UNIVERSITY OF SURREY

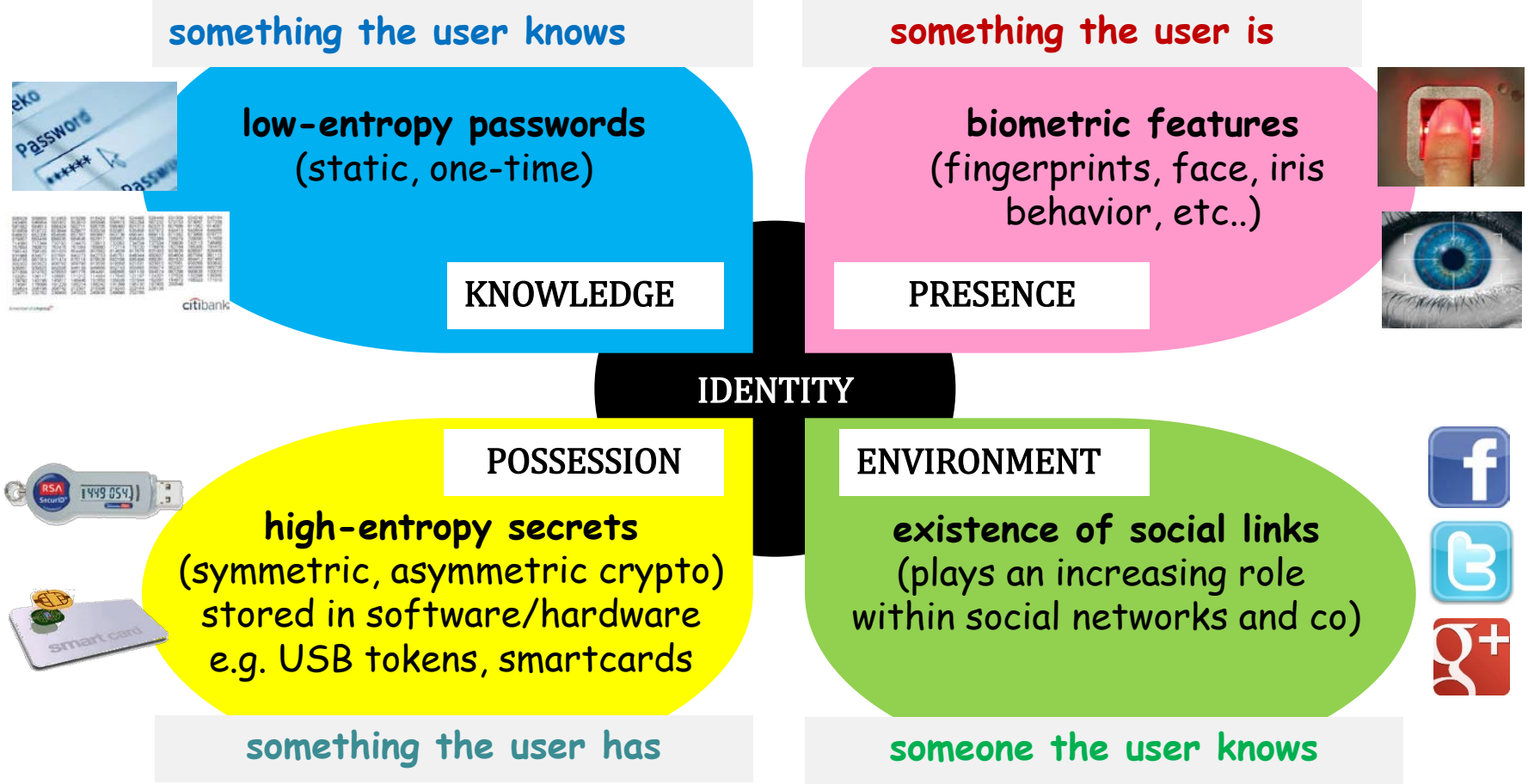
A Modular Framework for Multi-Factor Authentication & Key Exchange

Nils Fleischhacker, Mark Manulis, Amir Azodi

SSR 2014@RHUL December 17, 2014

Multi-Factor Authentication

MFA (client) authentication by combination of 2+ authentication factors
aims to guarantee security if one of the factors remains uncompromised



MFA standards and products

UNIVERSITY OF SURREY

MFA patents and standards

- PCI Data Security Standard, Ver. 2, 2010 (payment card industry)
- NIST Special Publication 800-63, 2011
- HMAC-based One-Time Password (HOTP, RFC 4226)
- Time-based One-Time Password (TOTP, RFC 6238)
- FIDO Alliance Universal Authentication Framework (UAF), Oct 2014

MFA(KE) products

- 2FA websites: https://twofactorauth.org/
- OATH HOTP/TOTP
- RSA SecureID
- Google Authenticator, ...

MFA research highlights

UNIVERSITY OF SURREY

- various MFA(KE) protocols
  - e.g. [HAD06, A07, BSS+09, D09, FL09]
  - mostly loose security definitions and partially missing analysis
- first MFAKE model by Pointcheval and Zimmer [PZ08]
  - three factors (1 password, 1 private key, 1 biometric)
  - liveness assumption (for biometrics) for the client
  - optional pk-based server authentication
  - impersonation attacks found by Hao and Clarke [HC12]
- provably secure MFAKE protocol by Stebila, Udupi, and Chang [SUC10]
  - n factors as a mix of (one-time) passwords and symmetric keys
  - pk-based server authentication

Example MFAKE protocol [PZ08]

Three factors biometric template $W_C = (w_1, ..., w_n)$, each w_i is a bit

private/public key $(sk, pk = g^{sk})$

password pw

$W_C = (w_1, ..., w_n)$ sk pw		$\{g^{r_i}, pk^{r_i} g^{w_i}\}_i$ $pk = g^{sk}$ pw
	$g^b v^{pw}$ →	
	$\{g^{t_i} g^{r_i}\}_i , g^a u^{pw}$ ←	
$K = (g^a u^{pw} / u^{pw})^b = g^{ab}$		$K = (g^b v^{pw} / v^{pw})^a = g^{ab}$
$\{K_i = (g^{t_i} g^{r_i})^{sk} g^{w_i}\}_i$		$\{K_i = pk^{t_i} pk^{r_i} g^{w_i}\}_i$
	$\{H(K, pw, K_i, ...)\}_i$ →	
		accept C if at least τ hash values match: $\{H(K, pw, K_i, ...)\}_i$

Example MFAKE protocol [PZ08]

- attack in [HC12] shows that knowledge of pw can be used to reveal W_C
- this shows that it is <u>dangerous to mix authentication factors</u>

$W_C = (w_1, ..., w_n)$ sk pw		pk pw
	$g^b v^{pw}$ →	
	← $\{g^{s_i}\}_i$, $g^a u^{pw}$	
$K = (g^a u^{pw} / u^{pw})^b = g^{ab}$		$K = (g^b v^{pw} / v^{pw})^a = g^{ab}$
$\{K_i = (g^{s_i})^{sk} g^{w_i}\}_i$		$\{K_{0,i} = pk^{s_i} g^0\}_i$ $\{K_{1,i} = pk^{s_i} g^1\}_i$
	$\{H(K, pw, K_i, ...)\}_i$ →	
		find w_i by comparing hashes with $H(K, pw, K_{0,i} ...)\}$ or $H(K, pw, K_{1,i} ...)$

Aiming at Modular MFA Framework

UNIVERSITY OF SURREY

Advantage through modularity

- cleaner design
  - each factor is processed independently of any other factor
  - helps to avoid problems with “mixing” factors
- more flexible
  - backwards compatibility with single-factor solutions
  - easier to replace factors, easier to add new factors
  - flexibility in the adoption of multiple communication channels
- simpler
  - simpler security analysis
  - simpler development and maintenance of MFA standards

Eventual drawbacks

- in general less efficient than customized solutions but
  - optimisation potential (comms, comps)
  - may gain efficiency by deployment (e.g. single vs. multiple channels)

Aiming at Generalised MFA

(α,β,γ)-MFAKE framework

- support for arbitrary combinations of authentication factors
- initially three factor types: passwords, private/public keys, biometrics
- each type may occur multiple times or not occur at all
  - α passwords $\mathbf{pw}_C = [pw_{C,1}, ..., pw_{C,\alpha}]$ with $\mathbf{pw}_C[i] \in \mathcal{D}_{pw}$
  - β private keys $\mathbf{sk}_C = [sk_{C,1}, ..., sk_{C,\beta}]$ (and corresp. $\mathbf{pk}_C$)
  - γ biometrics $\mathbf{W}_C = [W_{C,1}, ..., W_{C,\gamma}]$ with $\mathbf{W}_C[i] \in Dist_{C,i}$
- independent of channels/devices being used

Examples

(2,0,0)-MFAKE

pw
one-time pw

(1,1,0)-MFAKE

(one-time) pw
sk

(1,0,1)-MFAKE

pw
fingerprint

(1,1,1)-MFAKE

pw
sk
fingerprint

Generalised MFA through Modularity

UNIVERSITY OF SURREY

Build on existing single-factor protocols

- view existing single-factor solutions as instances of (α,β,γ)-MFA
  - (1,0,0)-MFA = (one-time) password authentication
  - (0,1,0)-MFA = public-key authentication
  - (0,1,0)-MFA = biometric authentication

- construct **General MFA** as a product of single-factor authentication protocols

$$(\alpha,\beta,\gamma)\text{-MFA} = \alpha \times (1,0,0)\text{-MFA} + \beta \times (0,1,0)\text{-MFA} + \gamma \times (0,0,1)\text{-MFA}$$

with some extra work

- this would allow for backward compatibility and simple addition of new factors
  - simply by increasing/decreasing α, β, γ
  - adding new factor types, e.g. (0,0,0,1)-MFA for social authentication

First Attempt: Template for (α,β,γ)-MFA

Use single-factor protocols offering *at least* unilateral authentication.

possibly encrypted hashed

W_C sk_C pw_C

W_C pk_C pw_C

$pw_C[i] \rightarrow$ α password authentication $\leftarrow pw_C[i]$, $\rightarrow$ accept

$sk_C[i] \rightarrow$ β public key authentication $\leftarrow pk_C[i]$, $\rightarrow$ accept

$W_C[i] \rightarrow$ γ biometric authentication $\leftarrow W_C[i]$, $\rightarrow$ accept

This construction is not secure!

... but imagine its benefits

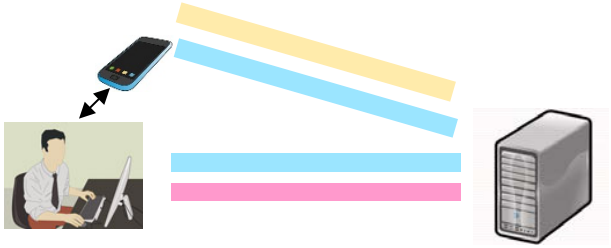
If all factors accepted then accept C

Massive Flexibility through Isolation

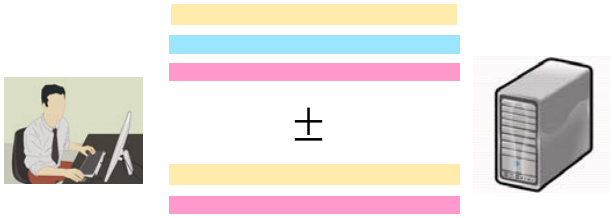
UNIVERSITY OF SURREY

arbitrary order

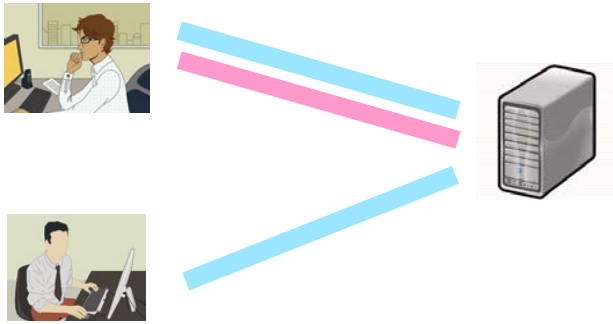
multiple channels



addition/removal of factors



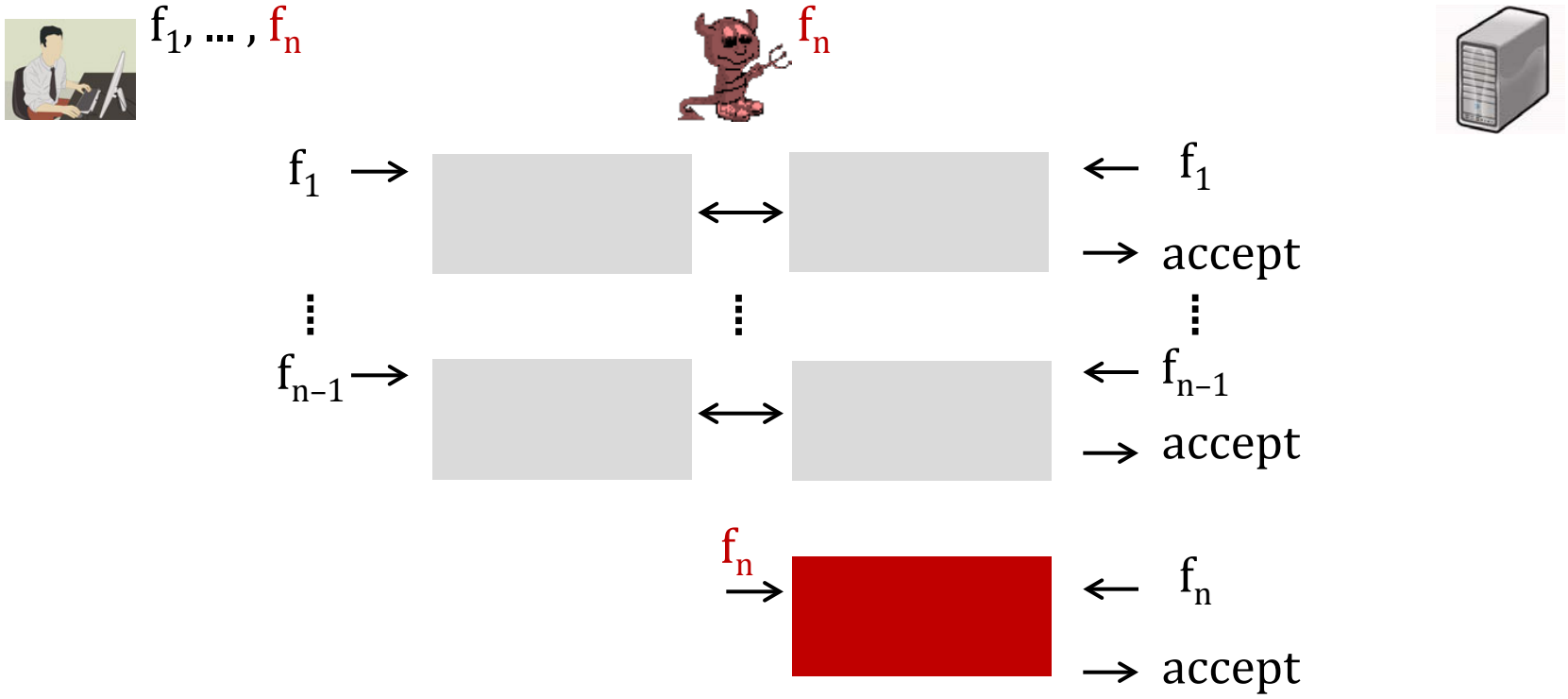
backward compatibility



Impersonation Attack on Isolated Factors

MFA security model must allow active attacks for up to n – 1 compromised factors.

An active adversary has typically full control over the network.



Server should not be allowed to accept based on completely isolated factors.

Designing secure (α,β,γ)-MFA: Step 1

If S accepts there must exist an instance of C with the „same view“ as S.

Same view is typically defined through matching protocol transcripts.

α

$pw_C[i]$ → ← $pw_C[i]$

password authentication

tran[i] ← → accept, tran[i]

β

$sk_C[i]$ → ← $pk_C[i]$

public key authentication

tran[i] ← → accept, tran[i]

γ

$W_C[i]$ → ← $W_C[i]$

biometric authentication

tran[i] ← → accept, tran[i]

$tran_C$ → ← $tran_S$

<u>authenticated</u> matching of $tran_C = tran_S$

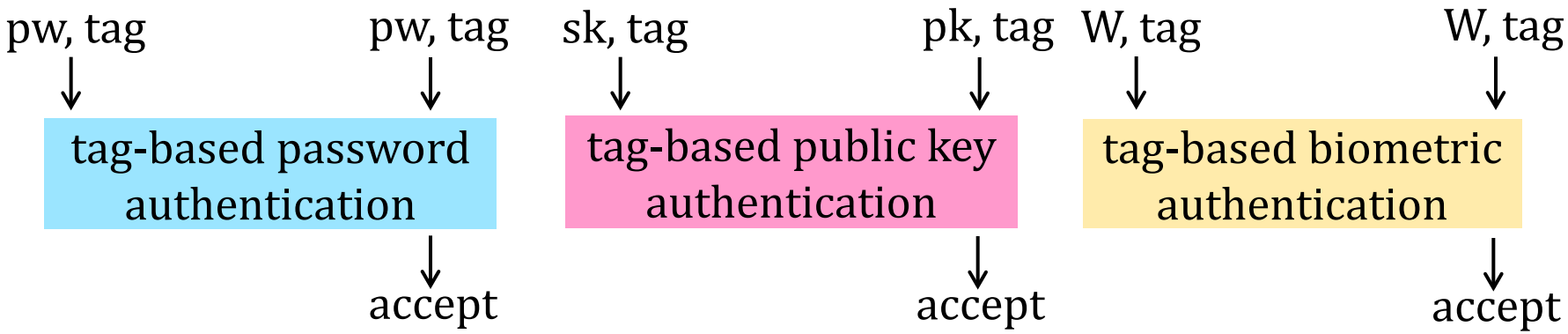
→ If all factors accepted and $tran_C = tran_S$ then accept C

This needs to be realized without additional factors.

Designing secure (α,β,γ)-MFA: Step 2

Tag-based Authentication

- authentication protocols that also ensures matching of input tags



Unauthenticated Key Exchange

- (forward-)secure key exchange in presence of passive adversary
  - adversary sees (tran, key) pairs and should not be able to tell if key is real or random
- e.g. standard Diffie-Hellman KE protocol with ephemeral private keys

Final (α,β,γ)-MFA Framework

UKE

tran, key ← → tran, key

tag_C = H1(C, S, tran, key)

tag_S = H1(C, S, tran, key)

α

tag_C, $\mathbf{pw}_C[i]$ → ← tag_S, $\mathbf{pw}_C[i]$

tag-based password authentication

tran[i] ← → accept, tran[i]

β

tag_C, $\mathbf{sk}_C[i]$ → ← tag_S, $\mathbf{pk}_C[i]$

tag-based public key authentication

tran[i] ← → accept, tran[i]

γ

tag_C, $\mathbf{W}_C[i]$ → ← tag_S, $\mathbf{W}_C[i]$

tag-based biometric authentication

tran[i] ← → accept, tran[i]

μ_C = H2(C, S, $tran_C$, key) —μ_C→ μ_S = H2(C, S, $tran_S$, key)

If all factors accepted and $\mu_C = \mu_S$ then accept C

Final (α,β,γ)-MFA Framework with KE

tran, key ← UKE → tran, key

$tag_C = H1(C, S, tran, key)$

$tag_S = H1(C, S, tran, key)$

α

tag_C, $\mathbf{pw}_C[i]$ → tag-based password authentication ← tag_S, $\mathbf{pw}_C[i]$

tran[i] ← tag-based password authentication → accept, tran[i]

β

tag_C, $\mathbf{sk}_C[i]$ → tag-based public key authentication ← tag_S, $\mathbf{pk}_C[i]$

tran[i] ← tag-based public key authentication → accept, tran[i]

γ

tag_C, $\mathbf{W}_C[i]$ → tag-based biometric authentication ← tag_S, $\mathbf{W}_C[i]$

tran[i] ← tag-based biometric authentication → accept, tran[i]

$\mu_C = H2(C, S, tran_C, key)$

$K = H3(C, S, tran_C, key)$

μ_C ⟶

$\mu_S = H2(C, S, tran_S, key)$

If all factors accepted and $\mu_C = \mu_S$ then compute

$K = H3(C, S, tran_C, key)$

(α,β,γ)-MFA(KE) with Server Authentication

pk_S

sk_S

tran, key ← UKE → tran, key

$tag_C = H1(C, S, tran, key)$

$tag_S = H1(C, S, tran, key)$

tag-based password authentication

tag-based public key authentication

tag-based biometric authentication

tag_C, pk_S →

← accept, tran

tag-based public key authentication

← tag_S, sk_S

→ tran

$\mu_C = H2(C, S, tran_C, key)$

$\nu_C = H2(S, C, tran_S, key)$

If accept and $\nu_S = \nu_C$ then compute

$K = H3(C, S, tran_C, key)$

μ_C →

← ν_S

$\mu_S = H2(C, S, tran_S, key)$

$\nu_S = H2(S, C, tran_S, key)$

If all factors accepted and $\mu_C = \mu_S$ then compute

$K = H3(C, S, tran_C, key)$

Client MFA(KE) Security Goals

Client Multi-Factor Authentication

- A is given access to Invoke, Send, BioComp, RevealSK, and CorruptC oracles
- A wins if S accepts without an existing instance of C with a matching transcript

$$\left| \Pr[\text{S accepts}] - q \times \left(\frac{\alpha}{|\mathcal{D}_{pw}|} + \sum_{i=1}^{\gamma} \text{false}^{pos}{}_i \right) \right| \leq \text{negl}$$

AKE security

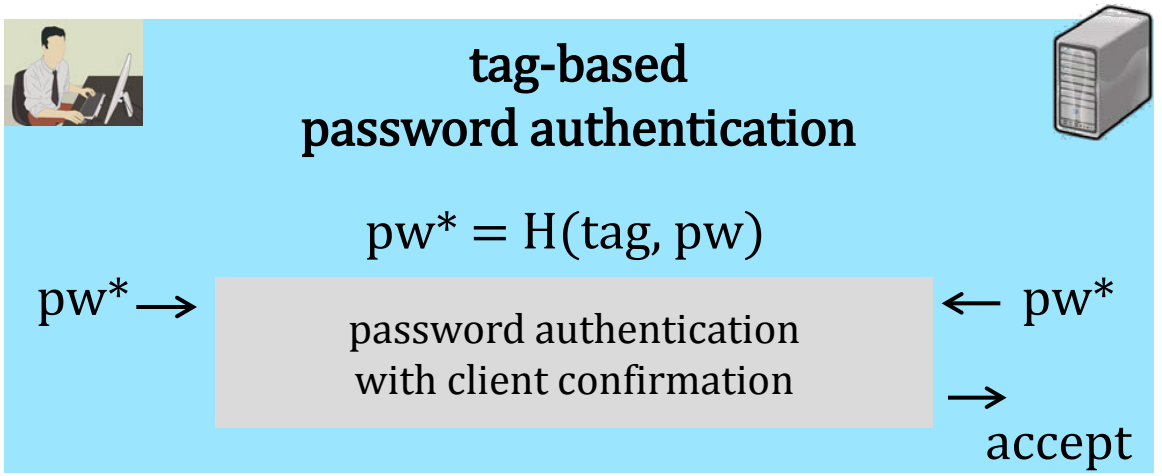
- A has further access to Test_b oracle in „real-or-random“ model [AFP05]
- A wins if it outputs $b^* = b$ such that

$$\left| \Pr[b^* = b] - q \times \left(\frac{\alpha}{|\mathcal{D}_{pw}|} + \sum_{i=1}^{\gamma} \text{false}^{pos}{}_i \right) - \frac{1}{2} \right| \leq \text{negl}$$

Tag-based Single-Factor Authentication

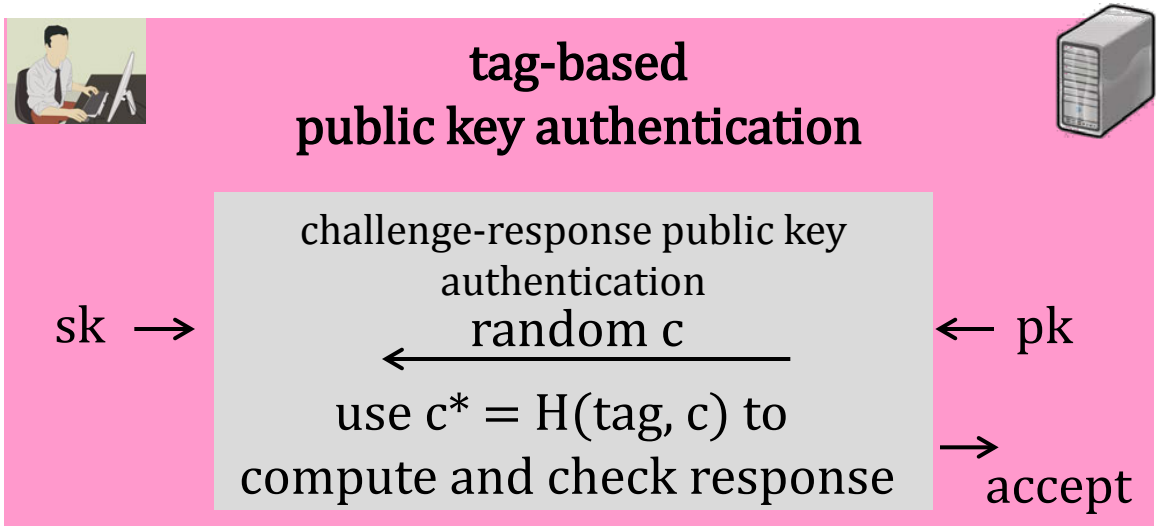
with passwords

- PAKE protocols in the model from [BPR00] with client confirm
- applicable to Ω-method from [GMR06]
- non-invasive



with public keys

- introduced in [JKSS10]
- challenge-response protocols: signature-based and 3-move ZK
- invasive



Tag-based Single-Factor Authentication

with biometric templates

- e.g. using robust fuzzy extractors (Ext, Rec) from [BDK+05]
- $(K, pub) \leftarrow Ext(W_C)$ stored on server side after registration
- $K' = K$ if $\delta(W'_C, W_C) \leq \tau$ (e.g. matching based on Hamming distance)
- invasive on client side

tag-based biometric authentication

K, pub

pub ⟵

measure W'_C

$K' \leftarrow Rec(W'_C, pub)$

$K' = H(C, S, K', tag, pub)$ $K = H(C, S, K, tag, pub)$

$K' \rightarrow$ challenge-response private key authentication $\leftarrow K$

$\rightarrow$ accept

Summary and Relations

UNIVERSITY OF SURREY

General MFA(KE) framework

- single UKE session plus
- (parallelizable) single-factor authentication sessions
  - α tag-based password authentication sessions
  - β tag-based public key authentication sessions
  - γ tag-based biometric authentication sessions
- optional server authentication without security risks (in contrast to [PZ08])
- has a lot of flexibiity and optimization potential

Proven relations

- symmetric PAKE model in [BPR00] is equivalent to (1,0,0)-MFAKE
- public-key AKE model in [BJM97] is equivalent to (0,1,0)-MFAKE
- (α,0,0)-MFAKE implies multi-password setting in [SUC10]

Thank you!