)
                                 = resumption_master_secret
```

TLS Task 4: Deriving Secrets and Keys

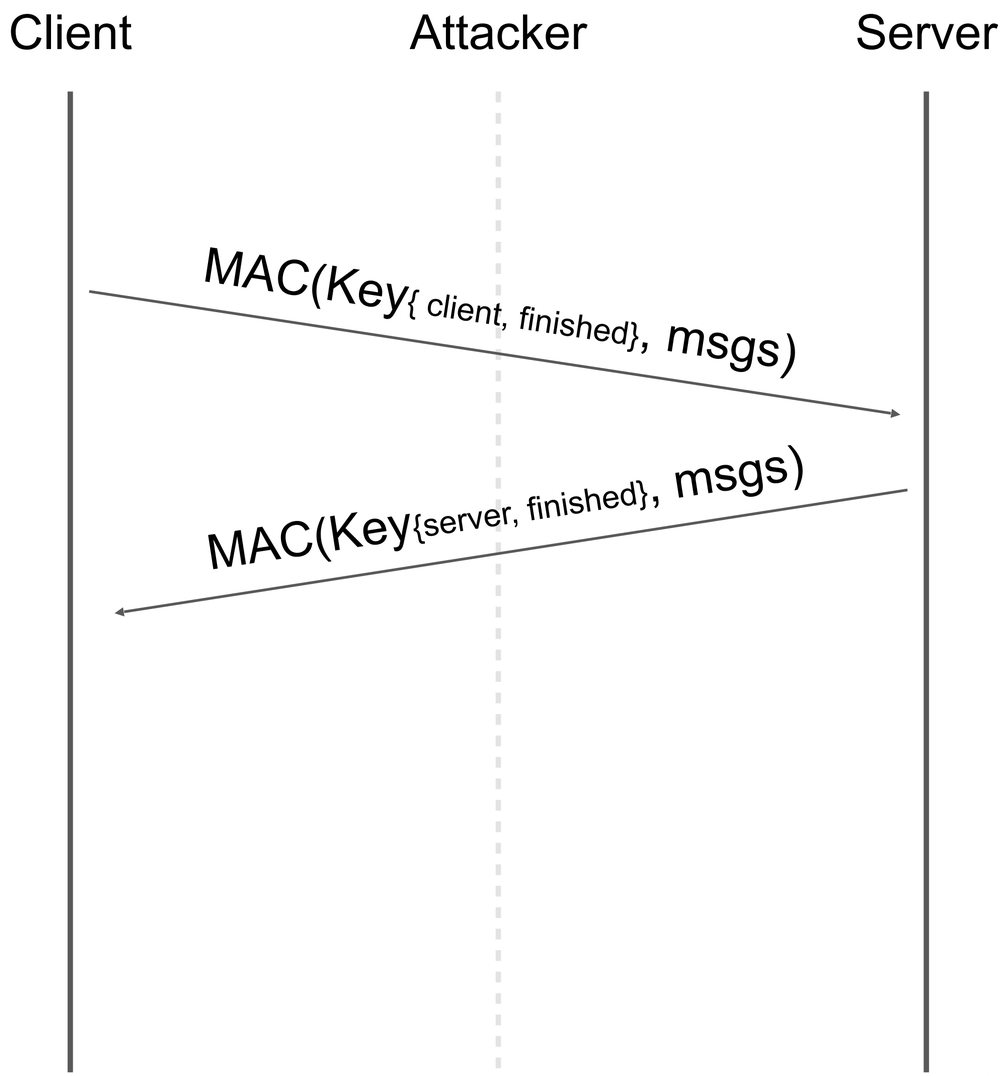
- How?
  - Generate traffic keys (and IVs) from respective traffic secrets

```
key = HKDF-Expand-Label(Secret, "key", "", key_length)
iv  = HKDF-Expand-Label(Secret, "iv", "", iv_length)
```

- Given key materials generated, how can both sides verify they are on the same page?
  - verifying the integrity of the handshake
  - detecting tampering from the MITM attacker
  - verifying their local key derivation process is identical, i.e., resulting in the same set of key materials

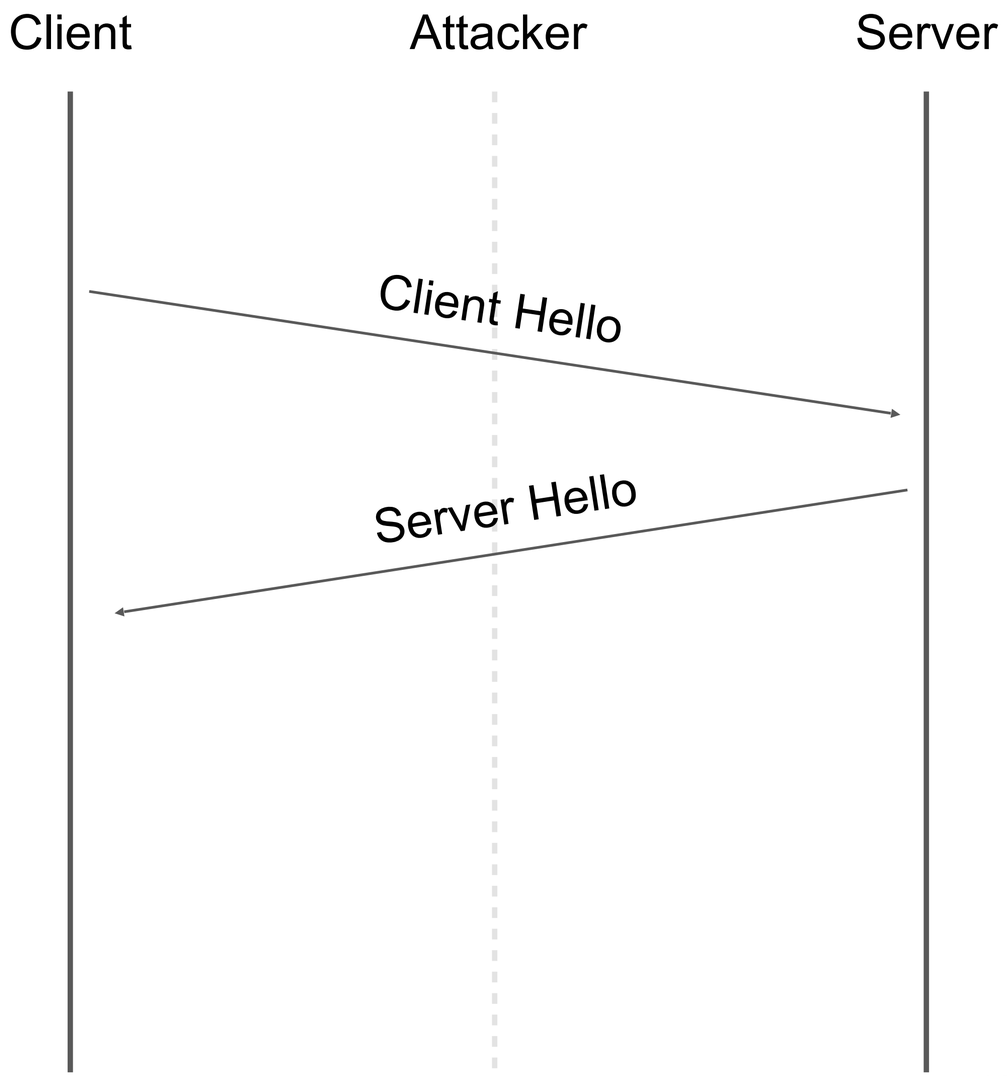
TLS Task 5: Verifying the Handshake

- The server and client exchange MACs on all the messages of the handshake so far
  - Recall MACs: Any tampering on the handshake will be detected
  - The MAC key is computed by following aforementioned key derivation process
  - If the resulting MACs are verified on both sides:
    - The handshake integrity is verified
    - The local key derivation process is verified



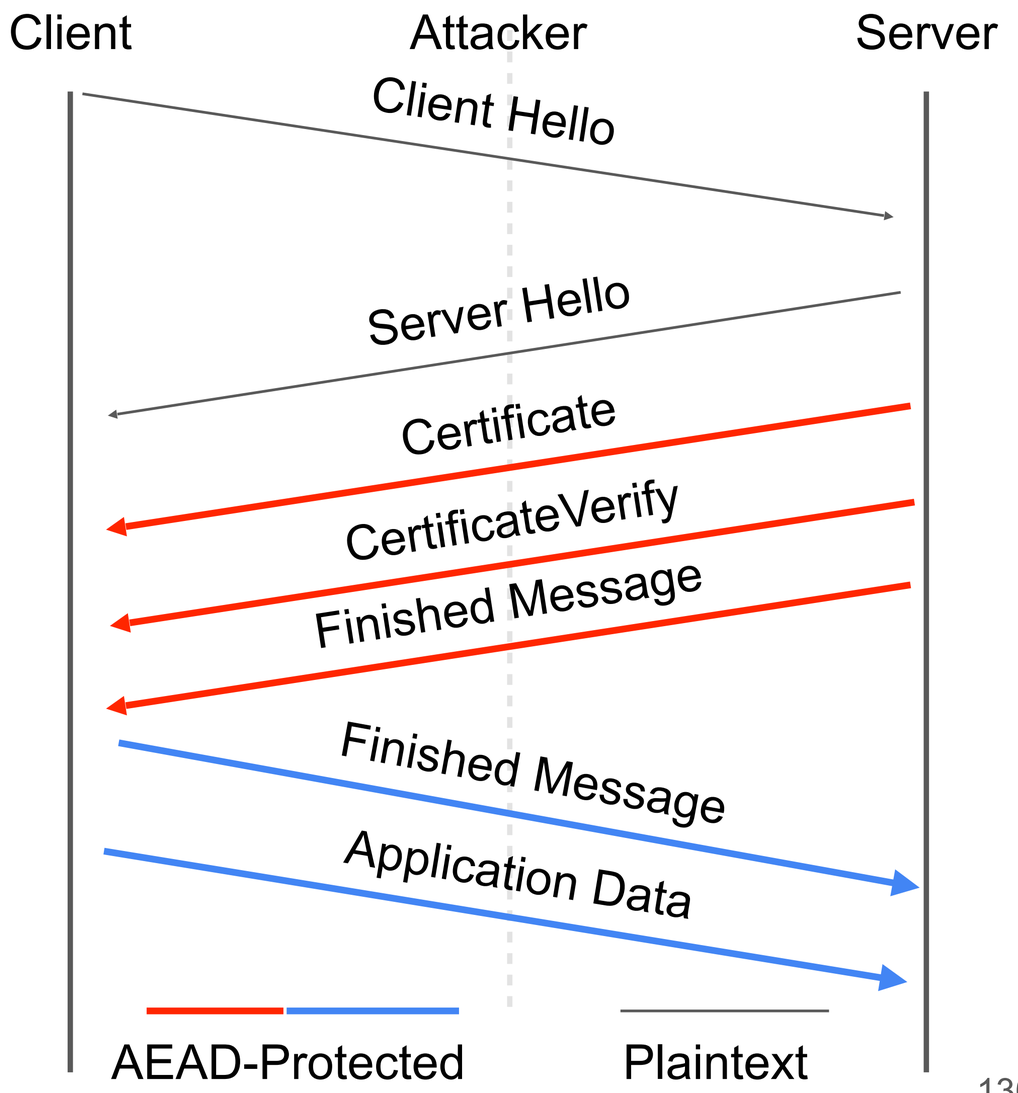
TLS Task 1-5: The Handshake Protocol

- Client Hello
  - Random nonce
  - Cipher suites
  - List of key shares for preferred DHE groups
  - Signature algorithm
- Server Hello
  - Random nonce
  - The chosen cipher suite (e.g. TLS_AES_128_GCM_SHA256)
  - The key share for the chosen DHE group
- where are the signatures?
  - E.g., the one sent when negotiating cipher suite and the one sent when negotiating key materials



TLS Task 1-5: The Handshake Protocol

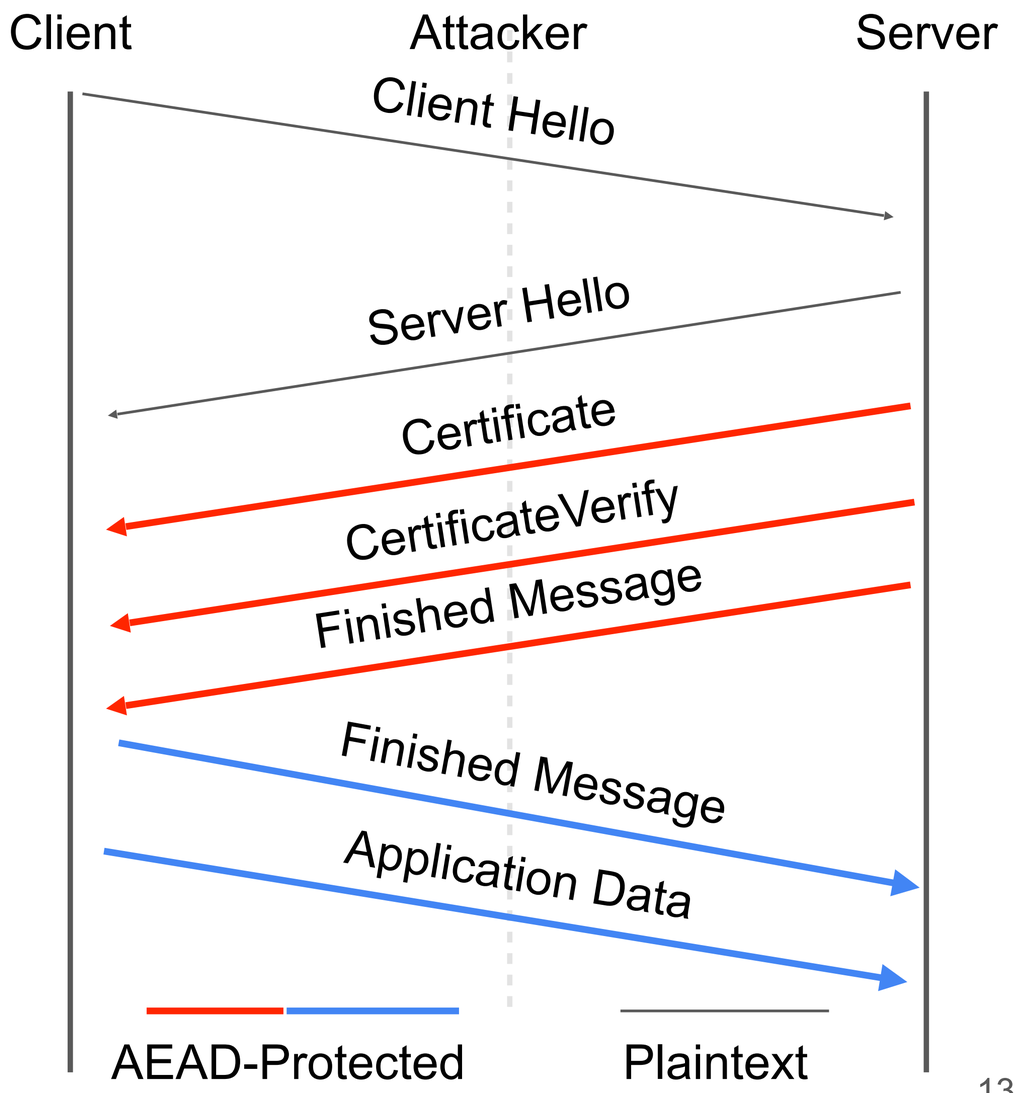
- Server **Certificate**
- Server **CertificateVerify:** a signature of all above messages including the certificate
  - It also verifies the integrity of both the client hello and server hello, and thus the cipher suites, and DHE key share
- Server **finished message**: MAC(all handshake messages by this point)
- By this point, it takes only one round trip of time
- Client **finished message**
- Application data



TLS Task 1-5: The Handshake Protocol

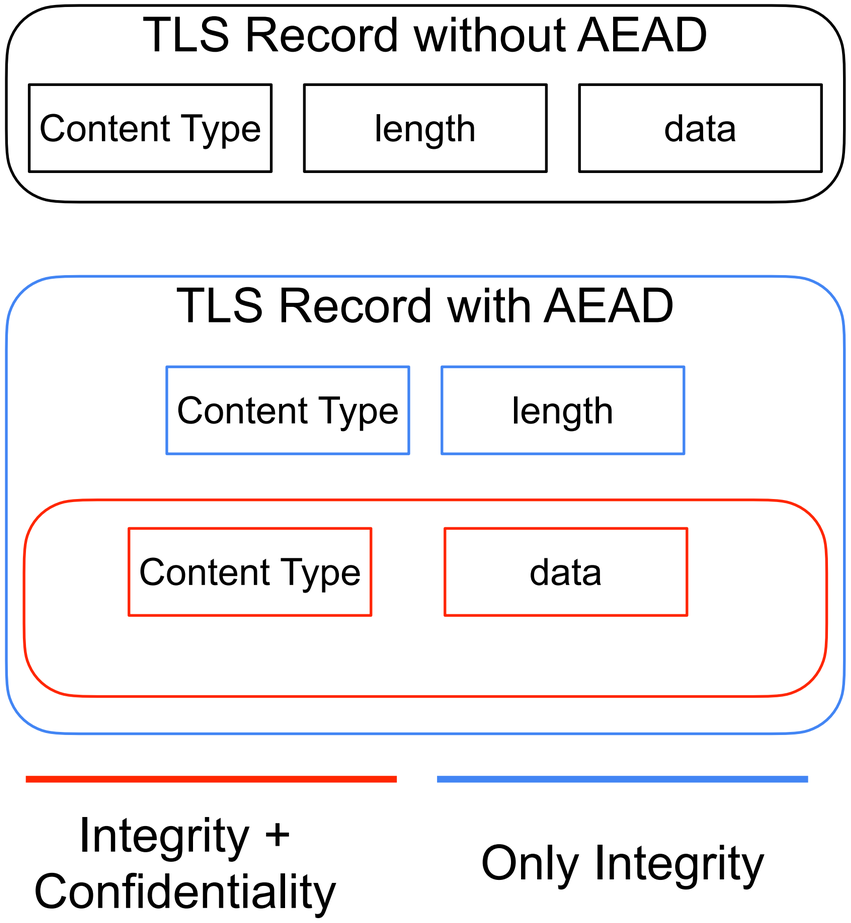
- How to distinguish different handshake messages?
  - Each handshake message has a field to denote its handshake type, e.g., client_hello, server_hello, finished

- How to distinguish between handshake messages and application data?
  - We have TLS record protocol



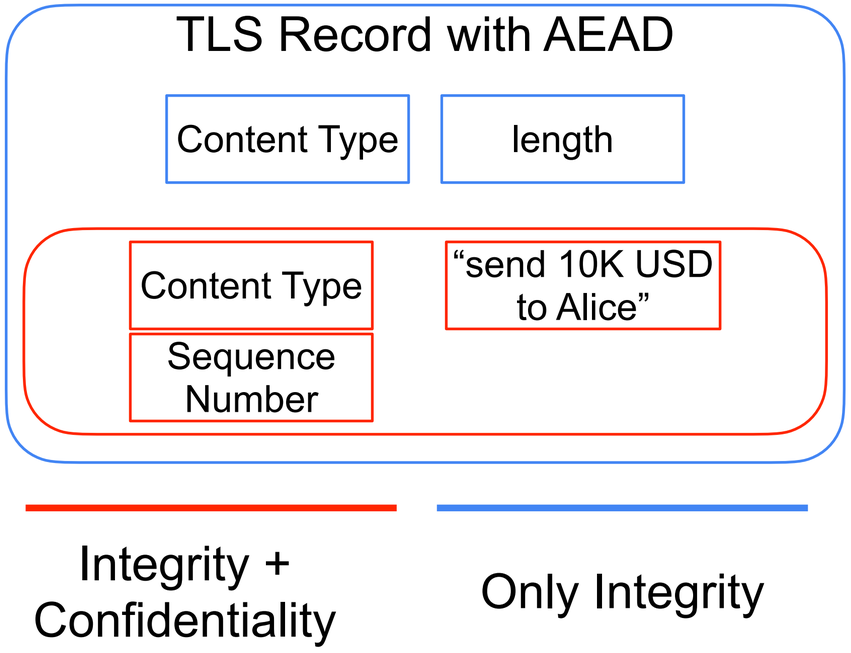
TLS Record Protocol

- Each TLS record has a content type field to distinguish handshake, application data, and alert messages
  - When AEAD key materials has yet to be decided, messages are transmitted in plaintext, e.g., ClientHello, ServerHello
- When AEAD is enabled
  - The additional data is provided with integrity, e.g., type, and length
  - The plaintext is protected with both confidentiality and integrity



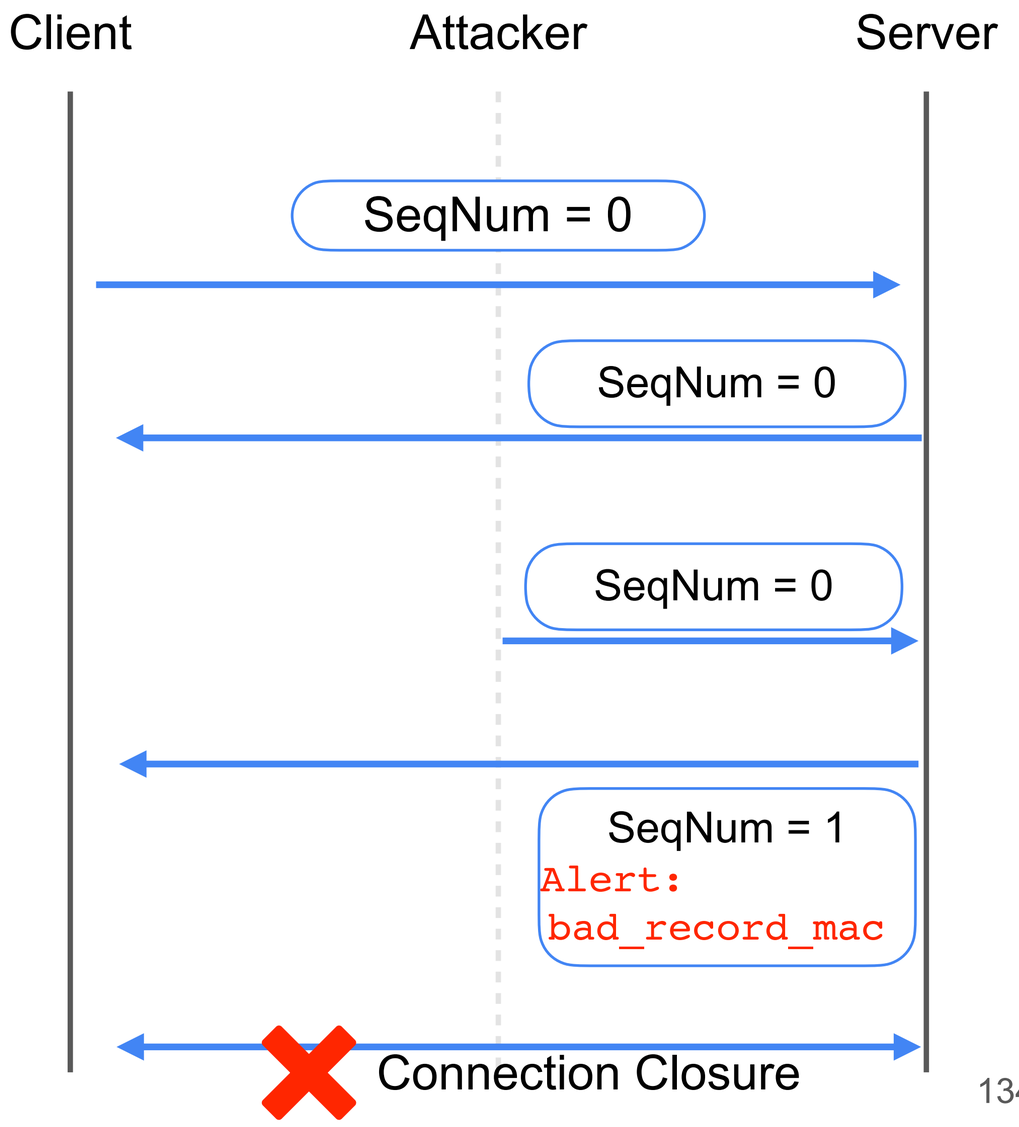
TLS Record Protocol

- What if an attacker replay an observed TLS record, e.g., “send 10K USD to Alice”?
  - The TLS record is valid, and a replay attack for records of the same connection is possible
- in TLS, we assign each TLS record a unique 64-bit sequence number, which is AEAD protected
  - Specifically, the sequence number is used to generate the AEAD nonce through XOR with the IV (initialization vector)
  - Replaying a AEAD-protected record will trigger decryption error
- What if errors occur?



TLS Alert Protocol

- Alert Types
  - Closure alerts
  - Error alerts
- Closure Alerts
  - close_notify: notifies the recipient that the sender will not send any more messages
- Error alerts
  - both parties MUST immediately close the connection, should an error alert is given



Key Issues of TLS 1.2

- Security
  - 37 cipher suites, and many are vulnerable to attacks
  - RSA key exchange doesn’t guarantee forward secrecy
  - The MAC-then-Encryption (MtE) is vulnerable to multiple practical attacks, e..g, padding oracle attacks for CBC mode encryption
- Performance
  - A latency of typical two RTTs is incurred for the TLS handshake

TLS 1.3 Changes

- Only 5 cipher suites are supported, rather than tens in TLS 1.2
- RSA no longer supported (only DHE)
  - Guarantees forward secrecy
  - Also, only 10 DHE groups are supported for (EC) DHE
- One-RTT handshake
  - Only one RTT for the handshake
  - The client sends key shares (e.g., $g^b \bmod p$) for preferred DHE groups in ClientHello
  - If the server agrees to use DHE, the server sends $g^a \bmod p$ (with signature) in ServerHello
- Signing all handshake messages in CertificateVerify
- Only supports AEAD for symmetric encryption and integrity
  - Recall AEAD (authenticated encryption with additional data): a block cipher mode that guarantees confidentiality and integrity at the same time

Limitations of TLS

TLS Does Not Provide Anonymity

- **Anonymity**: Hiding the client's and server's identities from attackers
- An attacker can figure out who is communicating with TLS
  - The client may indicate the name of the server in the ClientHello (called Server Name Indication, or SNI)
  - An attacker can see IP addresses and ports of the underlying IP and TCP protocols
- We have anonymity networks, e.g., Tor to achieve this security property

TLS Does Not Provide Availability

- **Availability**: Keeping the connection open in the face of attackers
- An attacker can stop a TLS connection
  - MITM can drop encrypted TLS packets
  - On-path attacker can still do RST injection to abort the underlying TCP connection
- Result: A TLS connection can still be censored
  - The censor can block TLS connections

TLS for Applications

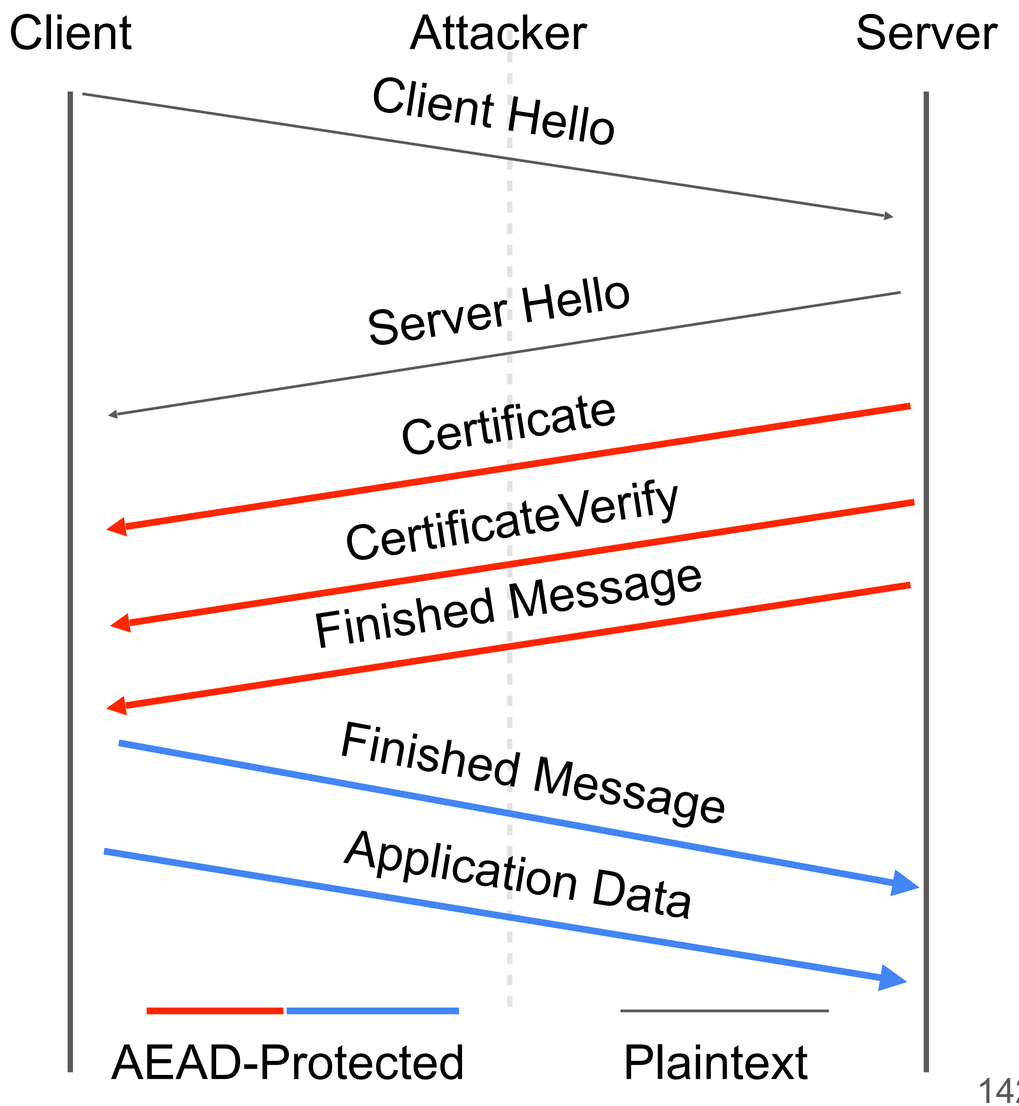
- Recall Internet layering: TLS provides services to higher layers (the application layer)
- **HTTPS**: The HTTP protocol run over TLS
  - In contrast, HTTP runs over plain TCP, with no TLS added
- Other secure application-layer protocols besides HTTPS exist
  - Example: Email protocol can use the STARTTLS command to uses TLS to secure communications
  - DNS over TLS to further anonymize traffic
- TLS does not defend against application-layer vulnerabilities
  - Example: SQL injection, XSS, CSRF, and buffer overflow vulnerabilities in the application are still exploitable over TLS

Stripping Attacks

- Browsers often default to using unencrypted HTTP
  - If a user types `google.com` into the browser, the browser opens `http://www.google.com`
  - To mitigate this, websites will often redirect from the HTTP to the HTTPS version of its site
  - **This requires the client to first receive the unprotected HTTP redirect response**
- **SSL stripping**: Forcing a user to use unencrypted HTTP instead of HTTPS
  - A MITM attacker intercepts the first HTTP request and creates their own HTTPS connection to the server
  - The user never receives a redirect to HTTPS, so it believes the site wants them to use HTTP
  - Defense: HTTP Strict-Transport-Security (HSTS) header tells browsers to only access the server with HTTPS

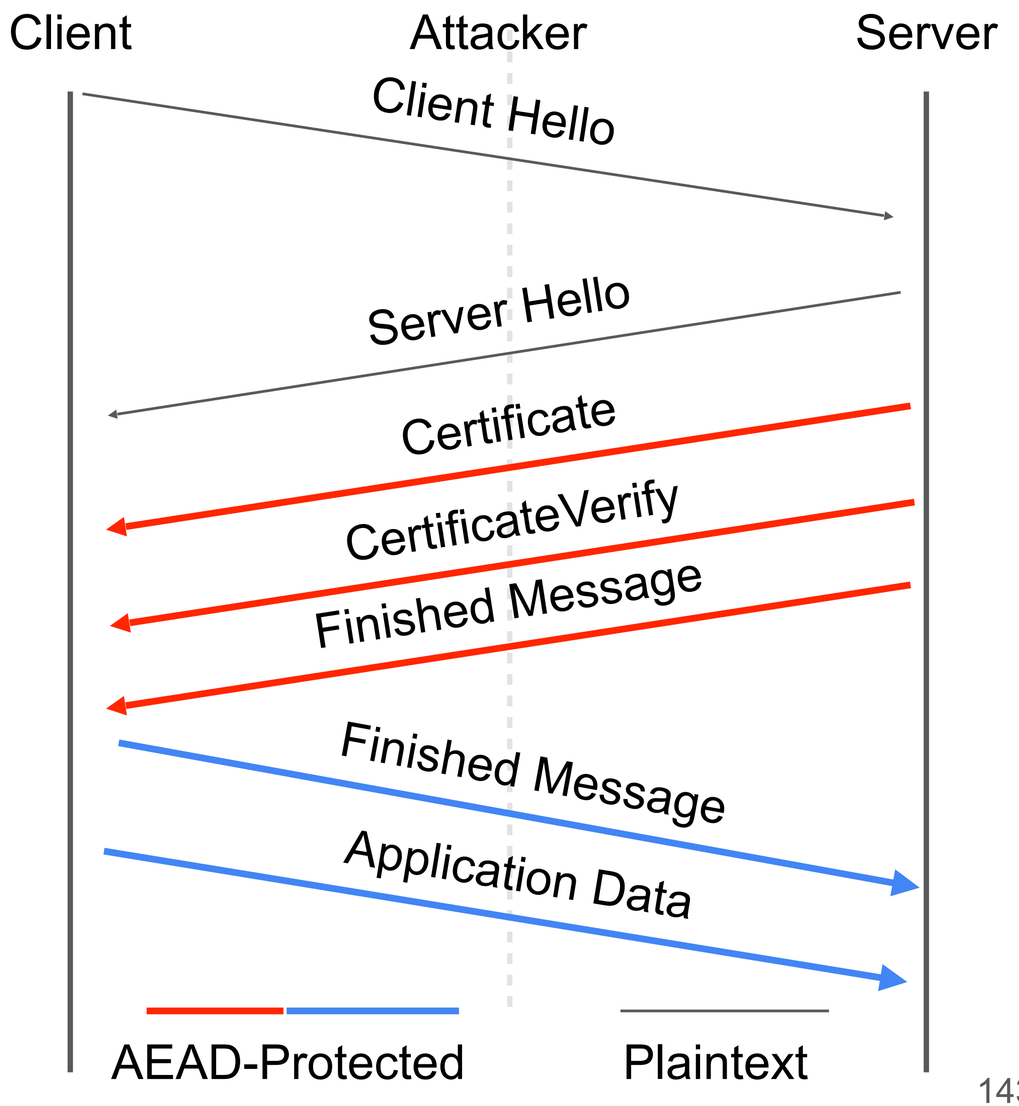
TLS: Summary

- How TLS works
  - Nonces make every handshake different (prevents replay attacks across connections)
  - RSA or DHE helps client and server agree on a pre-master secrete (PS)
    - RSA key exchange is deprecated in TLS 1.3
  - HKDF helps to derive key materials from PS
  - Signature (CertificateVerify) **authenticates** the server's identity
  - Finished message (MACing) verifies handshake integrity
  - AEAD simultaneously provides integrity and confidentiality
  - AEAD-protected record sequence number prevents replaying messages of the same connection



TLS: Summary

- Limitations of TLS
  - It doesn't provide anonymity
  - It does't provide availability
  - Real-world usage may still be vulnerable to stripping attacks
  - It doesn't defend against application-layer attacks



References of TLS

- The Transport Layer Security (TLS) Protocol Version 1.3
- A Detailed Look at RFC 8446 (a.k.a. TLS 1.3) by Nick Sullivan
- Computer Security by David Wagner, Nicholas Weaver, et al.
- Network Security of the CyBOK by Christian Rossow and Sanjay Jha

Q&A