Direct Anonymous Attestation (DAA)

Liqun Chen
Trusted Systems Laboratory
Hewlett Packard Laboratories, Bristol

12 October 2005

The slides presented here were made for a DAA seminar last year

outline

- what is DAA?
- what is DAA for?
- why DAA?
- how does DAA work?

outline

- what is DAA?
- what is DAA for?
- why DAA?
- how does DAA work?

DAA is a signature scheme

- DAA is a signature scheme designed for TCG
  - signer: TPM (trusted platform module)
  - verifier: an external partner
- the name of DAA is from
  - Direct proof – without a TTP involvement
  - Anonymous – do not disclose the identity of the signer
  - Attestation – statement/claim from a TPM
- DAA was adopted by TCG and specified in TCG TPM Specification Version 1.2, available at www.trustcomputinggroup.org
- designers: Ernie Brickell of Intel, Jan Camenisch of IBM and Liqun Chen of HP

category of signature schemes – from a verifier's point of view

- 1–out–1 signatures: *ordinary signatures*
  - a verifier is given an authenticated public key of a signer
- 1–out–n signatures: *ring signatures, designated-verifier signatures, concurrent signatures, ……*
  - a verifier is given authenticated public keys of all potential signers
- 1–out–group signatures: *group signatures, DAA*
  - a verifier is given an authenticated group public key

group signatures and DAA

- a group signature has fixed-traceability and unlinkability
  - a group member certificate indicates an identity-disclosure authority
  - the authority can recover the identity of the real signer from a group signature
- a DAA signature has flexible-traceability and flexible-linkability
  - there is no identity-disclosure authority (a DAA signature cannot be opened by any TTP)
  - a DAA signature provides the user-control link that can be used to link some selected signatures from the same signer for the same verifier

outline

- what is DAA?
- what is DAA for? – for TCG
- why DAA?
- how does DAA work?

goals of the TCG architecture

protect user's information

ensure user's choice on use of security mechanism

protect user's computing environment

protect user's privacy

obstacle to achieving the goals of the TCG architecture

security might be fundamentally incompatible with privacy

obstacle to achieving the goals of the TCG architecture

security might be fundamentally incompatible with privacy

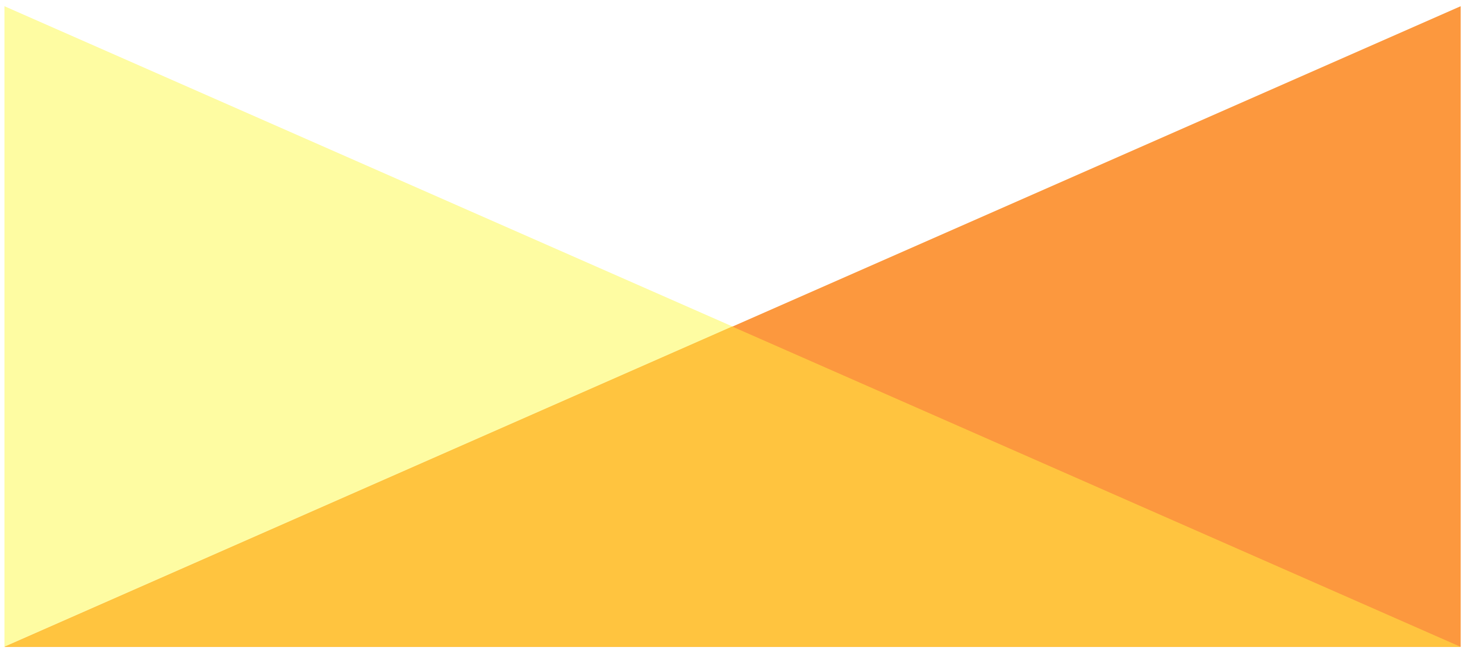
high security
&
low privacy

obstacle to achieving
the goals of the TCG architecture

security might be fundamentally incompatible with privacy

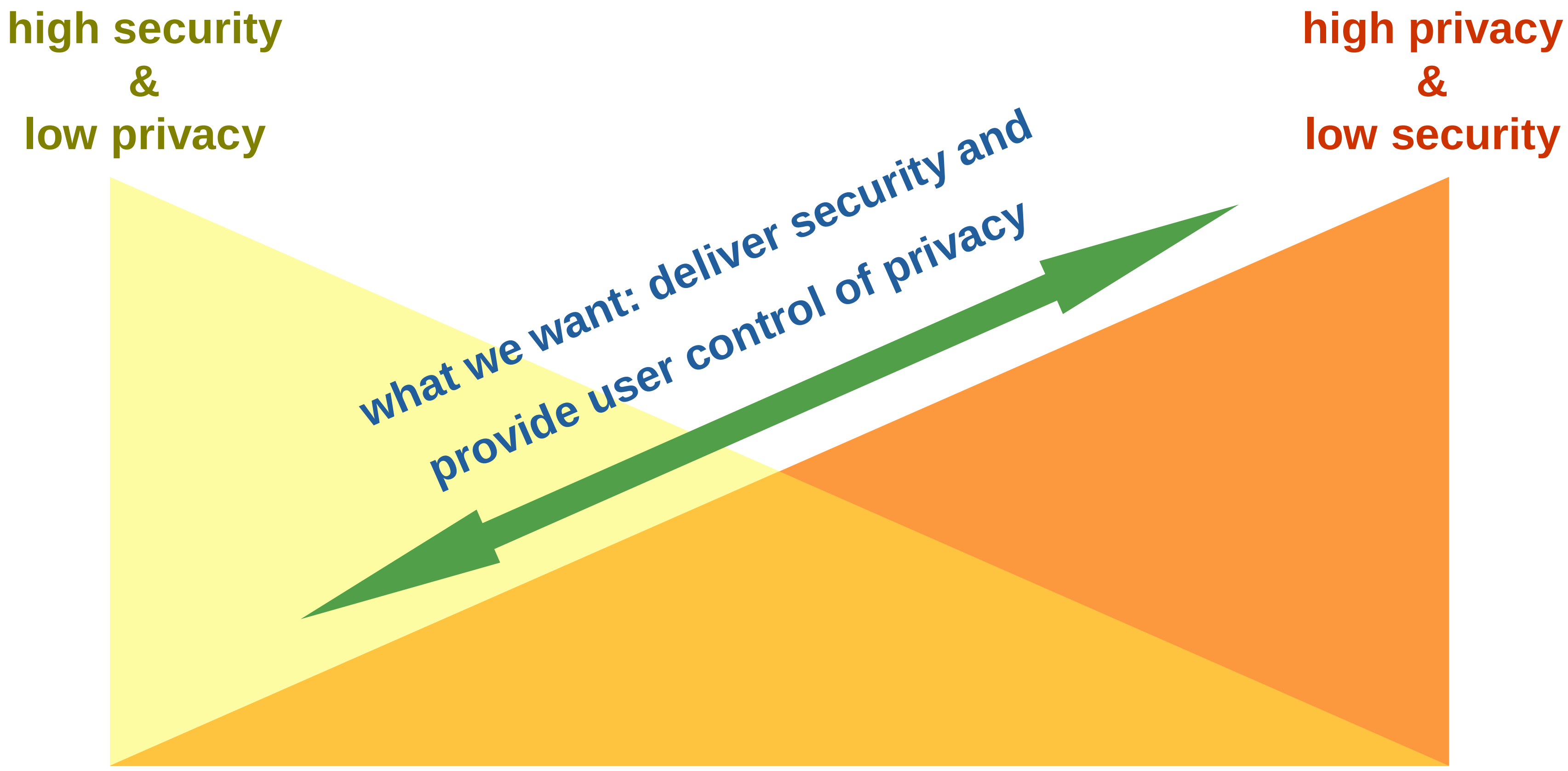
high security
&
low privacy

high privacy
&
low security



obstacle to achieving the goals of the TCG architecture

security might be fundamentally incompatible with privacy



TPM (trusted platform module)

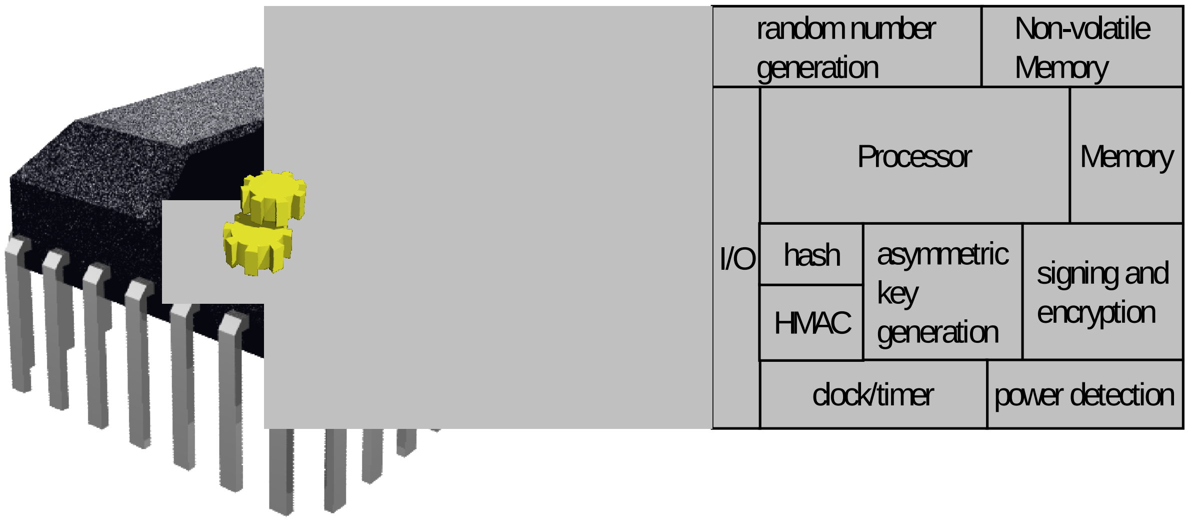
the TPM is the root of trust for reporting -

- it offers smartcard-like security capability embedded into the platform
- it is trusted to operate as expected (conforms to the TCG spec)
- it is uniquely bound to a single platform
- its functions and storage are isolated from all other components of the platform (e.g., the CPU)

TPM (trusted platform module)

the TPM is the root of trust for reporting -

- it offers smartcard-like security capability embedded into the platform
- it is trusted to operate as expected (conforms to the TCG spec)
- it is uniquely bound to a single platform
- its functions and storage are isolated from all other components of the platform (e.g., the CPU)

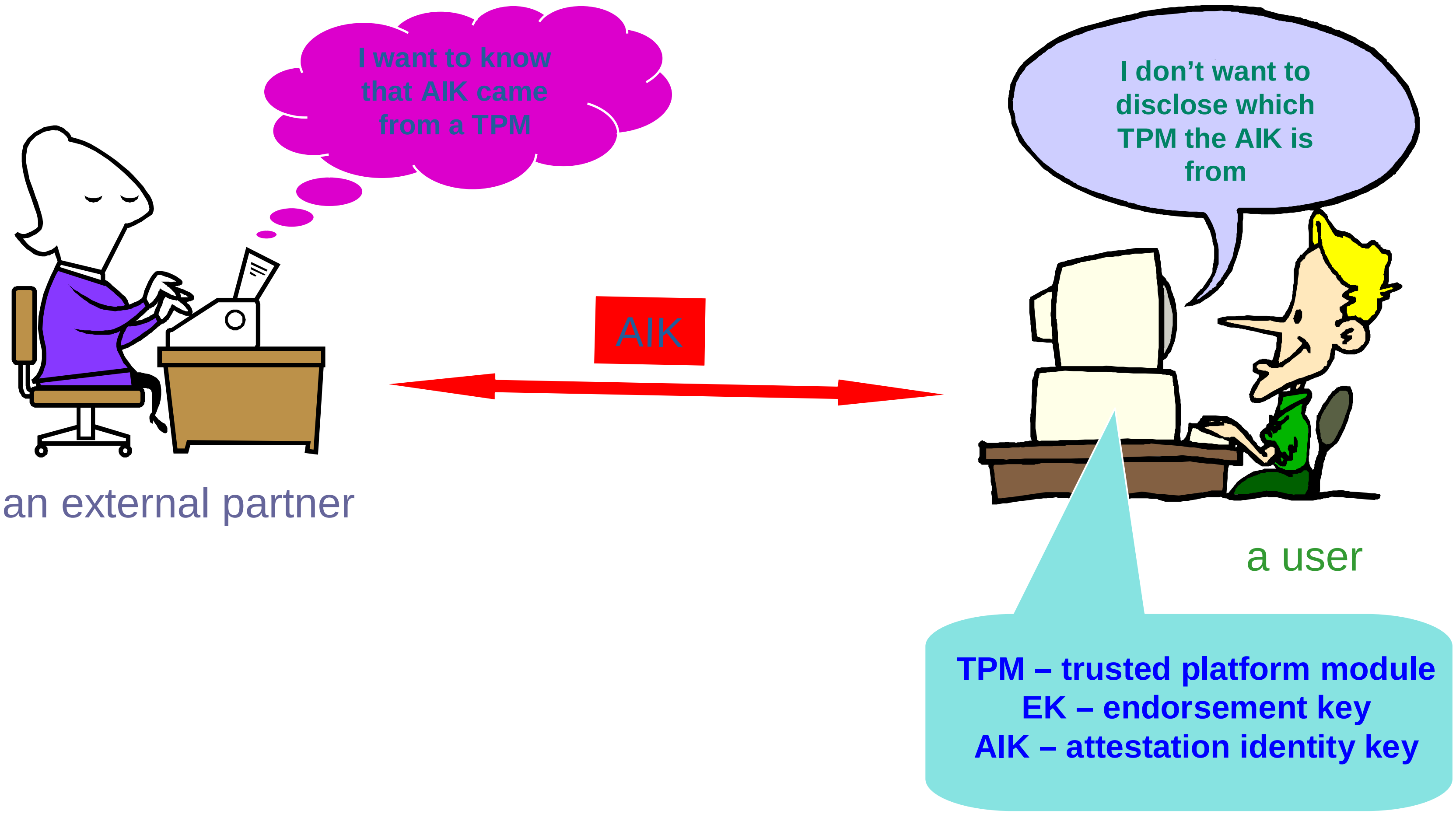


platform attestation

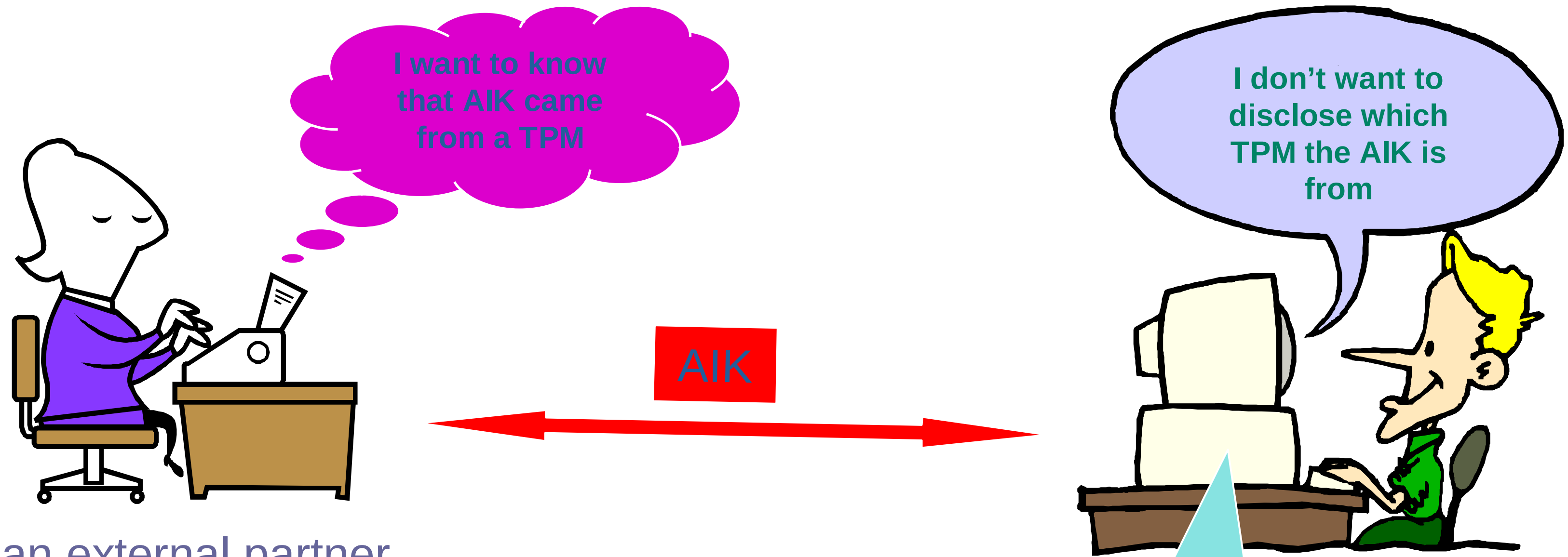
- TCG requires a TPM to have an embedded "endorsement key (EK)", to prove that a TPM is a particular genuine TPM
- EK is not a platform identity
- TCG lets a TPM control "multiple pseudonymous attestation identities" by using "attestation identity key (AIK)"
- AIK is a platform identity, to attest to platform properties

we need a link between EK and AIK

privacy issue



privacy issue



an external partner

a user

we seek a solution to convince an external party that an AIK is held in a TPM without identifying the TPM

TPM – trusted platform module
EK – endorsement key
AIK – attestation identity key

outline

- what is DAA?
- what is DAA for?
- why DAA?
- how does DAA work?

previous solution is not good enough

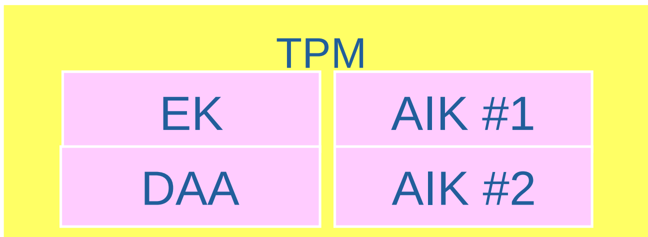
the previous solution (before TCG TPM spec. v1.2) -

- involves a TTP to issue certificates
- allows choice of any (different) certification authorities (privacy-CA) to certify each TPM identity
- can help prevent correlation, however
  anonymity is dependent upon the private-CA

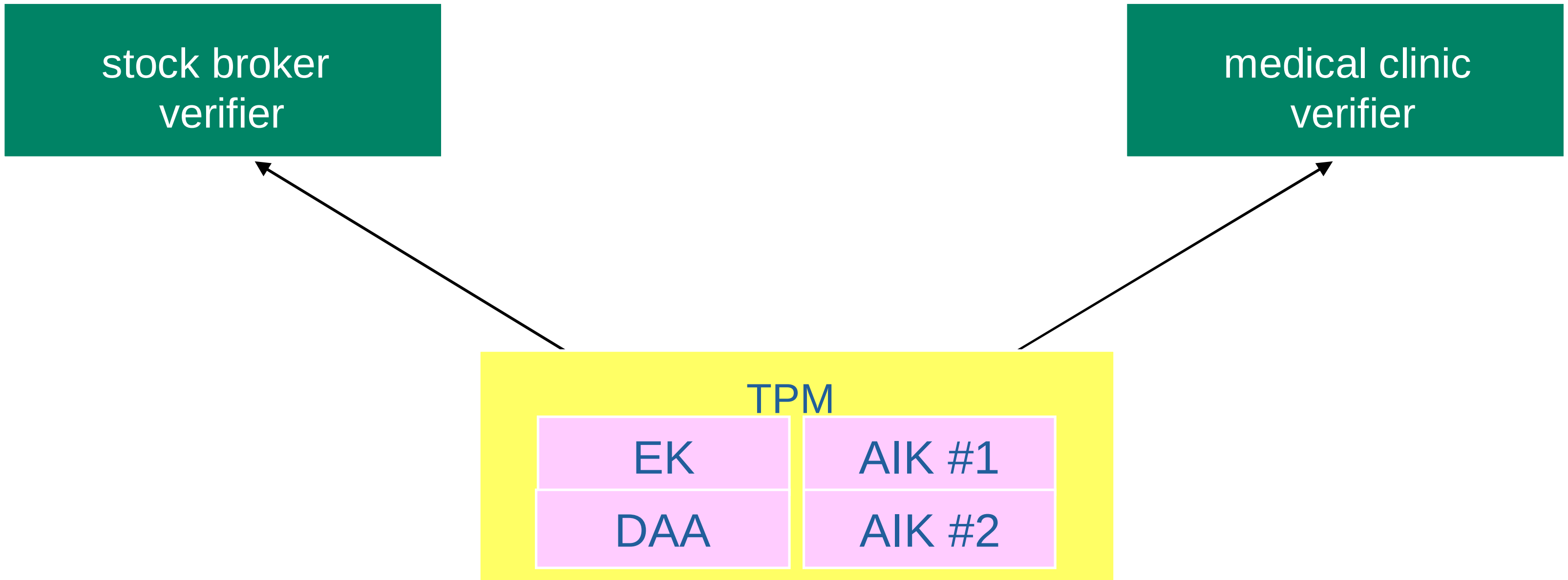
our goal and solution

- our goal: a solution provides
  – anonymity without a TTP
  – authentication without a certificate
- our solution:
  – **direct anonymous attestation (DAA)**
  direct proof replaces the TTP

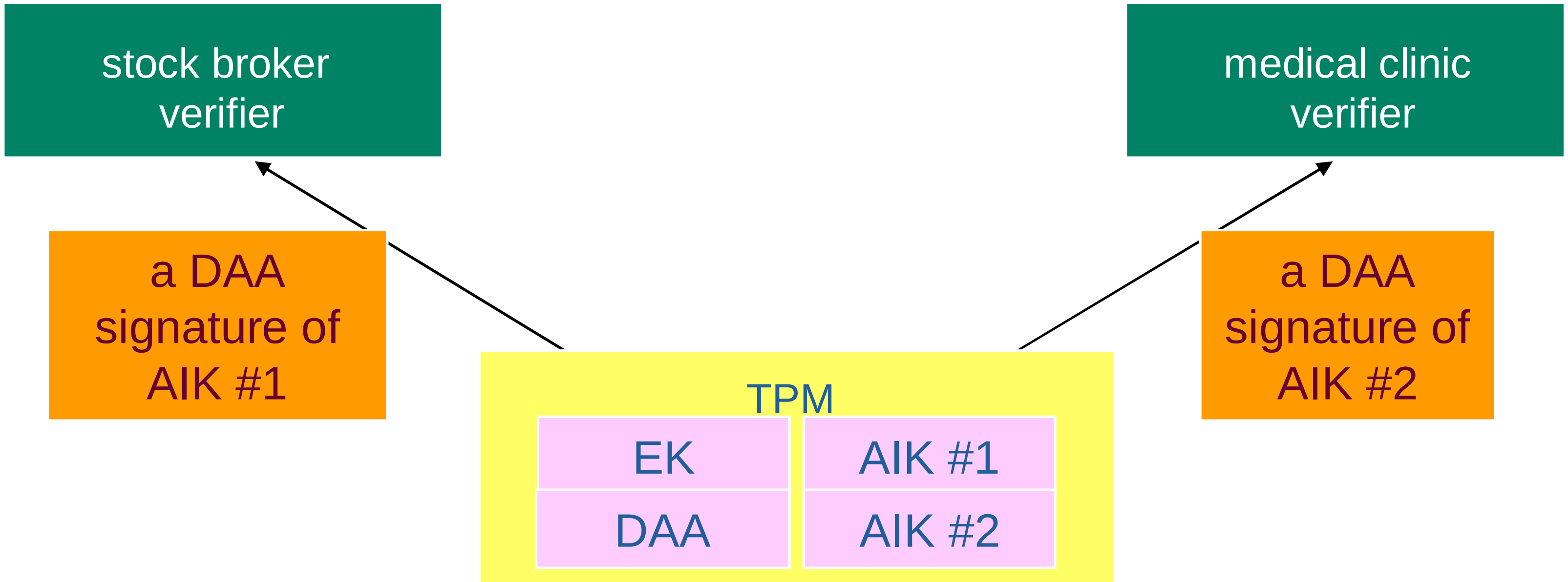
a simple picture of DAA



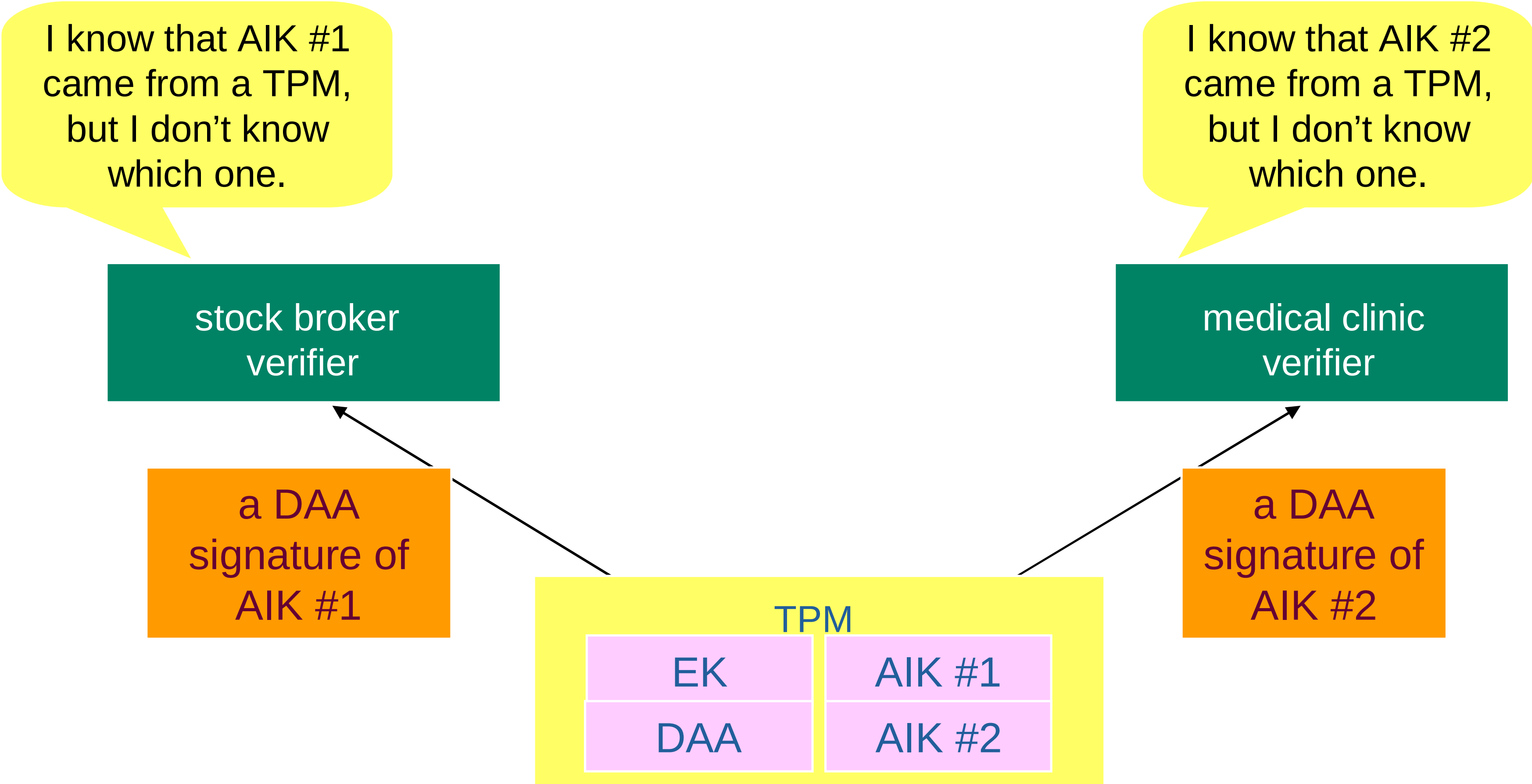
a simple picture of DAA



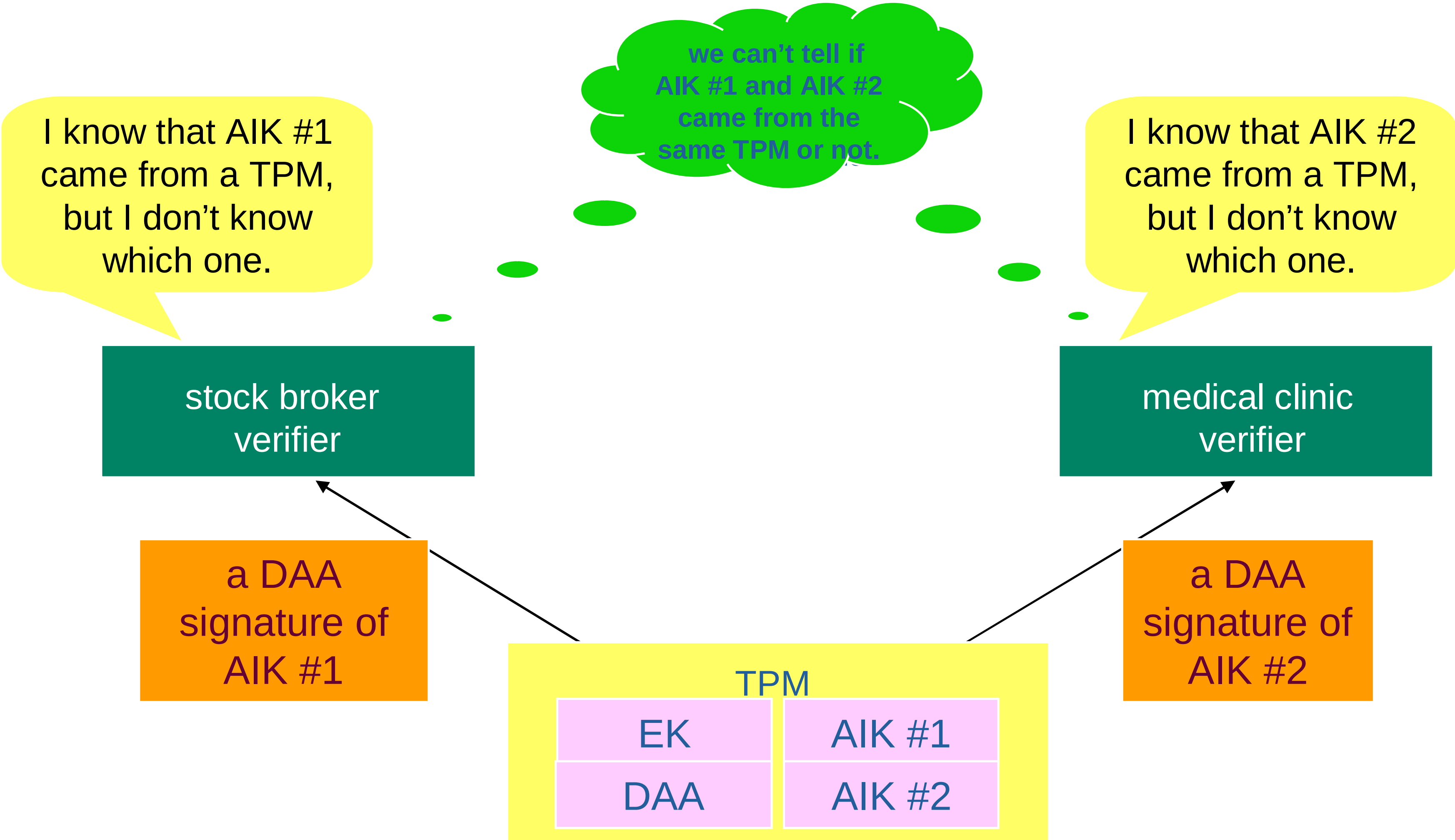
a simple picture of DAA



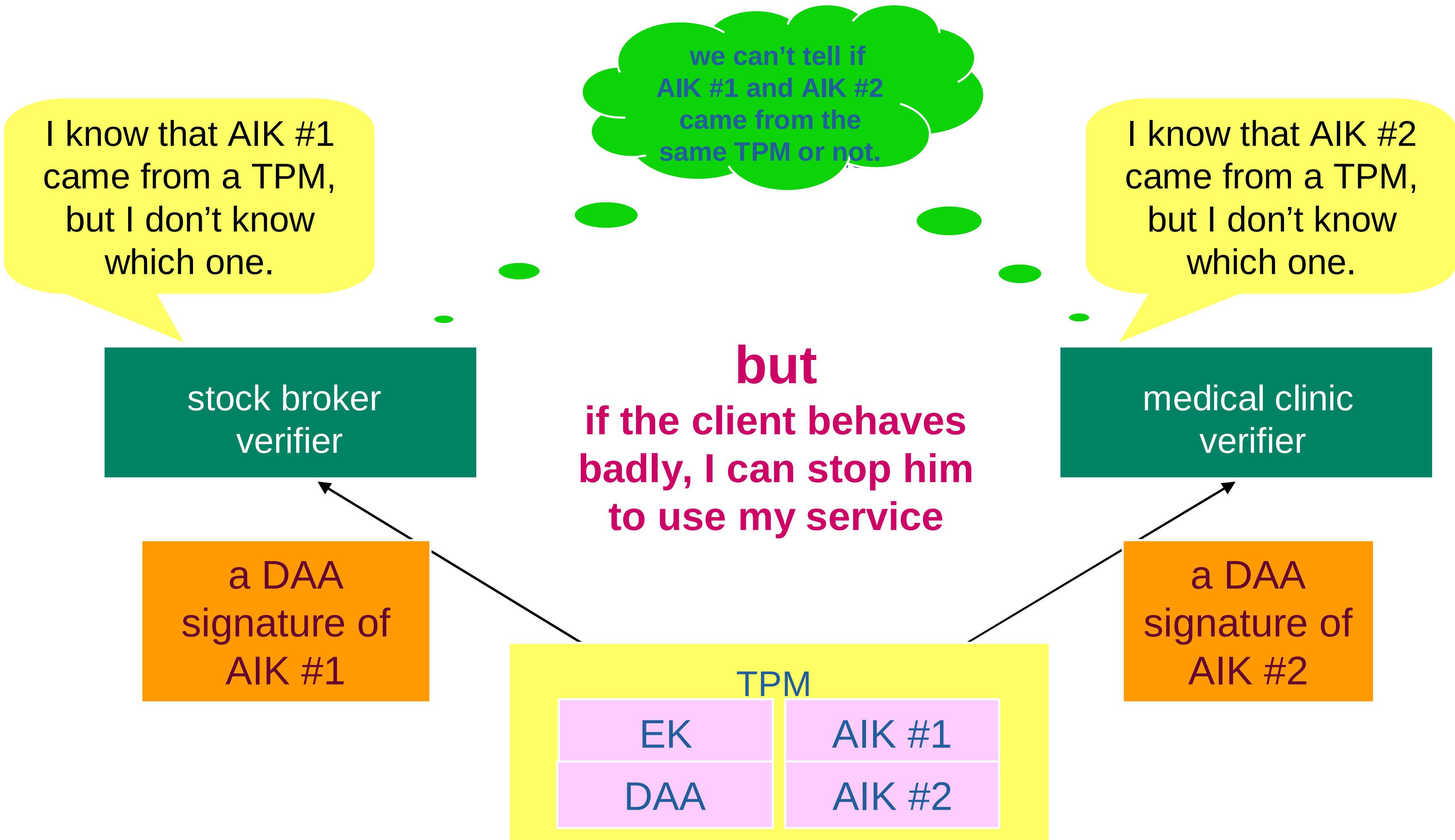
a simple picture of DAA



a simple picture of DAA



a simple picture of DAA



outline

- what is DAA?
- what is DAA for?
- why DAA?
- how does DAA work?

the DAA scheme outline

- entities
  - DAA issuer: a DAA certificate issuer (e.g., a manufacturer of TCG platforms)
  - DAA signer: a trusted platform module (TPM) with help from a host platform
  - DAA verifier: an external partner (e.g.,a service provider)
- primitives
  - system and issuer setup
  - join protocol
  - signing algorithm
  - verifying algorithm
  - solution of restricted link
  - solution of revocation

setup

- Issuer public key: $PK_I = (hk, n, g', g, h, S, Z, R_0, R_1, g, \Gamma, r)$
  - RSA parameters with

    n – an RSA modulus

    $g' \in QR_n$

    $g, h \in \langle g' \rangle$

    $S, Z \in \langle h \rangle$

    $R_0, R_1 \in \langle S \rangle$
  - a group of prime order with

    Γ - modulus (prime)

    r - order (prime, s.t. $r | \Gamma - 1$)

    g - generator ($g^r = 1 \bmod \Gamma$)
  - a hash function

    H_{hk} - a hash function of length hk
- private key: factorisation of n

a non-interactive proof of correctness of key generation (using the Fiat-Shamir heuristic)

join

entities: TPM, Host and Issuer

- DAA signing key (created by TPM):
  - f_0, f_1 (104-bit)
- DAA certificate (created with Issuer):
  - v (2536-bit)
  - A (2048-bit)
  - e (prime $\in_R [2^{367}, 2^{367} + 2^{119}]$)

$$R_0^{f_0} R_1^{f_1} S^v A^e = Z \pmod{n}$$

values R_0, R_1, S, Z, n are part of PK_I

- TPM stores f_0, f_1, v, $H(A||e||PK_I)$
- Host stores A and e

join

entities: TPM, Host and Issuer

- DAA signing key (created by TPM):
  - f_0, f_1 (104-bit)
- DAA certificate (created with Issuer):
  - v (2536-bit)
  - A (2048-bit)
  - e (prime $\in_R [2^{367}, 2^{367} + 2^{119}]$)

$$R_0^{f_0} R_1^{f_1} S^v A^e = Z (\bmod n)$$

values R_0, R_1, S, Z, n are part of PK_I

- TPM stores $f_0, f_1, v, H(A||e||PK_I)$
- Host stores A and e

an authentic channel between TPM and Issuer using the endorsement key (EK) of TPM

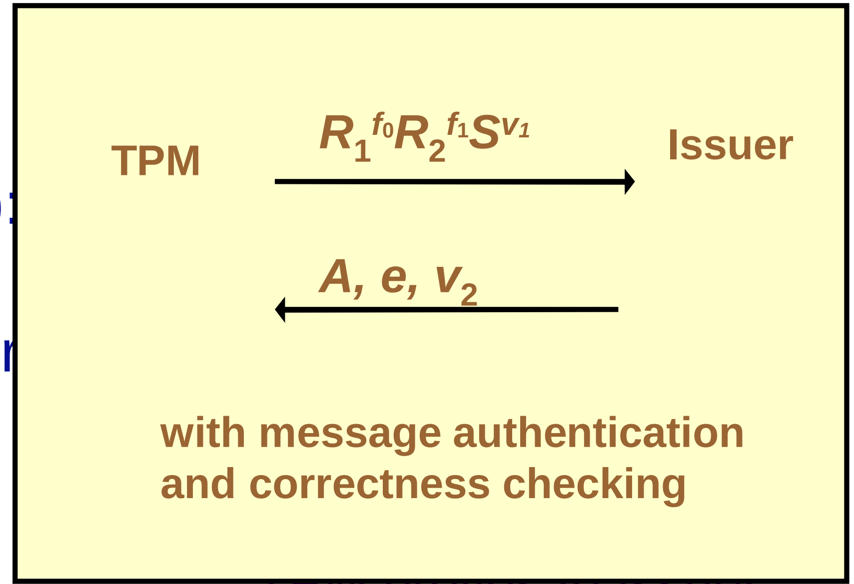
v is contributed by both TPM and Issuer

TPM proves to Issuer knowledge of f_0, f_1 and its contribution on v

Issuer proves to Host correctness of certificate generation

join

entities: TPM, Host and Issuer



- DAA signing key (created by TPM):
  - f_0, f_1 (104-bit)
- DAA certificate (created with Issuer
  - v (2536-bit)
  - A (2048-bit)
  - e (prime $\in_R [2^{367}, 2^{367} + 2^{119}]$)

$$R_0^{f_0} R_1^{f_1} S^v A^e = Z \pmod{n}$$

values R_0, R_1, S, Z, n are part of PK_I

- TPM stores f_0, f_1, v, $H(A||e||PK_I)$
- Host stores A and e

TPM proves to Issuer knowledge of f_0, f_1 and its contribution on v

Issuer proves to Host correctness of certificate generation

join

the Camenisch-Lysyanskaya signature scheme and based on the strong RSA problem
given n and z
find a and e
s.t. $a^e = z \pmod{n}$

entities: TPM, H

- DAA signing ke
  - f_0, f_1 (104-bit)
- DAA certificate
  - v (2536-bit)
  - A (2048-bit)
  - e (prime $\in_R [2^{367}, 2$

$$R_0^{f_0} R_1^{f_1} S^v A^e = Z \pmod{n}$$

values R_0, R_1, S, Z, n are part of PK_I

- TPM stores f_0, f_1, v, $H(A||e||PK_I)$
- Host stores A and e

S^{v_1} Issuer

uthentication
s checking

TPM proves to Issuer knowledge of f_0, f_1 and its contribution on v

Issuer proves to Host correctness of certificate generation

sign

Schnorr signature

private/public key

$(x, y = g^x)$

signature

msg - message

$r \in_R \{0,1\}^l$

$t = g^r$

$c = H(t||msg)$

$s = r + xc$

$s = (c, s)$

verification

$c \equiv H(g^s y^{-c}||msg)$

DAA signature

private key : f_0, f_1

certificate : v, A, e, satisfying $R^{f_0} R^{f_1} S^v A^e = Z \pmod n$

public key : $PK_I = (hk, n, g', g, h, R_0, R_1, S, Z, g, \Gamma, r)$

commitment

$w, r \in_R \{0,1\}^l$ z – the base name

$T_1 = Ah^w \pmod n$ $T_2 = g^w h^e (g')^r \pmod n$

$N_v = z^{f_0 + f_1 2^{104}} \pmod \Gamma$

signature

msg, r, t, c, s, s

sign

a DAA signature is presented by

$$msg, r, t, c, s, \mathrm{s}$$

sign

$msg = b||m$
$b \in \{0,1\}$
$m \in$ {AIK, other string}
if $b = 0$,
m = AIK - RSA key
if $b = 1$
m = other string

DAA signature

msg, r, t, c, s, s

sign

$msg = b||m$
$b \in \{0,1\}$
$m \in$ {AIK, othe
string}
if $b = 0$,
m = AIK - RSA
if $b = 1$
m = other strin

re

$r = \{r_{v_1}, r_{v_2}, r_{f_0}, r_{f_1},$
$r_e, r_{ee}, r_w, r_r, r_{ew}, r_{er}\}$

$r_{v_1}, r_{v_2}, r_{f_0}, r_{f_1}$
are chosen by TPM

$r_e, r_{ee}, r_w, r_r, r_{ew}, r_{er}$
are chosen by Host

msg, r, t, c, s, s

sign

$msg = b||m$

$b \in \{0,1\}$

$m \in$ {AIK, othe

string}

if $b = 0$,

m = AIK - RSA

if $b = 1$

m = other strin

$r = \{r_{v_1}$

r_e, r_e

r_{v_1}, r_{v_2}, r

are ch

r_e, r_{ee}, r

are ch

$t = \{\tilde{T}_1, \tilde{T}_2, \tilde{T}'_2, \tilde{N}_v\}$

$\tilde{T}_1 = R_0^{r_{f0}} R_1^{r_{f1}} S^{r_{v1}} S^{r_{v2}} T_1^{r_e} h^{-r_{ew}} \pmod{n}$

$\tilde{T}_2 = g^{r_w} h^{r_e} g'^{r_r} \pmod{n}$,

$\tilde{T}'_2 = T_2^{-r_e} g^{r_{ew}} h^{r_{ee}} g'^{r_{er}} \pmod{n}$

$\tilde{N}_v = z^{r_{f0} + r_{f1} 2^{104}} \pmod{\Gamma}$

TPM computes $R_0^{r_{f0}} R_1^{r_{f1}} S^{r_{v1}} S^{r_{v2}}$

and $\tilde{N}_v$

Host computes others

msg, r, t, c, s, s

sign

msg = *b*||*m*
$b \in$ {0,1}
$m \in$ {AIK, othe
string}
if *b* = 0,
m = AIK - RSA
if *b* = 1
m = other strin

$r = \{r_{v_1}$
r_e, r_e
r_{v_1}, r_{v_2}, r
are ch
r_e, r_{ee}, r
are ch

d*n*)

c = {PK_I||z||
commitment||
t||n_v||n_t||*msg*}
where n_v and n_t
are nonce
chosen by
verifier & TPM
respectively

msg, *r*, *t*, *c*, *s*, s

sign

$msg = b||m$
$b \in \{0,1\}$
$m \in$ {AIK, othe
string}
if $b = 0$,
m = AIK - RSA
if $b = 1$
m = other strin

$r = \{r_{v_1}$
r_e, r_e
r_{v_1}, r_{v_2}, r
are ch
r_e, r_{ee}, r
are ch

c =
wh
are
ch
ve
res

$$s_{f_0} = r_{f_0} + cf_0$$
$$s_{f_1} = r_{f_1} + cf_1$$
$$s_v = r_v + cv$$
$$s_e = r_e + c(e - 2^{367})$$
$$s_{ee} = r_{ee} + ce^2$$
$$s_w = r_w + c_w$$
$$s_{ew} = r_{ew} + cew$$
$$s_r = r_r + cr$$
$$s_{er} = r_{er} + cer$$

msg, *r*, *t*, *c*, *s*, s

sign

$msg = b||m$

$b \in \{0,1\}$

$m \in$ {AIK, othe

string}

if $b = 0$,

m = AIK - RSA

if $b = 1$

m = other strin

$r = \{r_{v_1}$

r_e, r_e

r_{v_1}, r_{v_2}, r

are ch

r_e, r_{ee}, r

are ch

c =

wh

are

ch

ve

res

signature:

$$s = (z, \text{commitment}, c, n_t, s)$$

$$= (z, T_1, T_2, N_v, c, n_t,$$

$$s_v, s_{f_0}, s_{f_1}, s_e, s_{ee},$$

$$s_w, s_{ew}, s_r, s_{er})$$

msg, *r*, *t*, *c*, *s*, s

verify

input - message, signature and public key of Issuer

$$b \| m, \sigma = (\zeta, T_1, T_2, N_v, c, n_t, s_v, s_{f_0}, s_{f_1}, s_e, s_{ee}, s_w, s_{ew}, s_r, s_{er})$$

$$PK_I = (hk, n, g, g', h, R_0, R_1, S, Z, \gamma, \Gamma, \rho)$$

compute -

$$\hat{T}_1 = Z^{-c} T_1^{s_e + c2^{367}} R_0^{s_{f0}} R_1^{s_{f1}} S^{s_v} h^{-s_{ew}} \pmod{n}$$

$$\hat{T}_2 = T_2^{-c} g^{s_w} h^{s_e + c2^{367}} (g')^{s_r} \pmod{n}$$

$$\hat{T}'_2 = T_2^{-(s_e + c2^{367})} g^{s_{ew}} h^{s_{ee}} (g')^{s_{er}} \pmod{n}$$

$$\hat{N}_v = N_v^{-c} \zeta^{s_{f0} + s_{f1} 2^{104}} \pmod{\Gamma}$$

verify -

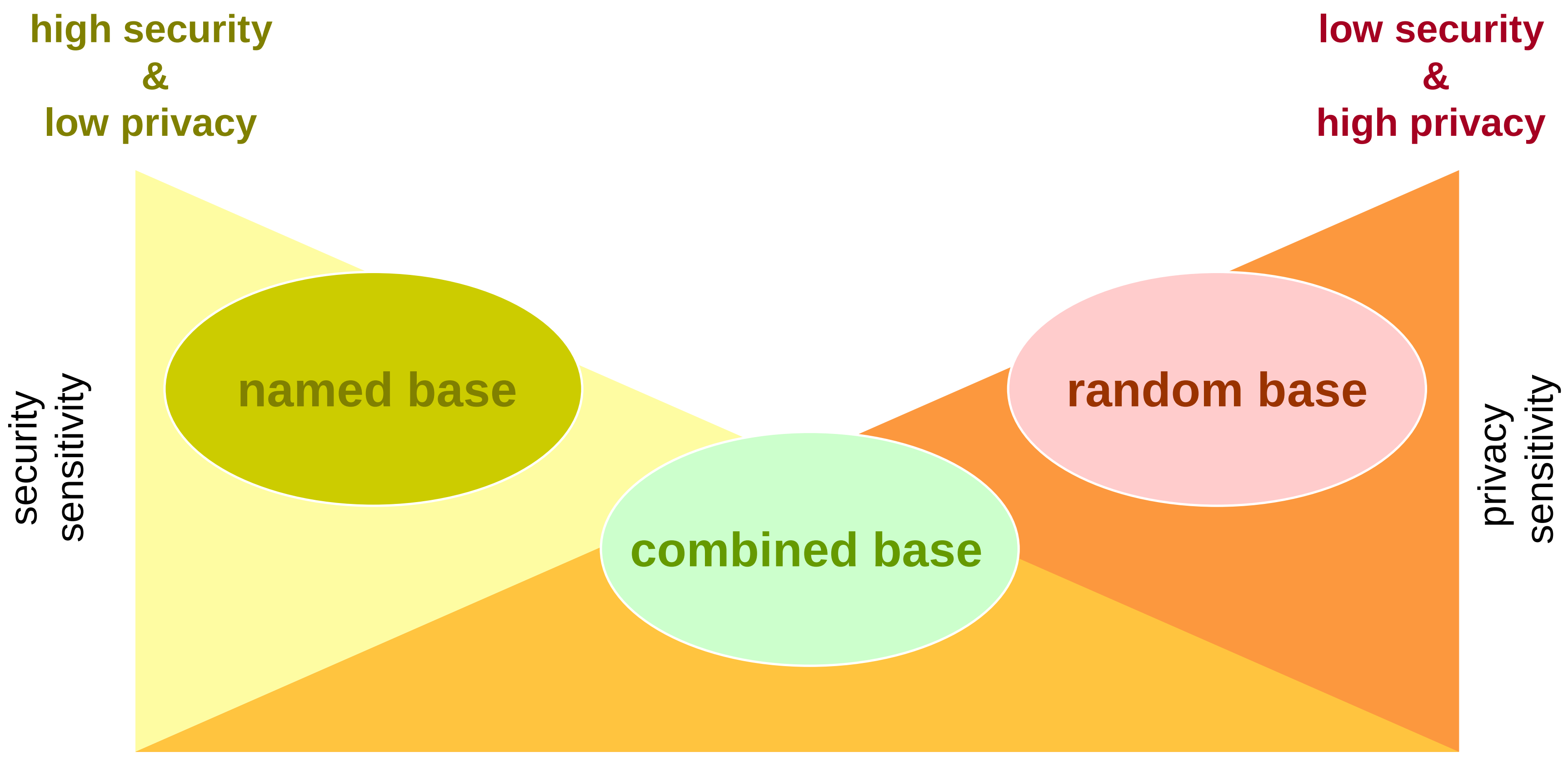
$$c \equiv H_{hk}(PK_I \| \zeta \| T_1 \| T_2 \| N_v \| \hat{T}_1 \| \hat{T}_2 \| \hat{T}'_2 \| \hat{N}_v \| n_t \| n_v \| b \| m)$$

$$N_v, \zeta \in_R \langle \gamma \rangle \quad \zeta = (H_\Gamma(1 \| bsn))^{(\Gamma - 1)/\rho} \pmod{\Gamma}$$

$$s_{f_0}, s_{f_1} \in \{0,1\}^{345} \quad s_e \in \{0,1\}^{361}$$

restricted link for a verifier – named/random base in a DAA signature

restricted link for a verifier – named/random base in a DAA signature



restricted link for a verifier – named/random base in a DAA signature

a base: $z \in_R \langle g \rangle$ or $z = (H(1||bsn))^{(\Gamma-1)/r} \pmod{\Gamma}$

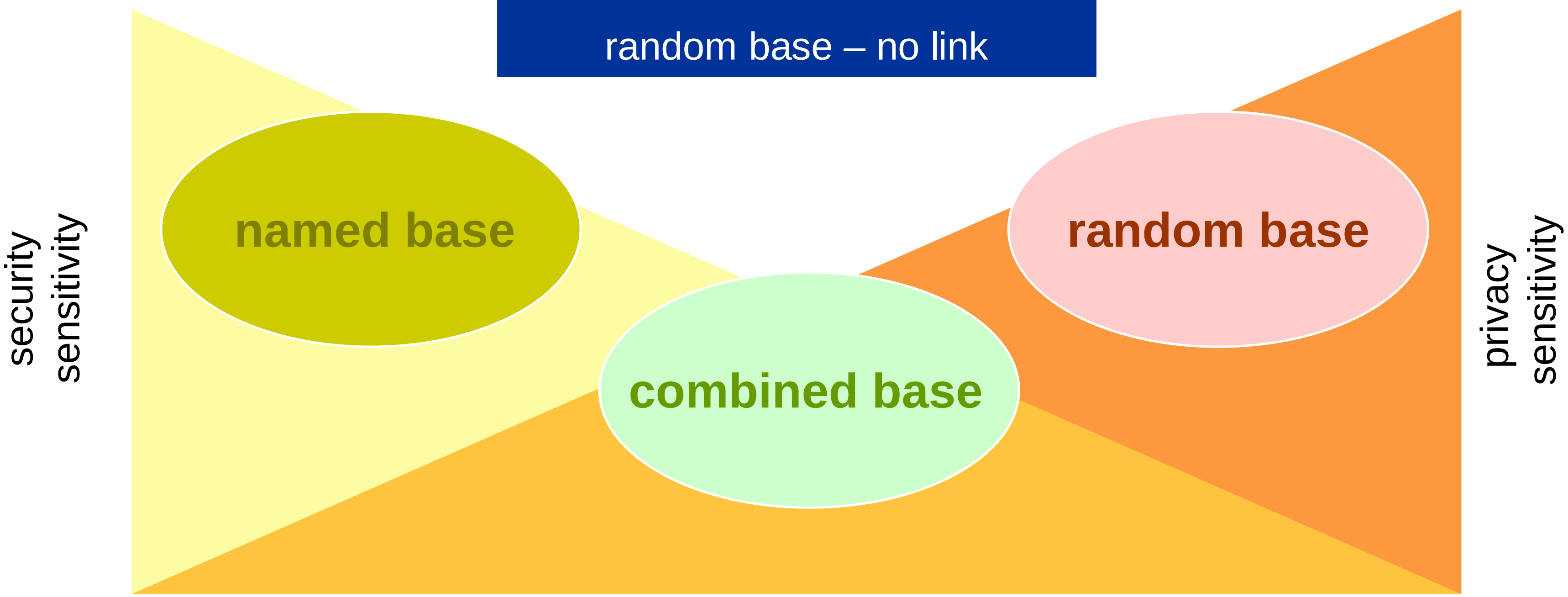
$$N_v = z^{f_0 + f_1 2^{104}} \pmod{\Gamma}$$

high security & low privacy

named base – verifier can link two signatures from the same TPM signed for the verifier

random base – no link

low security & high privacy



revoking a certificate

- if f_0 and f_1 are known
  - put f_0 and f_1 on a certificate revocation list and check the list in each verification process
- if f_0 and f_1 are not known
  - the name base solution can help a verifier to create his own certificate revocation list with

$$N_v = z^{f_0 + f_1 2^{104}} \pmod{\Gamma}$$

$$z = (H(1||bsn))^{(\Gamma-1)/r} \pmod{\Gamma}$$

security proof

- we prove the above DAA scheme is secure in the random oracle model under
  - the strong RSA assumption
  - the DDH assumption in QR_n and
  - the DDH assumption in $\langle g \rangle$
- By “the scheme is secure”, we mean
  - there exists no adversary that can adaptively run the join protocol, ask for signature by other (i.e., honest) members, and then output a signature containing a value N_v such that for all f_0 and f_1 extracted from the adversary in the join protocol N_v does not match

$$N_v = z^{f_0 + f_1 2^{104}} \pmod{\Gamma}$$

summary

DAA -

- is a signature scheme
- offers a zero knowledge proof of a key certificate
- provides a variety of balances between security and privacy by choosing
  - random base – for privacy sensitive cases
  - named base – for non privacy-sensitive cases
  - combinations
- has a security proof in the random oracle model based on:
  - the strong RSA assumption
  - the DDH assumption

future work

- more flexible privacy solutions
- more flexible revocation solutions

further information

- TCG initiatives:

  http://www.trustedcomputing.org

- E. Brickell, J. Camenisch and L. Chen. Direct anonymous attestation. In *Proc. 11th ACM Conference on Computer and Communications Security*, pages 132-145, ACM press, 2004

- B. Balacheff, L. Chen, S. Pearson, D. Plaquin and G. Proudler, Trusted Computing Platforms: TCPA technology in context, Prentice Hall PTR, 2003 invent