Entity Authentication

Roel Peeters

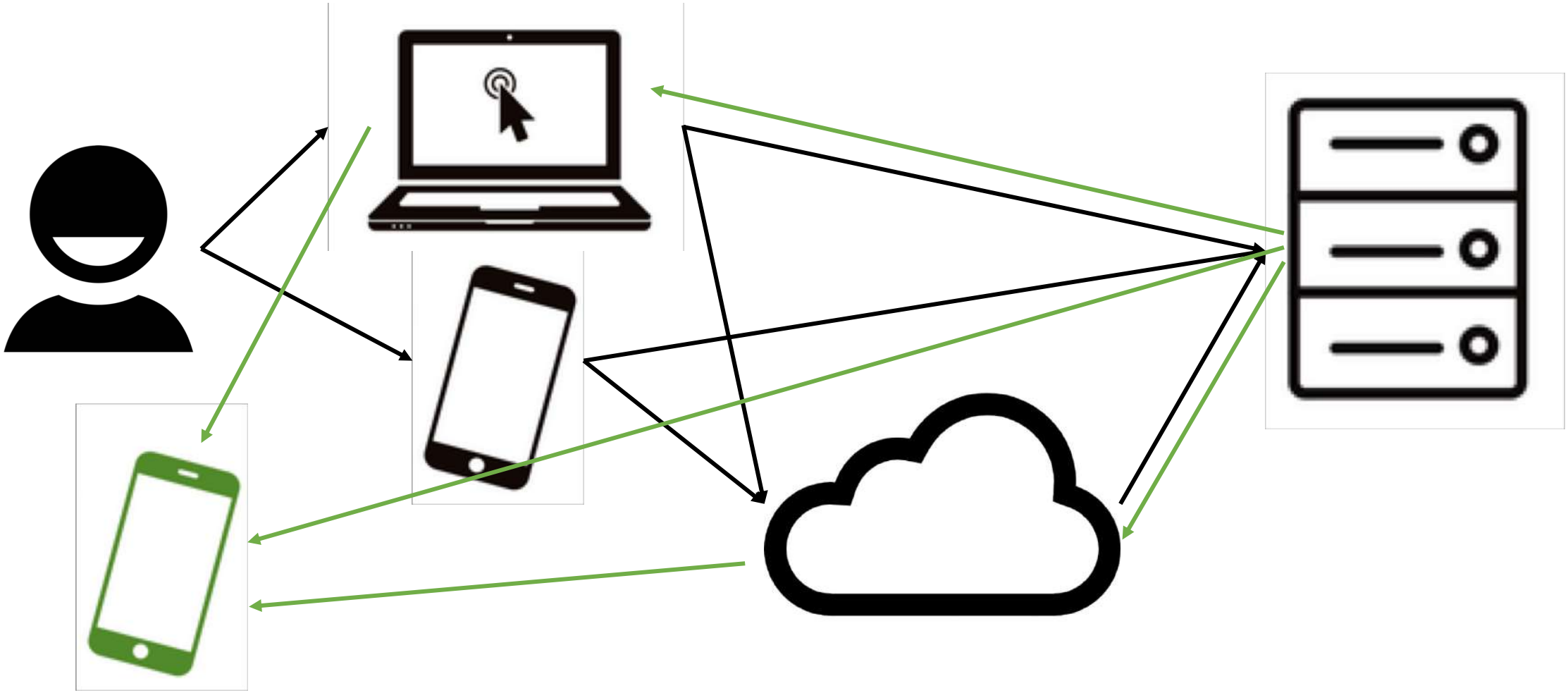
CTO nextAuth

roel.peeters@nextauth.com

Leuven, 18/02/2019

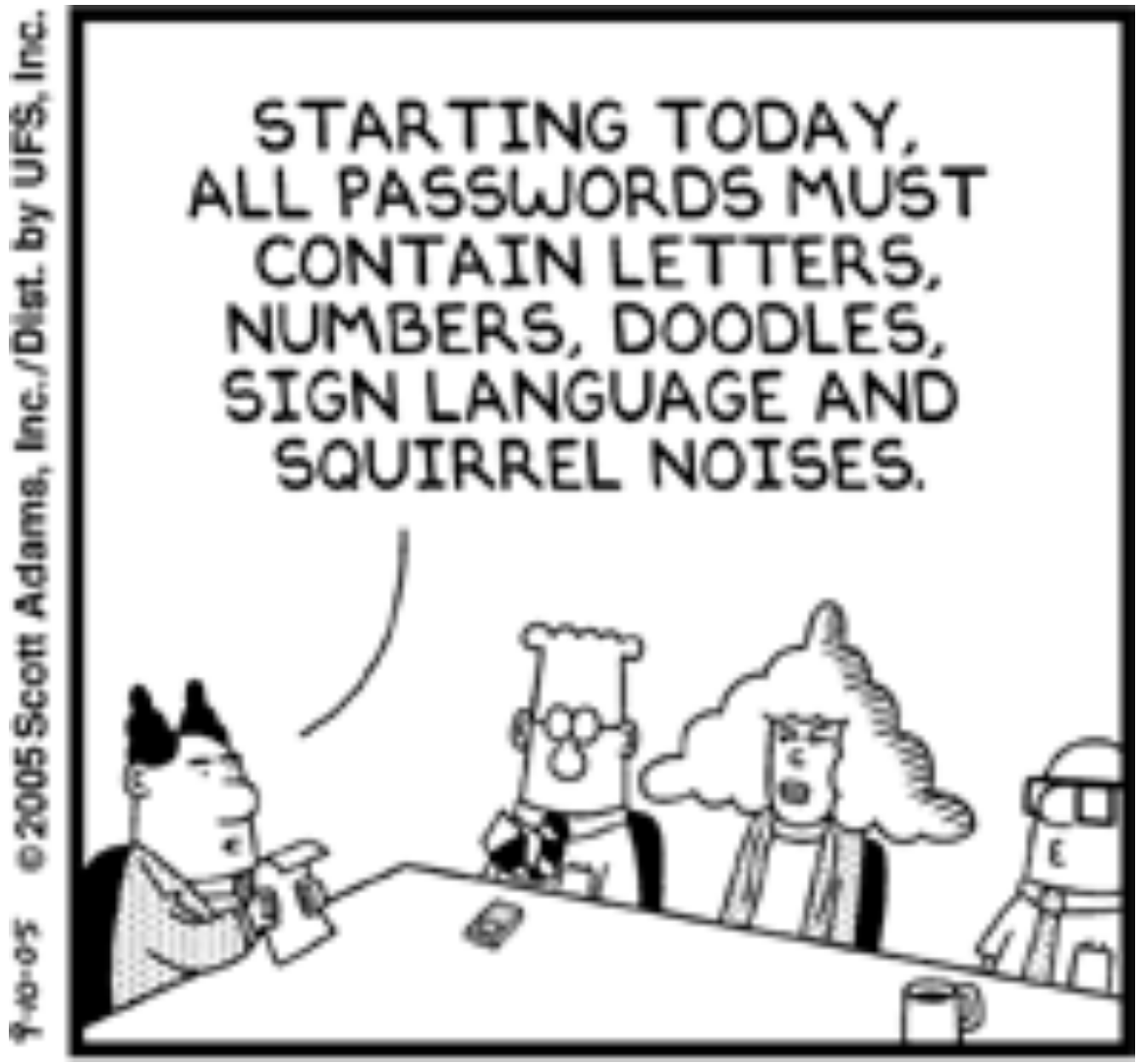

3 2 1
4
5
6
3 2 1
4
5
6
7

"On the Internet, nobody knows you're a dog."



Passwords

Something you know


STARTING TODAY, ALL PASSWORDS MUST CONTAIN LETTERS, NUMBERS, DOODLES, SIGN LANGUAGE AND SQUIRREL NOISES.
9-10-05 ©2005 Scott Adams, Inc./Dist. by UFS, Inc.


Cancel
Enter your new passcode
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
space
return

';--have i been pwned?

Check if you have an account that has been compromised in a data breach

email address or username | pwned?

220	3,805,757,030	50,533	48,199,630
pwned websites	pwned accounts	pastes	paste accounts

Top 10 breaches

593,427,119 Exploit.In accounts

457,962,538 Anti Public Combo List accounts

393,430,309 River City Media Spam List accounts

359,420,698 MySpace accounts

234,842,089 NetEase accounts

164,611,595 LinkedIn accounts

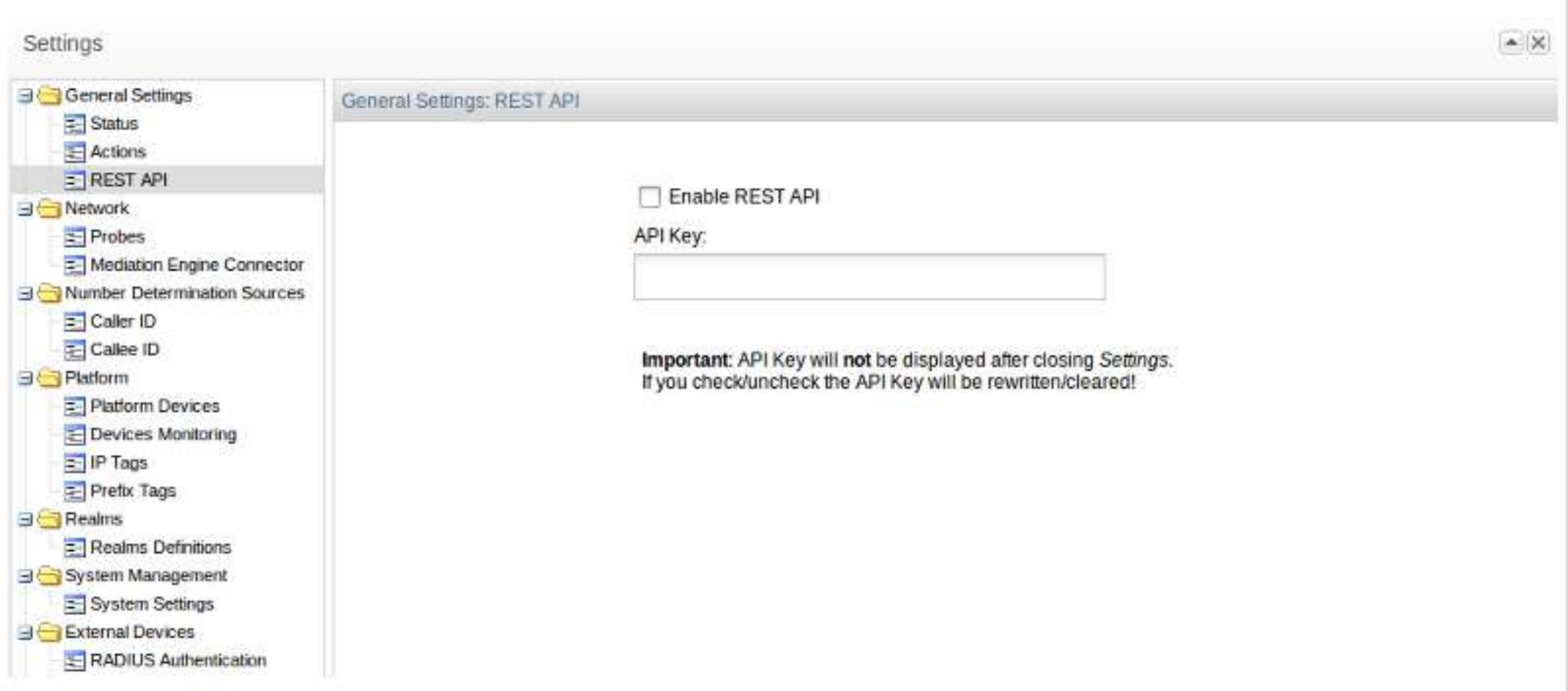
Improve security

- Use TLS
- if possible: TLS pinning
- Throttle guessing
- Hash and salt server password database

```
salt, H(password, salt)
```

- Iterative hashing, password hash functions: Argon2, scrypt, bcrypt, PBKDF2
- Bonus points: keyed hash of password before feeding it to password hash function (!key not stored along database)

API keys


Settings
General Settings
Status
Actions
REST API
Network
Probes
Mediation Engine Connector
Number Determination Sources
Caller ID
Callee ID
Platform
Platform Devices
Devices Monitoring
IP Tags
Prefix Tags
Realms
Realms Definitions
System Management
System Settings
External Devices
RADIUS Authentication
General Settings: REST API
Enable REST API
API Key:
Important: API Key will not be displayed after closing Settings.
If you check/uncheck the API Key will be rewritten/cleared!

Secure Remote Password (SRP) Protocol (1996)

host password verifier $v = g^x$

generate random value a — generate random value b

$username,\ A = g^a$ →

← $salt,\ B = kv + g^b$

$u = H(A, B)$ — $u = H(A, B)$

$S_{Carol} = (B - kg^x)^{(a + ux)}$, where $x=H(password, salt)$ — $S_{Steve} = (Av^u)^b$

$K_{Carol} = H(S_{Carol})$ — $K_{Steve} = H(S_{Steve})$

PACE

Static domain parameters D_{PICC}

1 MSE: Set AT	Select parameters and password π (MRZ/CAN/PIN/PUK) Certificate Holder Authorization Template (in EACv2)	
	OK	
2 General Authenticate	Empty Data Object	
	Choose random nonce s	
	$z = E(K_\pi, s)$	
		$s = D(K_\pi, z)$
3 General Authenticate	Data Required for MapToPoint()	
	Data Required for MapToPoint()	
	$D' = MapToPoint(D_{PICC}, s, data)$ Choose values for DH using D'	$D' = MapToPoint(D_{PICC}, s, data)$ Choose values for DH using D'
4 General Authenticate	DH Value DH_{PCD}	
	DH Value DH_{PICC}	
	K = DH Key $KS_{MAC} = KDF_{MAC}(K)$ $KS_{ENC} = KDF_{ENC}(K)$	K = DH Key $KS_{MAC} = KDF_{MAC}(K)$ $KS_{ENC} = KDF_{ENC}(K)$
5 General Authenticate	$T_{PCD} = MAC(KS_{MAC}, DH_{PICC})$	
	Verify T_{PCD}	
	$T_{PICC} = MAC(KS_{MAC}, DH_{PCD})$	
		Verify T_{PICC}

ISO 7816-4 commands

PINs

Something you know

	PIN	Freq
#1	1234	10.713%
#2	1111	6.016%
#3	0000	1.881%
#4	1212	1.197%
#5	7777	0.745%
#6	1004	0.616%
#7	2000	0.613%
#8	4444	0.526%
#9	2222	0.516%
#10	6969	0.512%
#11	9999	0.451%
#12	3333	0.419%
#13	5555	0.395%

Improve security

- **Hard throttle guessing !!!**
- HSM/smart card (cannot hash and salt PIN: too little entropy)

One Time Passwords

Something you have

You have selected as your new Apple ID. To verify this email address belongs to you, enter the code below on the email verification page:

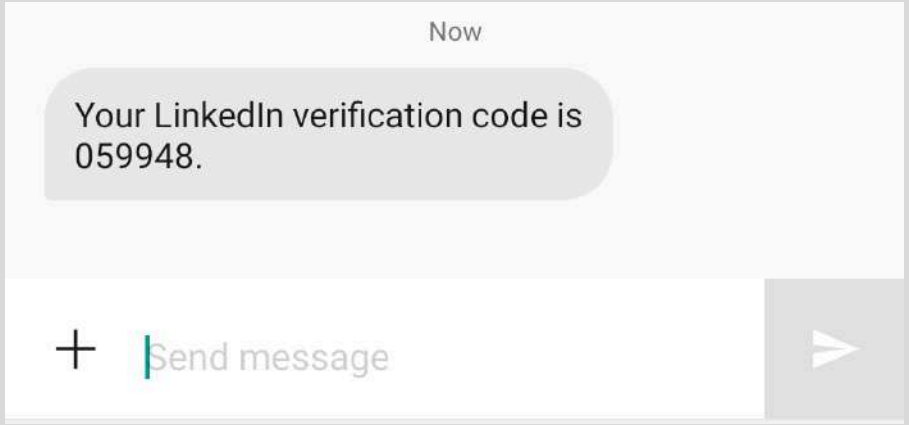
894567

This code will expire three hours after this email was sent.

Why you received this email.
Apple requires verification whenever an email address is selected as an Apple ID. Your Apple ID cannot be used until you verify it.

If you did not make this request, you can ignore this email. No Apple ID will be created without verification.

Apple Support



NIST Special Publication 800-63B

Digital Identity Guidelines *Authentication and Lifecycle Management*

The out-of-band device SHOULD be uniquely addressable and communication over the secondary channel SHALL be encrypted unless sent via the public switched telephone network (PSTN). **Methods that do not prove possession of a specific device, such as voice-over-IP (VOIP) or email, SHALL NOT be used for out-of-band authentication.**

The out-of-band authenticator SHALL uniquely authenticate itself in one of the following ways when communicating with the verifier:

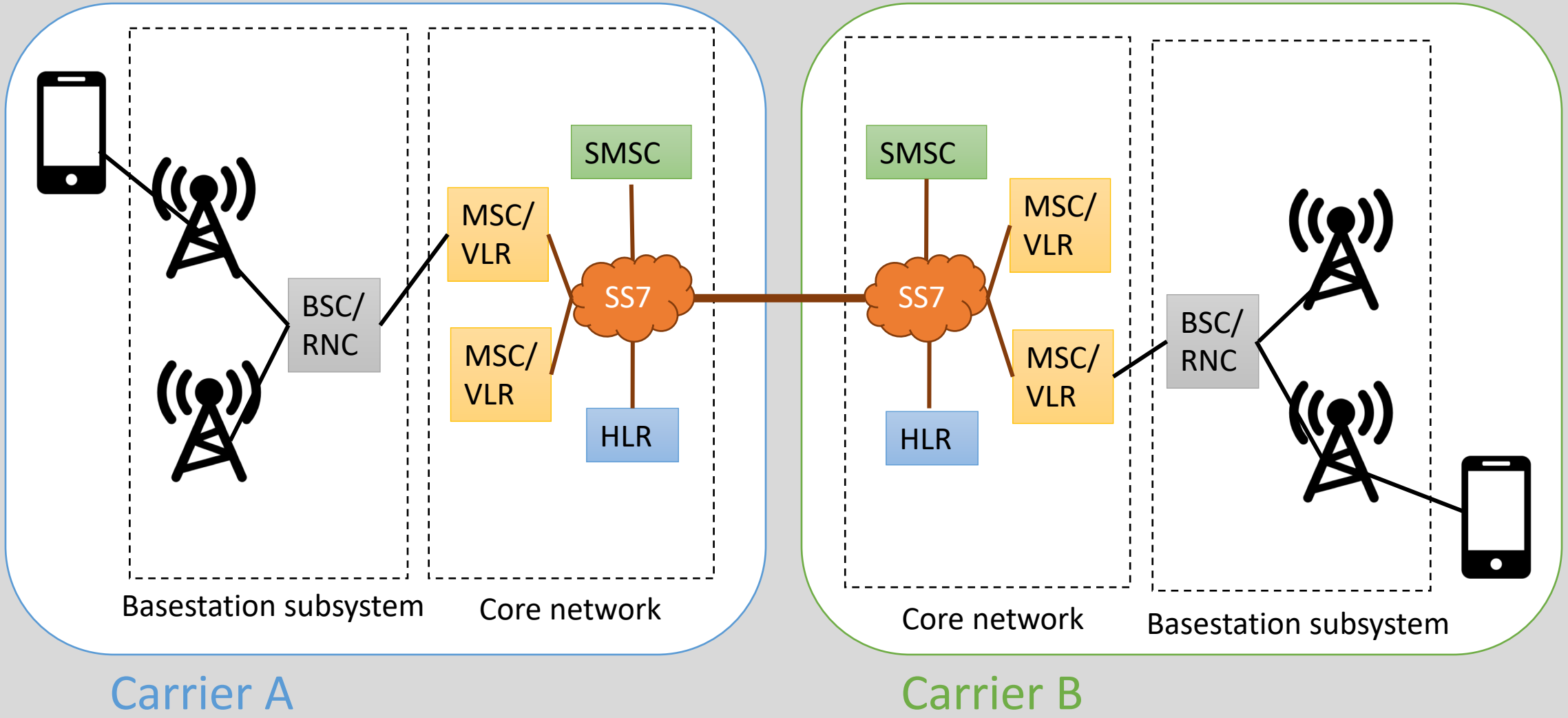
- Establish an authenticated protected channel to the verifier using approved cryptography. […]
- **Authenticate to a public mobile telephone network using a SIM card or equivalent** that uniquely identifies the device. This method SHALL only be used if a secret is being sent from the verifier to the out-of-band device via the **PSTN (SMS or voice)**.

If a secret is sent by the verifier to the out-of-band device, **the device SHOULD NOT display the authentication secret while it is locked** by the owner (i.e., requires an entry of a PIN, passcode, or biometric to view). However, authenticators SHOULD indicate the receipt of an authentication secret on a locked device.

5.1.3.3 Authentication using the Public Switched Telephone Network

Use of the PSTN for out-of-band verification is **RESTRICTED**. If out-of-band verification is to be made using the PSTN, the verifier SHALL verify that the pre-registered telephone number being used is associated with a specific physical device.

SS7 hacks

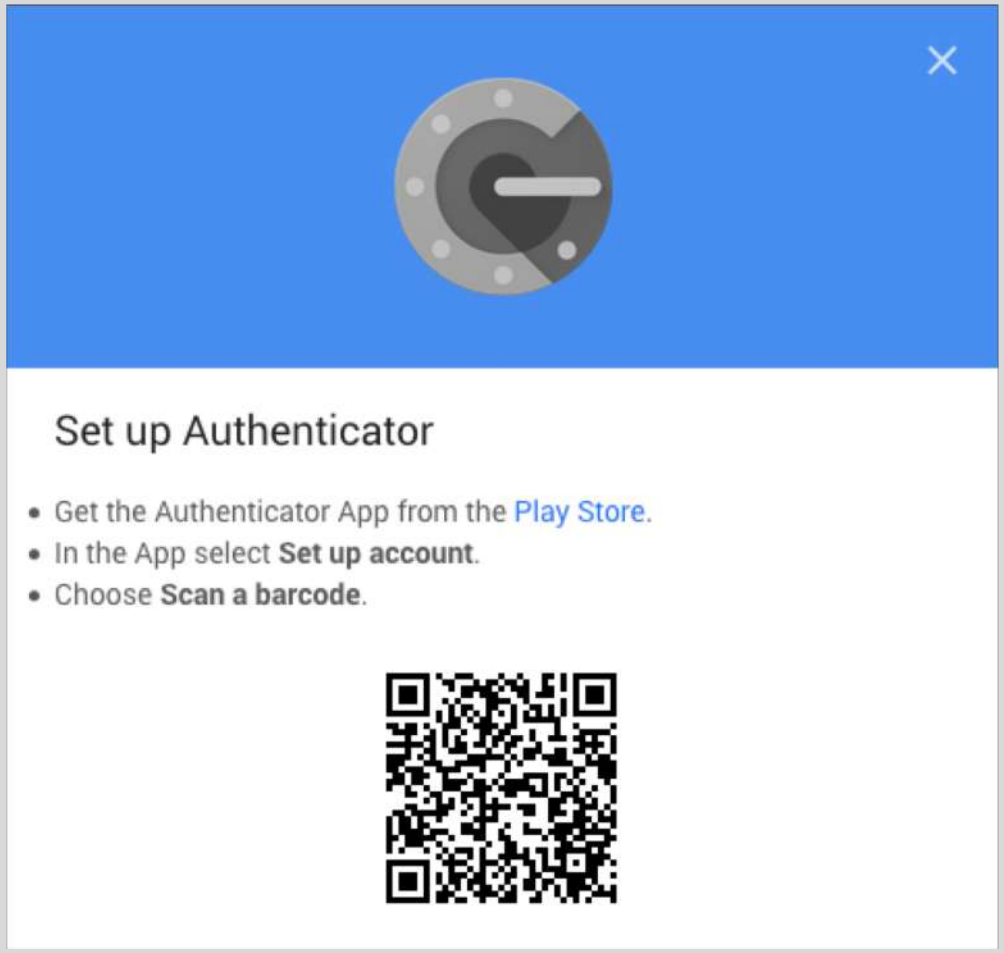

SMSC
MSC/
VLR
MSC/
VLR
SS7
HLR
BSC/
RNC
Basestation subsystem
Core network
SMSC
MSC/
VLR
MSC/
VLR
SS7
HLR
BSC/
RNC
Core network
Basestation subsystem
Carrier A
Carrier B

Symmetric Key Cryptography

Challenge-response: key cannot be learned from communication

Both server and authenticator need to store shared secret key

protected storage at server (HSM)



Chip Authentication Programme (CAP)* (2004)

The CAP standard is **secret** and so not subject to scrutiny, despite being a critical security component the public must rely on for banking transactions.

CAP operates in three modes – identify, respond, and sign. For all three modes a PIN is required first. Thereafter, **identify** just returns a one-time code; for **respond** a numerical challenge is required; and for **sign** an account number and a value are needed.

The numerical response code is a compressed version of a MAC (3DES CBC) computed by the card under its key; it is calculated over the information entered by the customer, and a transaction counter.

* CAP is the MasterCard brand; Visa's version is called Dynamic Passcode Authentication (DPA)

HOTP: HMAC-based One Time Password Algorithm (2005)

We can describe the operations in 3 distinct steps:

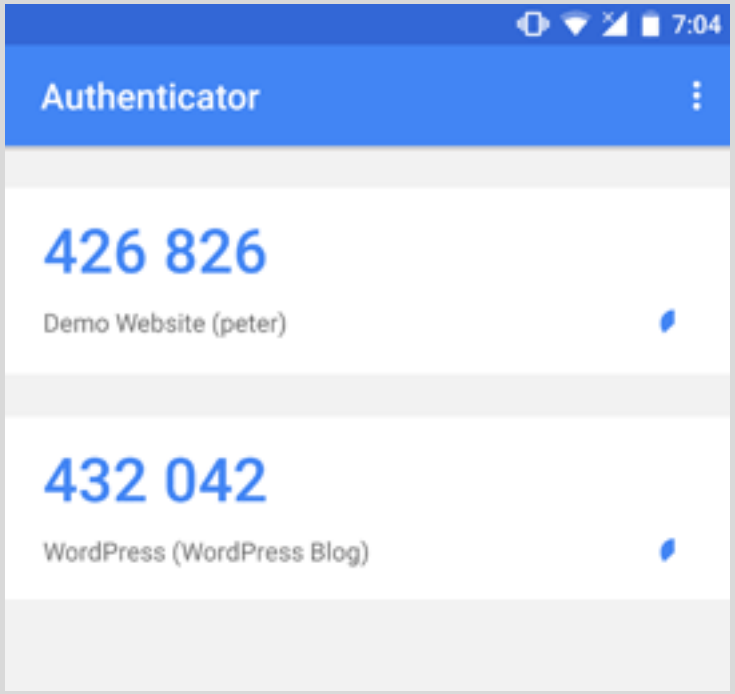
- Step 1: Generate an HMAC-SHA-1 value
  Let HS = HMAC-SHA-1(K,C) *HS is a 20-byte string*
- Step 2: Generate a 4-byte string (Dynamic Truncation)
  Let Sbits = DT(HS) *DT is a 31-bit string*
- Step 3: Compute an HOTP value
  Let Snum = StToNum(Sbits) *Convert S to a number in 0...2^{31}-1*
  Return D = Snum mod 10^Digit *D is a number in 0...10^{Digit}-1*

Implementations **MUST extract a 6-digit code at a minimum and possibly 7 and 8-digit code.** Depending on security requirements, Digit = 7 or more SHOULD be considered in order to extract a longer HOTP value.

TOTP: Time-based One Time Password Algorithm (2011)

TOTP is the time-based variant of this algorithm, where a value T, derived from a time reference and a time step, replaces the counter C in the HOTP computation.

TOTP implementations **MAY use HMAC-SHA-256 or HMAC-SHA-512 functions,** based on SHA-256 or SHA-512 [SHA2] hash functions, instead of the HMAC-SHA-1 function that has been specified for the HOTP computation in [RFC4226].



OCRA: OATH Challenge-Response Algorithm (2011)

We list the following **preferred modes of computation:**

- HOTP-SHA1-4: HOTP with SHA-1 as the hash function for HMAC and a dynamic truncation to a 4-digit value; this mode is not recommended in the general case, but it can be useful when a very short authentication code is needed by an application
- **HOTP-SHA1-6: HOTP with SHA-1 as the hash function for HMAC and a dynamic truncation to a 6-digit value**
- HOTP-SHA1-8: HOTP with SHA-1 as the hash function for HMAC and a dynamic truncation to an 8-digit value
- HOTP-SHA256-6: HOTP with SHA-256 as the hash function for HMAC and a dynamic truncation to a 6-digit value
- HOTP-SHA512-6: HOTP with SHA-512 as the hash function for HMAC and a dynamic truncation to a 6-digit value

OCRA: OATH Challenge-Response Algorithm

This table summarizes all possible values for the CryptoFunction:

Name	HMAC Function Used	Size of Truncation (t)
HOTP-SHA1-t	HMAC-SHA1	0 (no truncation), 4-10
HOTP-SHA256-t	HMAC-SHA256	0 (no truncation), 4-10
HOTP-SHA512-t	HMAC-SHA512	0 (no truncation), 4-10

Table 1: CryptoFunction Table

Bruteforce security

28 bits

Password

48 bits

Password + OTP

20 bits

only (push) OTP

Public Key Cryptography

Something you have

Public Key Cryptography

Server only has public keys (verification key)

Client has private keys (signing key)

where and how are the private keys generated, **randomness**

Full length (RSA: 1024-4096 bit, ECC: 512-1024 bit) , cannot be truncated

often PKI, but not necessary

Biometrics

Something you are

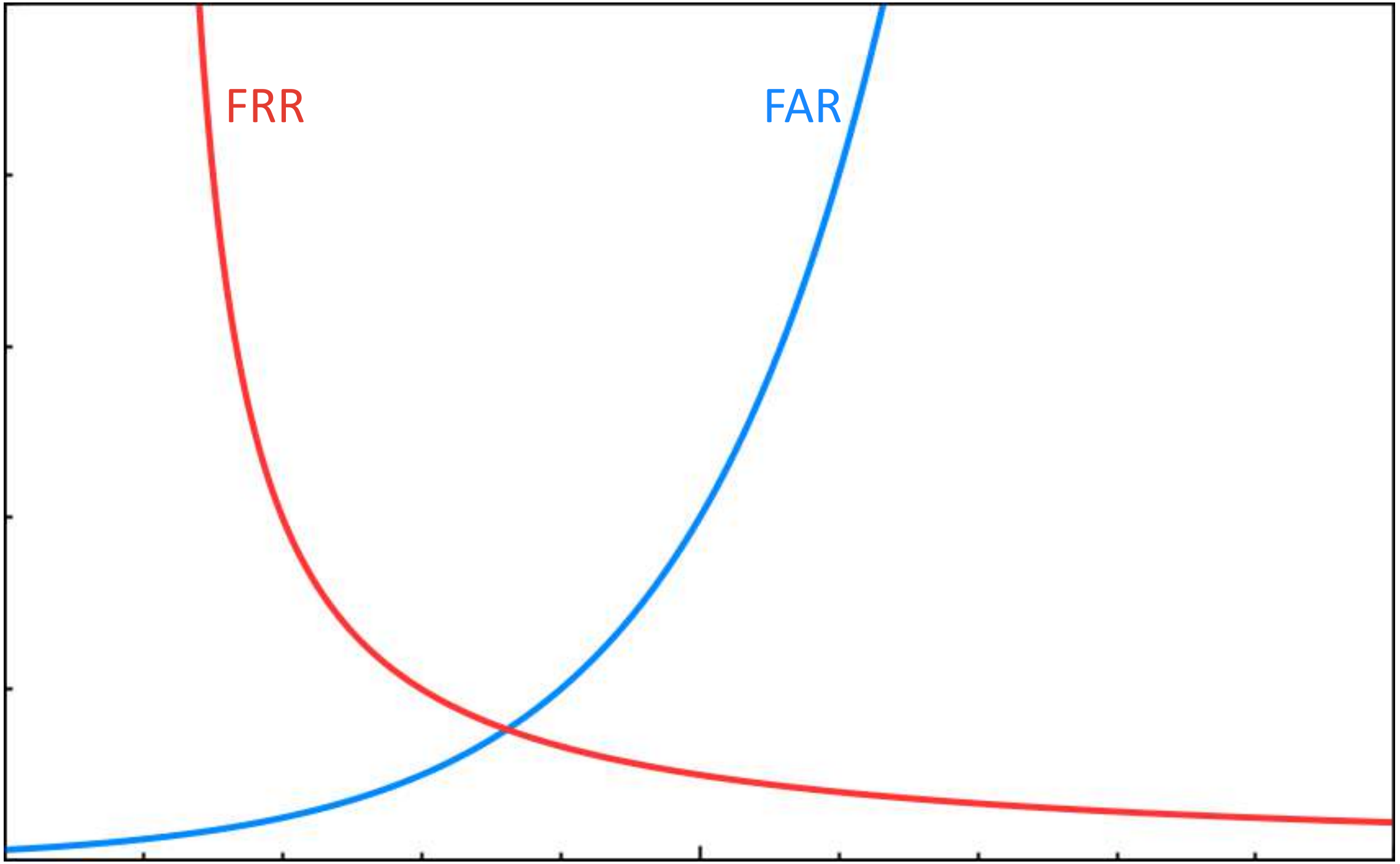
Biometrics

- **Fingerprint**: minutia, texture, vain patterns
- Iris
- **Face**
- Hand: geometry, palm print
- Ears
- Voice
- Gate
- Signature
- Typing password
- …

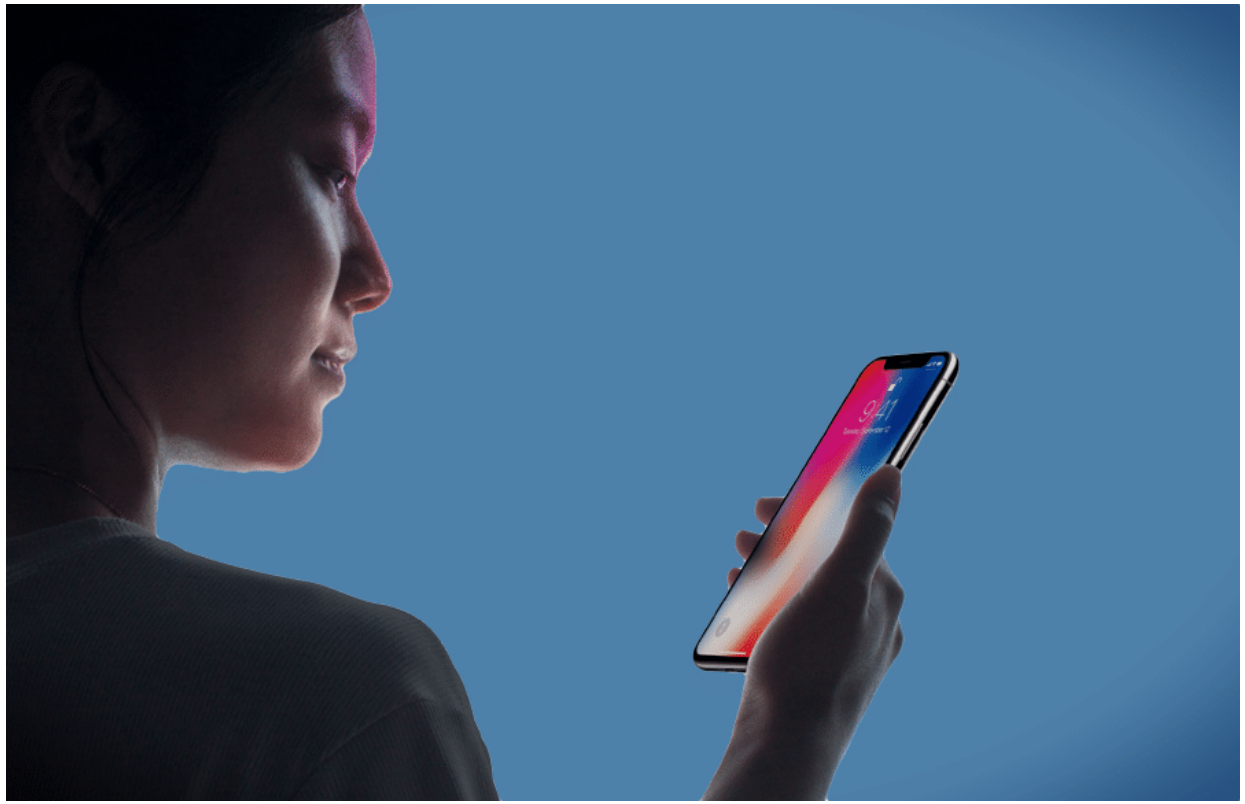
Biometrics

- Convenient, nothing that can be forgotten
- No shoulder surfing, no guessing

- Reproducibility
- Intersession variability
- Environment
- Lifeness
- Aging
- Privacy
- **Always have a fallback mechanism: accidents do happen**
- **DO NOT STORE PASSWORD/PIN UNDER BIOMETRIC**


FRR
FAR

One-to-one vs one-to-many



13.9.2018 09:10

OP Financial Group first in Finland to pilot facial recognition payments

OP's employees will be the first in Finland to trial facial recognition payments at the firm's staff restaurant in OP's premises in Vallila, Helsinki. The payment ecosystem is being rapidly disrupted, and facial recognition payments are expected to be the next big trend.

Facial recognition payments are based on biometric identification. The technology compares the customer's face to a face map captured on camera. After the customer's face has been identified, the payment itself is simple: no mobile phone, payment card, cash or other traditional payment method will be needed. This makes paying quick and easy.

– Facial recognition payments aren't really being used in Finland or even in the Western world, but at OP we believe in its ease of use and reliability. We are conducting the trial to better understand how the technology could be applied going forward, says **Harri Nummela**, OP Executive Vice President, Banking, Private and SME customers.

Facial recognition payments are expected to be the next big global trend in payments. Customers have been very pleased with facial recognition payment in international pilots. The technology used in facial recognition payment can be used in other applications too.

– For example in China, the technology is used to identify customer loyalty benefits and in access control. We can also see broader opportunities for application. As the technology is new, it is important to collect feedback on any fears and apprehensions users may have. Based on what we learn, we will then be able to take the right next steps in development, says **Kristian Luoma**, Head of OP Lab.

The pilot is already underway at OP premises in Vallila, Helsinki, and employees have enthusiastically welcomed the new payment method. The users participating in the pilot can currently use facial recognition when paying for e.g. lunch.

Supervised vs unsupervised

Privacy

Match on card (or biometric comparison according to ISO)

Protected templates

- Original biometric is not recoverable
- Renewability

- Template needs to be stored (securely), which could still contain privacy sensitive information (can be mitigated with second factor)

Physically Unclonable Functions (PUFs)

Physical properties of the device

Two main modes:

- Challenge-response directly
- Fuzzy extractor: derive cryptographic key

DeepMasterPrints: Generating MasterPrints for Dictionary Attacks via Latent Variable Evolution*

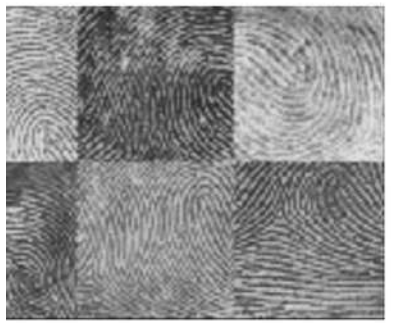
Philip Bontrager
New York University Tandon
philipjb@nyu.edu

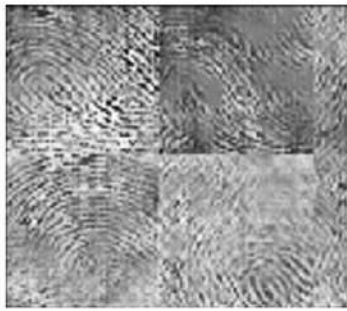
Aditi Roy
New York University Tandon
ar3824@nyu.edu

Julian Togelius
New York University Tandon
julian@togelius.com

Nasir Memon
New York University Tandon
memon@nyu.edu

Arun Ross
Michigan State University
rossarun@cse.msu.edu





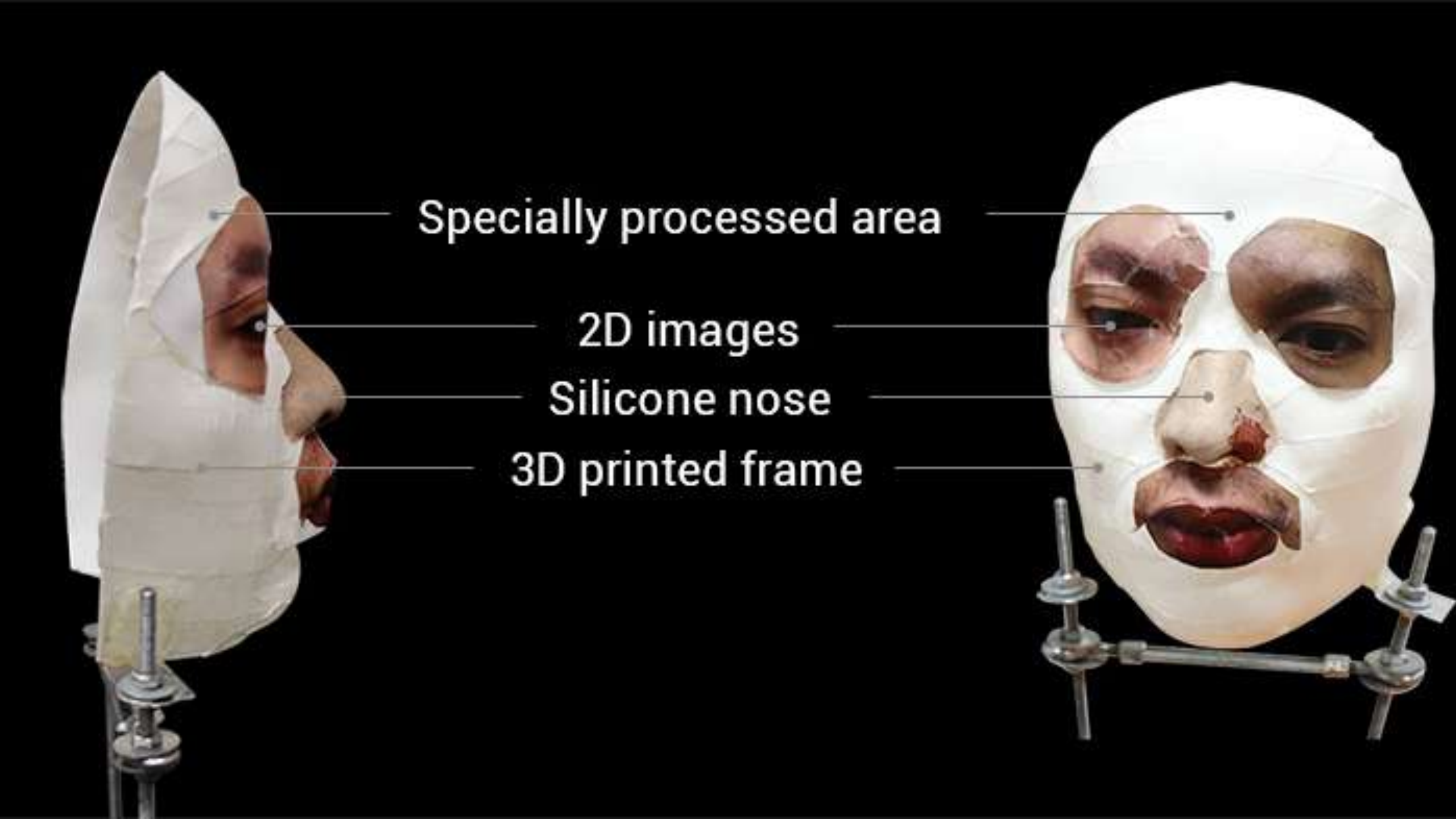
Abstract

Recent research has demonstrated the vulnerability of fingerprint recognition systems to dictionary attacks based on MasterPrints. MasterPrints are real or synthetic fingerprints that can fortuitously match with a large number of fingerprints thereby undermining the security afforded by fingerprint systems. Previous work by Roy et al. generated synthetic MasterPrints at the feature-level. In this work we generate complete image-level MasterPrints known as DeepMasterPrints, whose attack accuracy is found to be much superior than that of previous methods. The proposed method, referred to as Latent Variable Evolution, is based on training a Generative Adversarial Network on a set of

user's fingerprint. Since small portions of a fingerprint are not as distinctive as the full fingerprint, the chances of a partial fingerprint (from one finger) being incorrectly matched with another partial fingerprint (from a different finger) are higher. This observation was exploited by Roy *et al.* [25], who introduced the notion of *MasterPrints*. MasterPrints are a set of real or synthetic fingerprints that can fortuitously match with a large number of other fingerprints. Therefore, they can be used by an adversary to launch a dictionary attack against a specific subject that can compromise the security of a fingerprint-based recognition system. This means, it is possible to "spoof" the fingerprints of a subject without actually gaining any information about the subject's fingerprint.

Roy *et al.* [25] demonstrated that MasterPrints can either


A CONTACT LENS IS PLACED ON THE
PRINTED INFRARED IMAGE


Specially processed area
2D images
Silicone nose
3D printed frame

Behaviourometrics

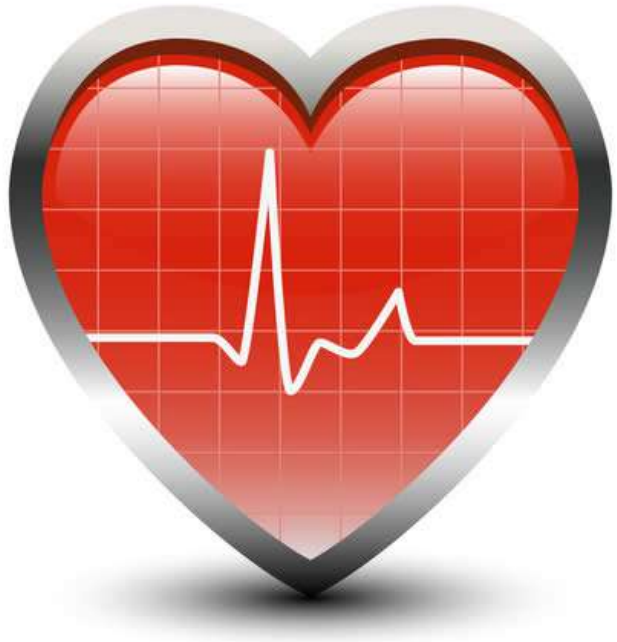
Something you do

Behaviourmetrics

- Voice
- Airsignature
- Continious user monitoring
- …

Continuous user monitoring

- Usage patterns
- Typing patterns
- Body sensors
- …



Risk based

Something else

Risk based = risky

- Rooted
- Tampered
- Device fingerprinting
- Other apps
- Location
- Contacts
- …

Conclusion

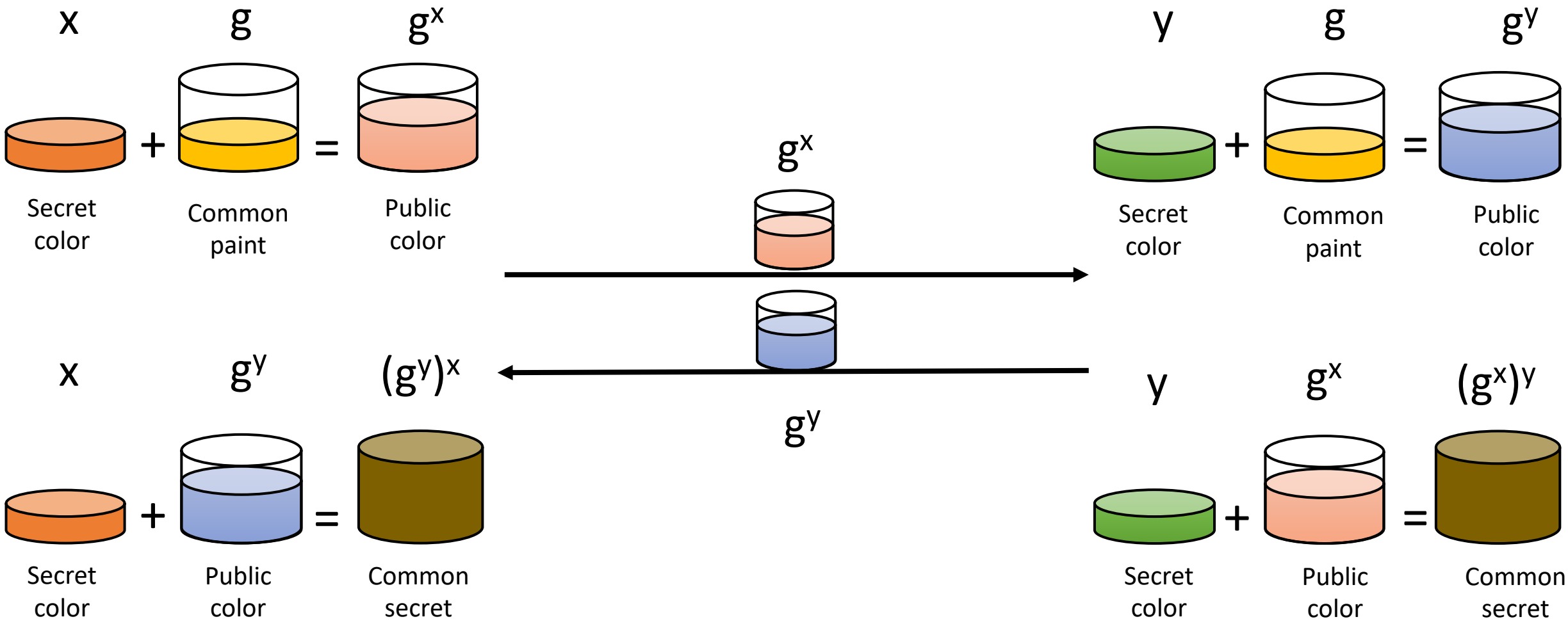
Each factors on their own has advantages and disadvantages

Combine them

Fallback and recovery methods

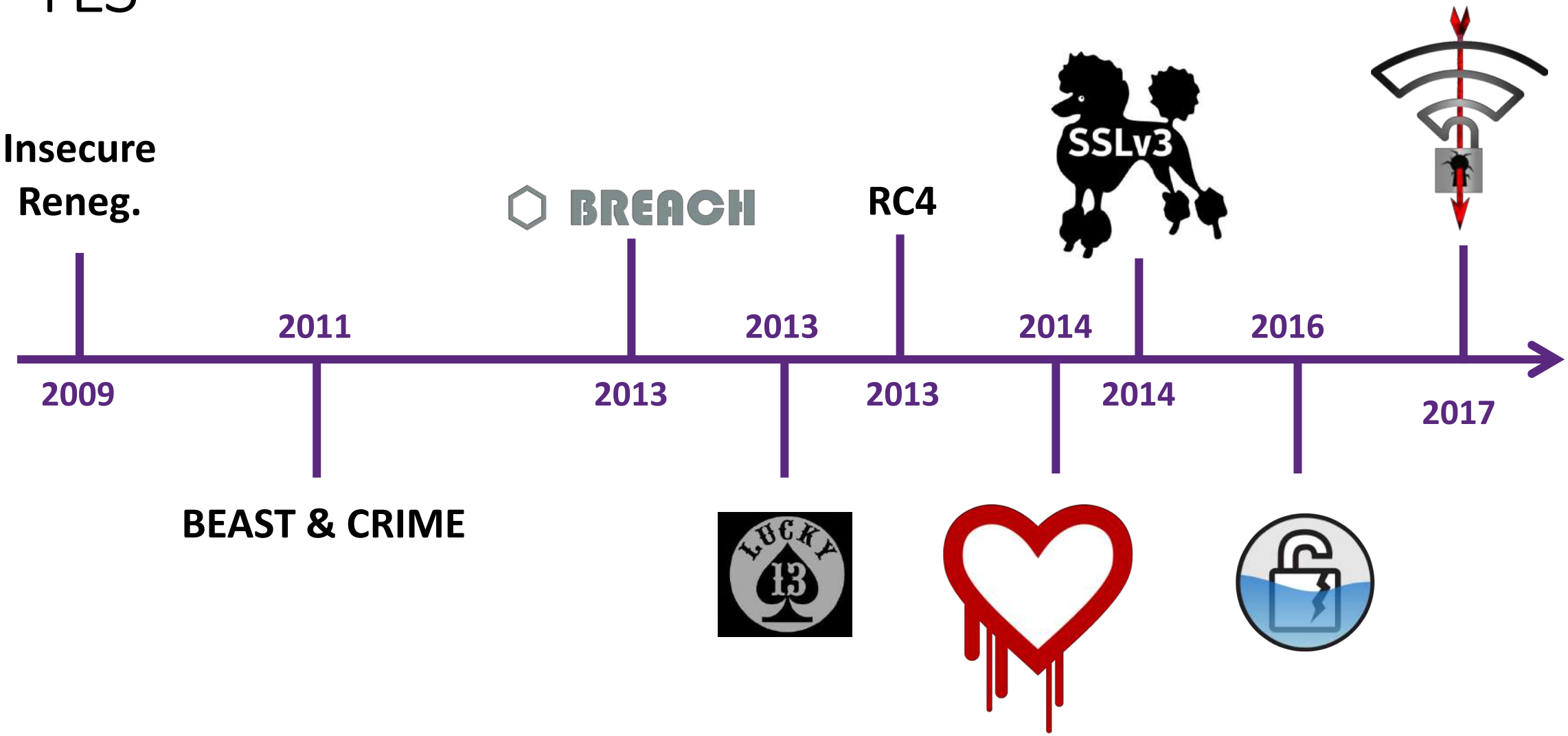
Protocols

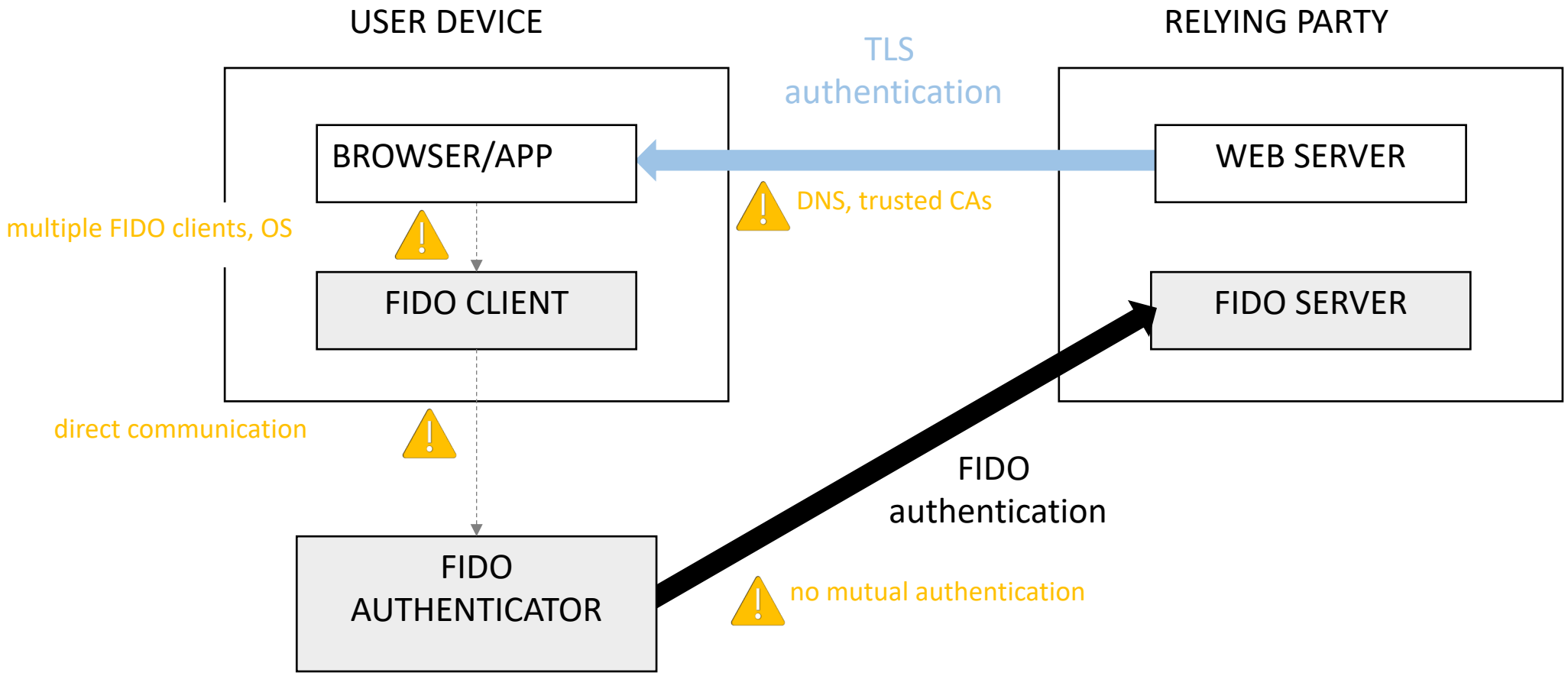
Diffie-Hellman



TLS

TLS

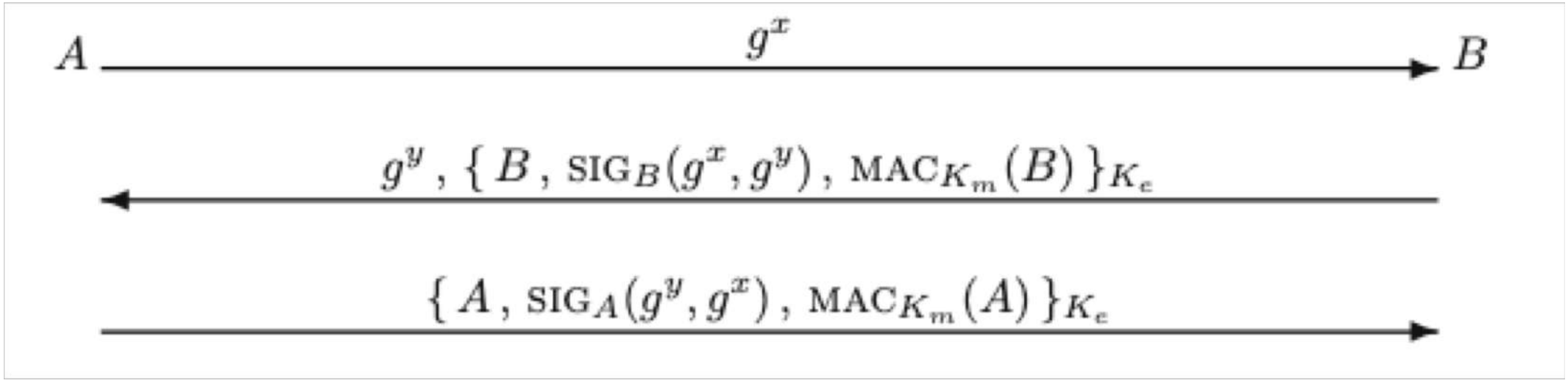

Insecure Reneg.
2009
2011
BEAST & CRIME
BREACH
2013
2013
LUCKY 13
RC4
2013
SSLv3
2014
2014
2016
2017


USER DEVICE
RELYING PARTY
TLS
authentication
BROWSER/APP
WEB SERVER
DNS, trusted CAs
multiple FIDO clients, OS
FIDO CLIENT
FIDO SERVER
direct communication
FIDO
authentication
FIDO
AUTHENTICATOR
no mutual authentication

SIGn-and-MAc (SIGMA) protocol

Cryptographically proven

3 variants: basic SIGMA, **SIGMA-I**, SIGMA-R



Future of entity authentication

Passwords will stay, but become more and more deprecated

Public keys cryptography will become more dominant

User authentication will be multi factor

Going from one-shot authentication to continuous authentication