A Decade of Direct Anonymous Attestation

From Research to Standard and Back

Jan Camenisch

IBM Research – Zurich

Joint work with Ernie Brickell, Liqun Chen, Manu Drivers, Anja Lehmann

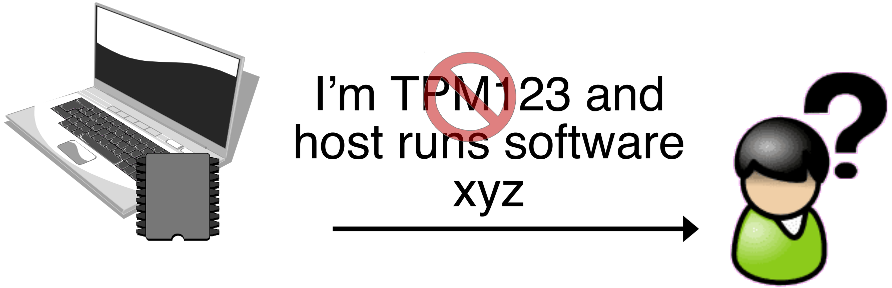
jca@zurich.ibm.com @JanCamenisch ibm.biz/JanCamenisch

Direct Anonymous Attestation – What is it?

Protocol standardized by TCG (trusted computing group)

- Attestation of computer state by TPM (root of trust)
- TPM measures boot sequence
- TPM attest boot sequence to third party
- Attestation based on cryptographic keys

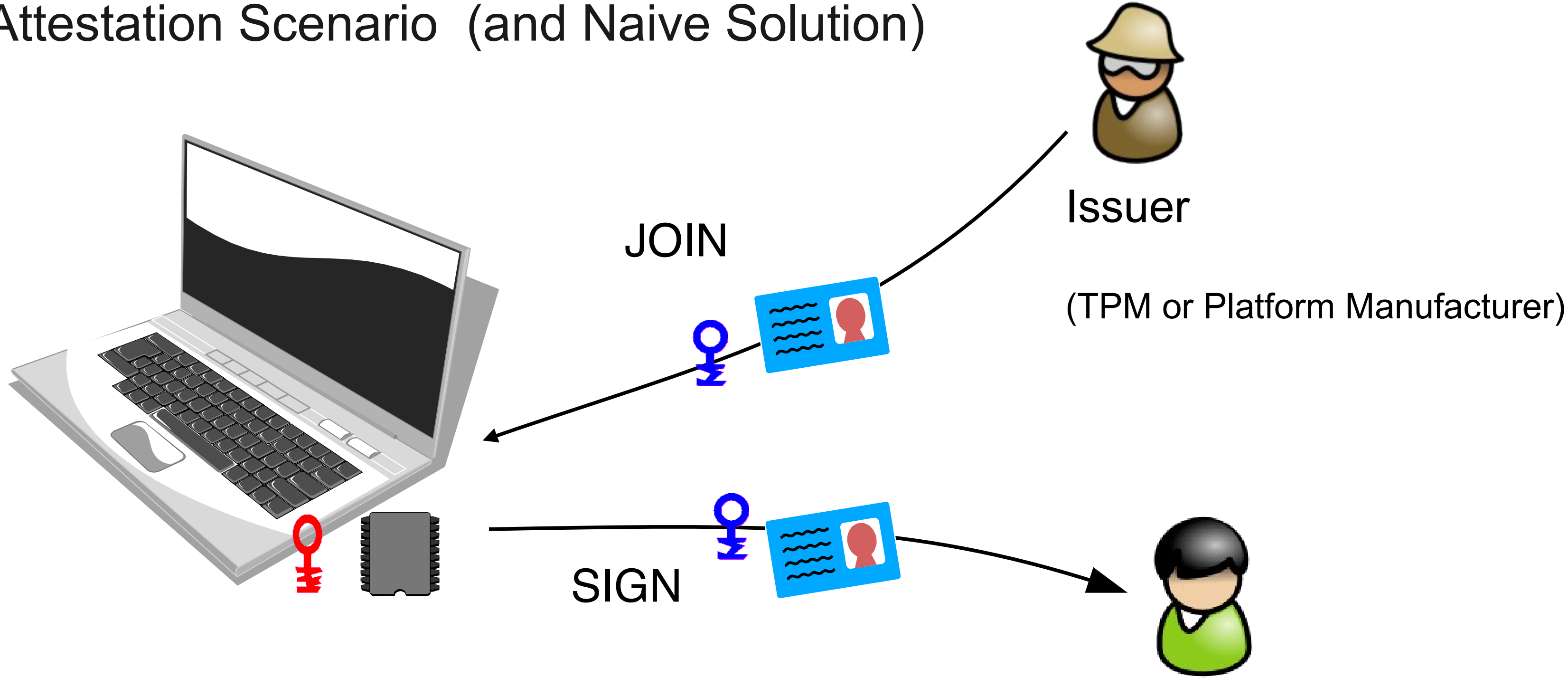
→ Strong authentication of TPM with privacy



Use cases apart from attestation:

- secure access to networks, services, any resources of devices
- can be extended to user of device

Attestation Scenario (and Naive Solution)



Verifier

(Bank, eShop, Tax authority, …)

Problem: using traditional certificates, all transactions of the same platform become linkable :-(

(one could of course give all TPMs the same certificate)

Security Requirements for Attestation (informal & Partial)

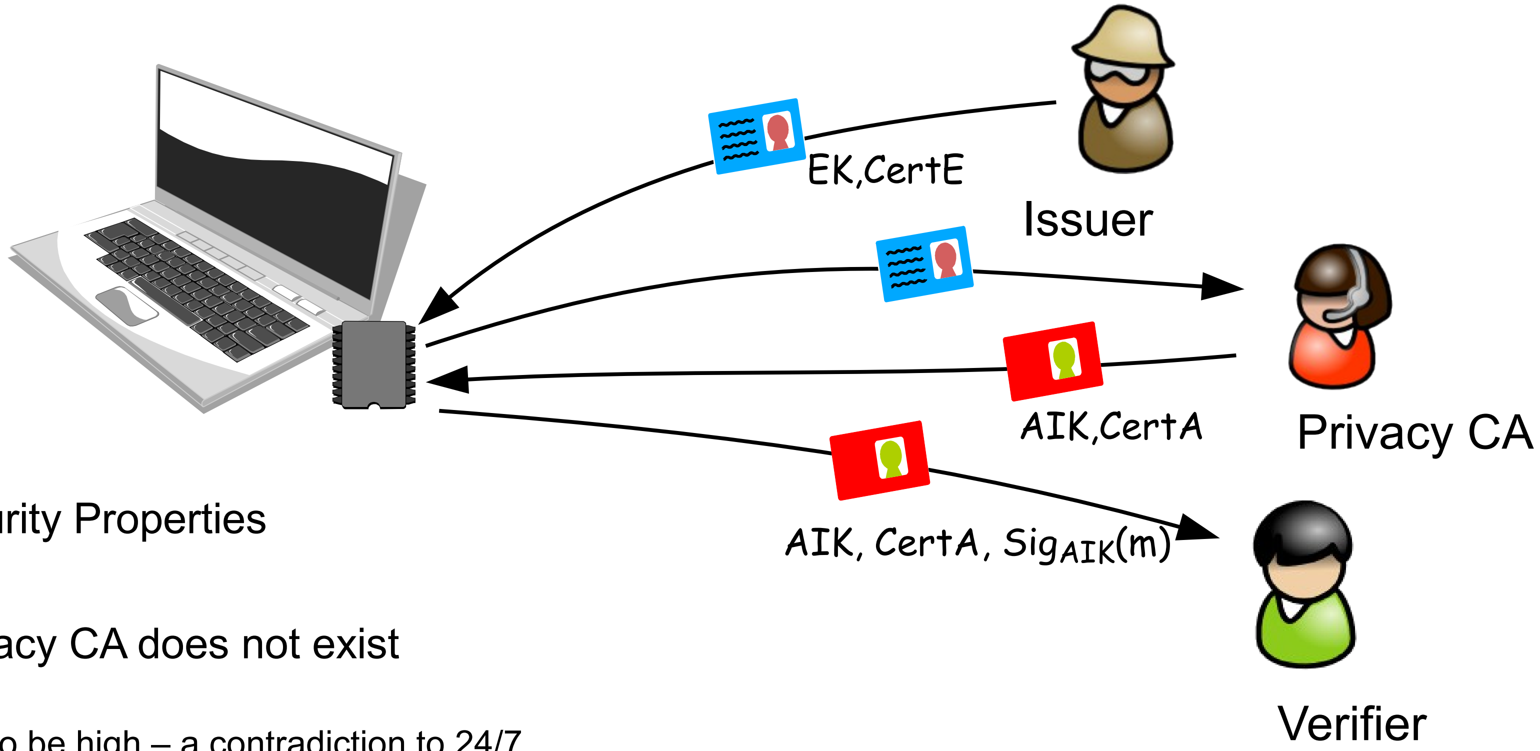
Unforgeability: No adversary can create signatures on messages that were never signed by a certified TPM.

Anonymity: signatures by an honest platform are unlinkable (at least across different domains).

Revocation: If a TPM is compromised, signatures from the compromised keys must no longer be accepted.

Non-frameability: One cannot create a signature on a message that links to an honest platform's signature provided the platform never signed this message.

Attestation – Privacy CA Solution (Traditional Credentials, Still Naive)

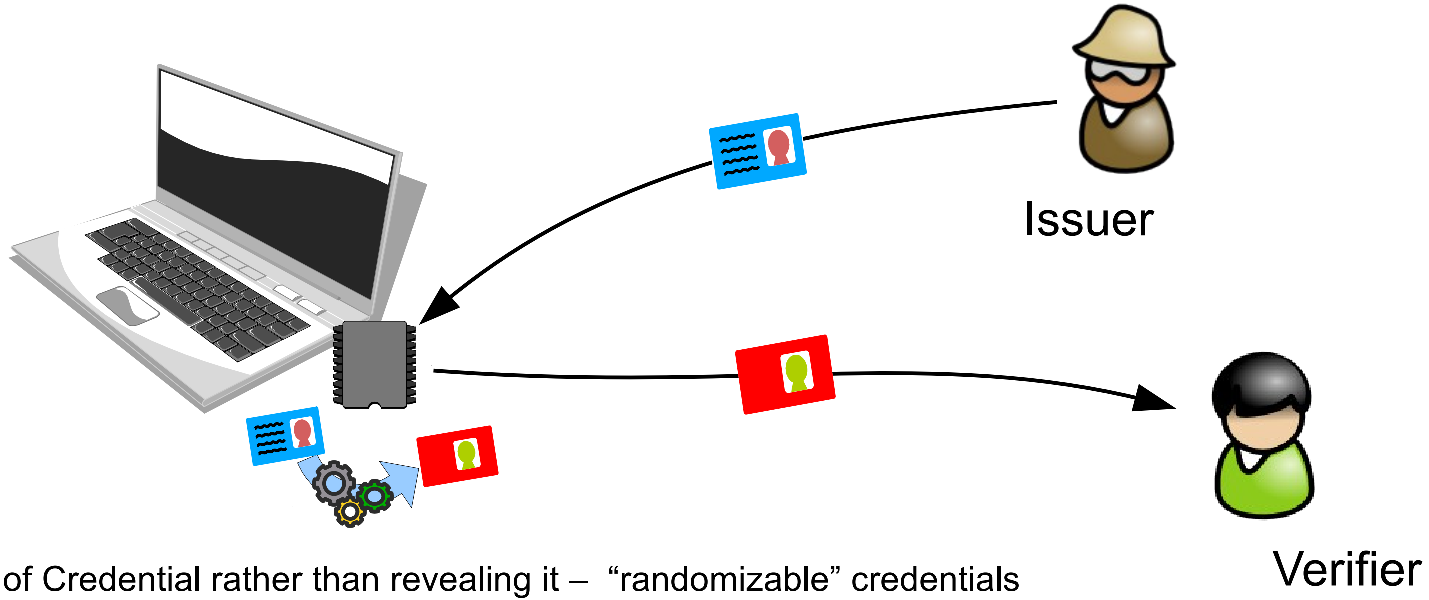


Satisfies Security Properties

Problem: Privacy CA does not exist

- operate 24/7
- security needs to be high – a contradiction to 24/7
- no business model (trust relationship w/ users and verifiers)
- can link transactions! (Big Brother)
- other security requirements would be fulfilled

Direct Anonymous Attestation (Brickell, Camenisch, Chen – 2003)



Proof knowledge of Credential rather than revealing it – "randomizable" credentials

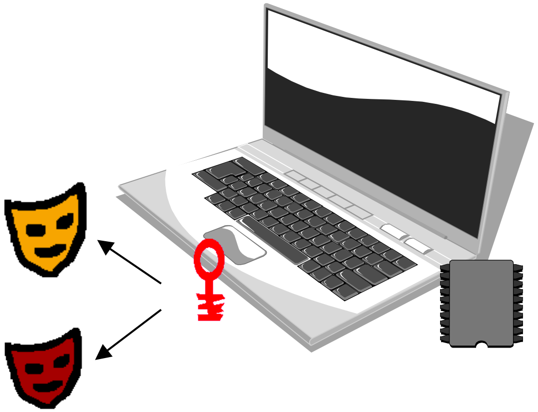
- TPM can transform original credential into new credentials that "looks like" a fresh credential

  → different randomize credentials cannot be linked (anonymity)

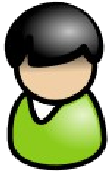
  → still credentials are unforgeable

- *Problem: no means to trace compromised TPMs (too much privacy?)*

Direct Anonymous Attestation (Brickell, Camenisch, Chen - 2003)



Issuer

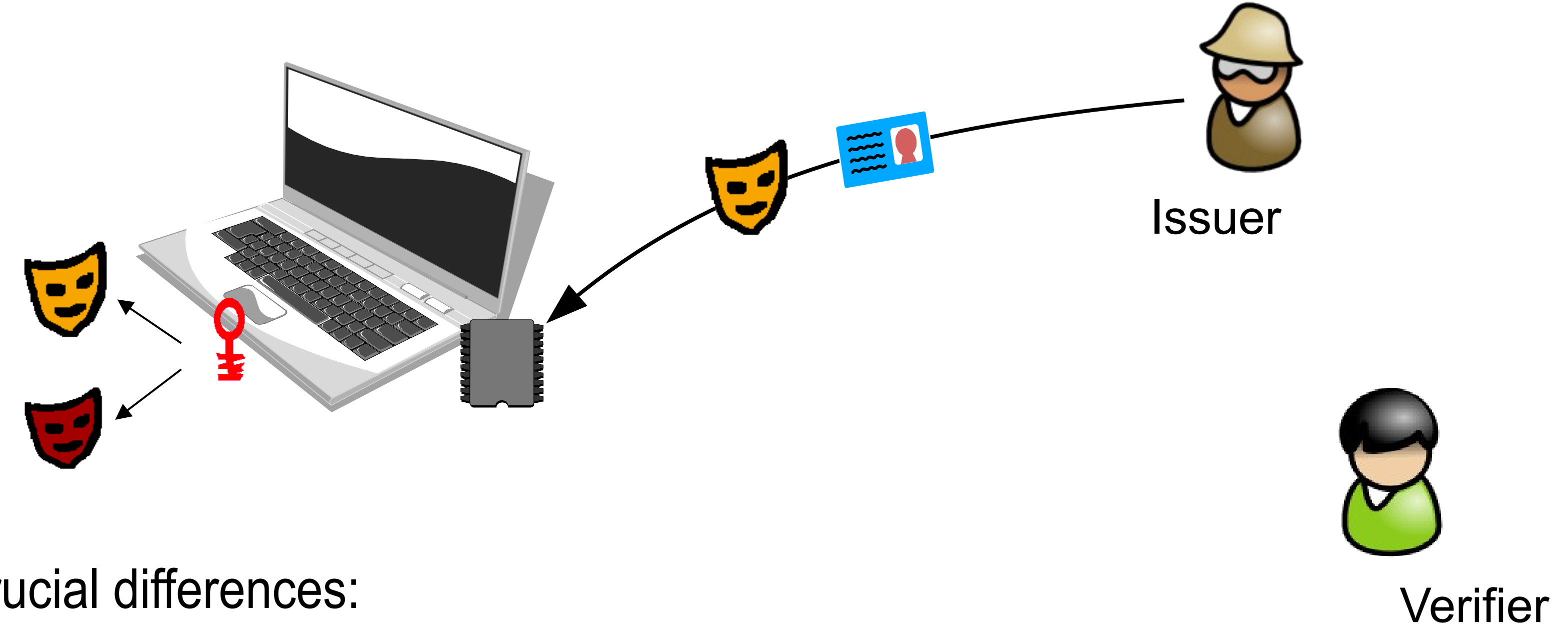


Verifier

Two crucial differences:

1. One secret key - several public keys

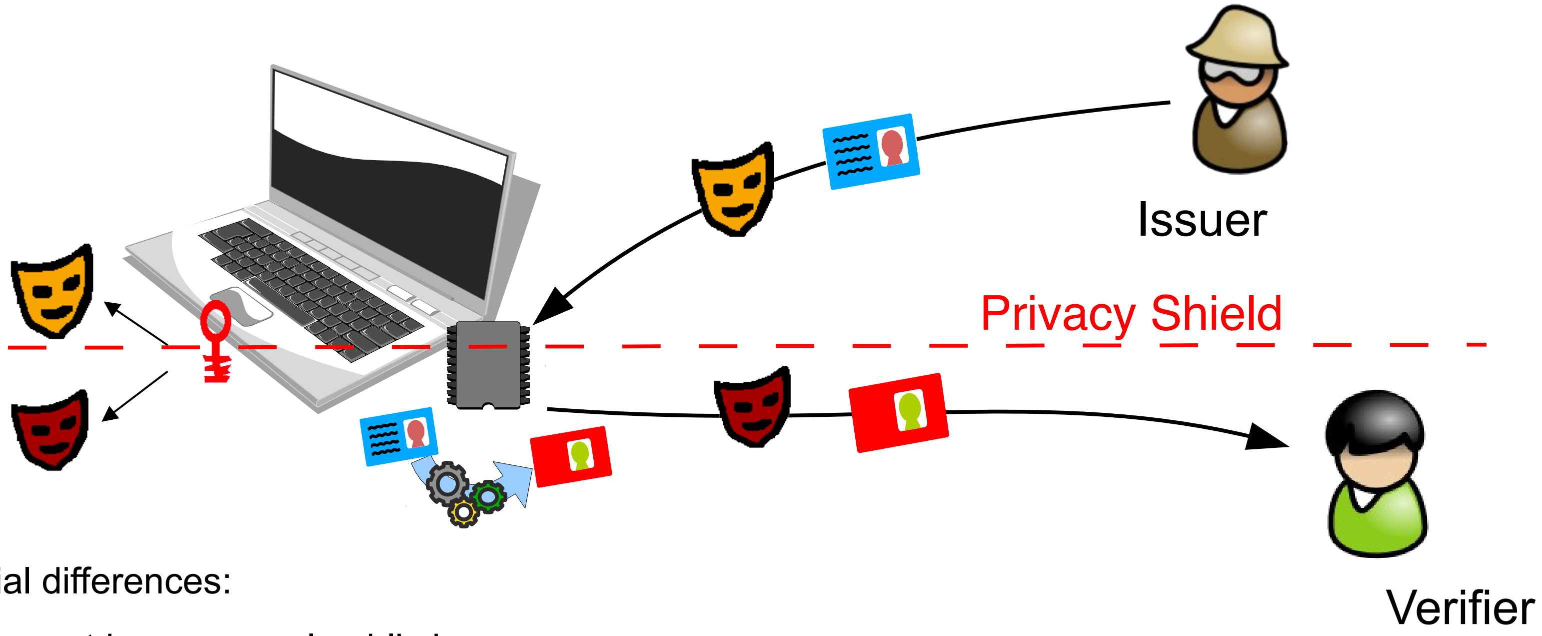
Direct Anonymous Attestation (Brickell, Camenisch, Chen - 2003)



Two crucial differences:

1. One secret key - several public keys

Direct Anonymous Attestation (Brickell, Camenisch, Chen - 2003)



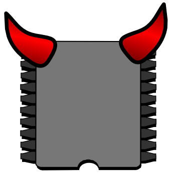
Two crucial differences:

1. One secret key - several public keys
2. Randomizable credentials: original credential into new credentials that “looks like” a fresh credential

   → different randomize credentials cannot be linked (anonymity)

   → still credentials are unforgeable

Direct Anonymous Attestation – Rogue TPMs



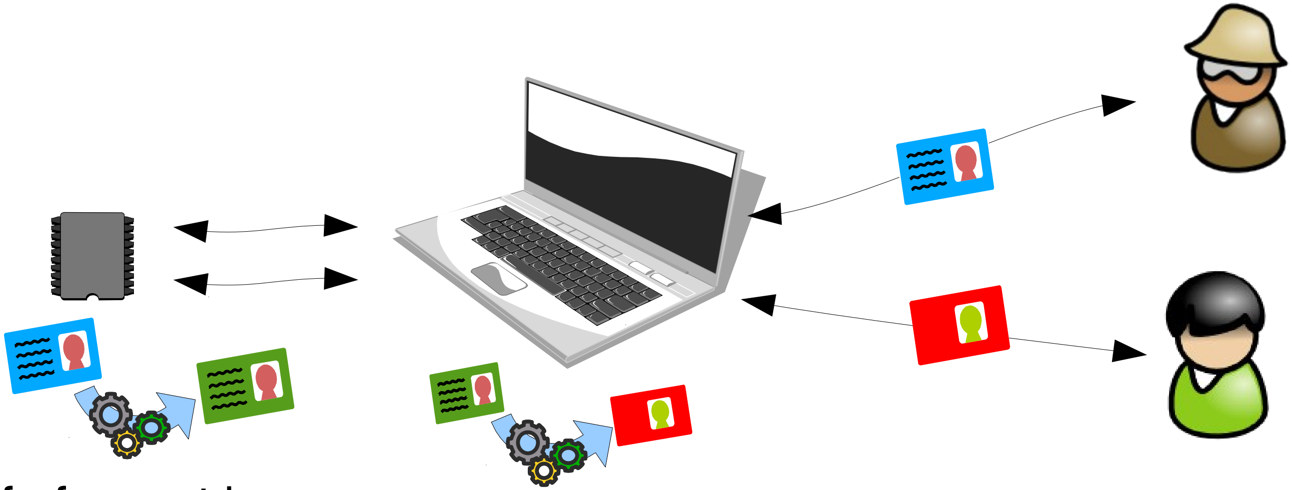
- TPM has been broken and keys have leaked
- Need to be able to distinguish those keys despite signatures are anonymous
- Solution: $Nym = f(DAA\text{-}secret) = \zeta^{DAA\text{-}secret} \bmod p$, where
  - if ζ is random: published keys can be detected,

    protocol is still anonymous
  - if ζ is fixed per verifier, e.g., derived from verifier's name (so-called basename): verifier can also make frequency analysis

    → signature by the same platform w.r.t. same basename can be linked!

    protocol is still pseudonymous
  - Defined via basename: $\zeta = H(basename)$

DAA in implementation: split operation between host and TPM



proof of secret key

ensuring privacy

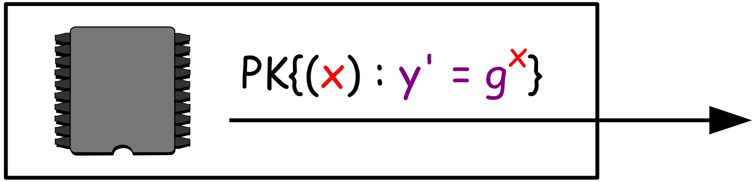
- Split has historic reason: wanna keep TPM small
- In principle easier to build it securely

Direct Anonymous Attestation – Brief History

- TCPA 0.44 – July 2000 until TCPA 1.1b – February 2002
  - w/out DAA, but used Privacy CA
  - Privacy groups criticized Privacy CA solution
- TPM 1.2 – July 2003 until Aug 2009 (revision 116)
  - DAA introduced as alternative to Privacy CA, goal to make privacy groups happy
  - DAA based on RSA
  - Host part specified in TSS (Trusted Software Stack)
  - Implementation on chips very slow (arithmetic co-processor)
- Active research on DAA protocols and schemes, ECC & Discrete Log based
- TPM 2.0 – October 2014
  - Elliptic curve-based DAA
  - ISO standard in 2015 (ISO/IEC 11889)
- Today: Interest in TPM revived, privacy & crypto on agendas
  - Security of mobile devices
  - FIDO authentication

Overview of Changes from TPM 1.2 to TPM 2.0

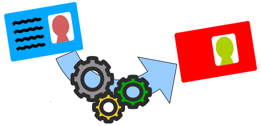
- From RSA groups to elliptic curve groups (faster, smaller keys)
- TPM V1.2 : DAA protocol spec is split between TPM and TSS (Trusted Software Stack) specs. For TPM V2.0, there is not TSS spec.



- On the positive side: supports many different credential signature schemes (CL, q-SDH, …)
- On the negative side:
  - no full specification – Chen & Li 2013 paper hard to match to TPM spec
  - provable security – Chen & Li 2013 security proof broken, current spec. *not provable secure*

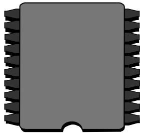
Realization of Direct Anonymous Attestation in TPM V2.0

Preliminaries: Schnorr Signatures

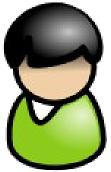


Given a group $\langle g \rangle$ and an element $y \in \langle g \rangle$.

Prover wants to convince verifier that she *knows* $x1, x2$ s.t. $y = g^{x1} h^{x2}$
such that verifier only learns y, g and h.



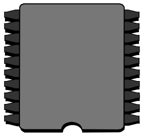
Prover: $PK\{(\alpha,\beta)\colon\ y = g^{\alpha} h^{\beta}\}$ Verifier:



Prover		Verifier
random $r1, r2$		
$t := g^{r1} h^{r2}$	$\xrightarrow{\quad t \quad}$	
	$\xleftarrow{\quad c \quad}$	random c
$s1 := r1 - cx1$	$\xrightarrow{\quad s1,\ s2 \quad}$	$t = y^{c} g^{s1} h^{s2}$
$s2 := r2 - cx2$		

Preliminaries: Schnorr Signatures

From Protocol $PK\{(\alpha,\beta): y = g^{\alpha} h^{\beta}\}$ to Signature $SPK\{(\alpha): y = g^{\alpha}\}(m)$:



Signing a message m:

- chose random $r1, r2 \in Z_q$ and
- compute $(c, s1, s2) := (H(g^{r1} h^{r2} || m),\ r1 - cx1\ ,\ r2 - cx2\ )$

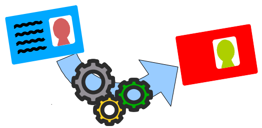
Verifying a signature $(c, s1, s2)$ on a message m:

- check $c = H(y^{c} g^{s1} h^{s2} || m)$?

Security:

- Discrete Logarithm Assumption holds
- Hash function $H(.)$ behaves as a "random oracle."

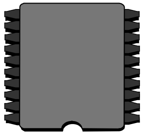
Preliminaries: Schnorr Signatures



Protocol can be extended to prove logical combination of terms

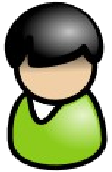
$$PK\{(\alpha,\beta): \; y = g^{\alpha} \wedge z = g^{\beta} \wedge u = g^{\beta}h^{\alpha}\}$$

$$PK\{(\alpha,\beta): \; y = g^{\alpha} \vee z = g^{\beta}\}$$



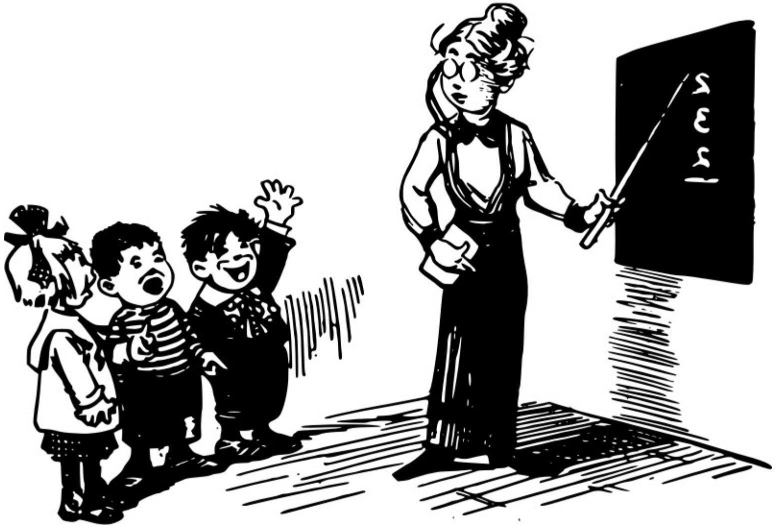
Prover:

Verifier:



Basics of Bi-Linear Maps

- Two groups G, Gt of order q (elliptic curve groups)
- Bi-linear map $e: G\text{x}G \rightarrow Gt$ with the following properties
  - Bi-linear: $e(g^a,h^b) = e(g,h)^{ab} = e(g^b,h^a)$
  - Non-degenerate: $e(g,g) \neq 1$
  - Efficiently computable: $(g,h) \rightarrow e(g,h)$



Remarks:

- Given $e(g,h)$ and g it is hard to compute h
- Bi-linear maps makes Decisional Diffie-Hellman in G easy:

  Recall DDH: distinguish between (g,g^a,g^b,g^{ab}) and (g,g^a,g^b,g^c)
  - $e(g^a,g^b) \quad = \quad e(g,g^{ab}) \quad \neq e(g,g^c)$?
- Often: $e: G1\text{x}G2 \rightarrow Gt$

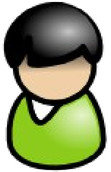
Signature Scheme used to Issue Certificate to TPM

Public key of signer: G, Gt of order q, generators $g, h, h_0, \ldots, h_k$, and element y

Secret key: value x such that $y = g^x$

To sign k messages $m1, \ldots, mk \in Z_q$:

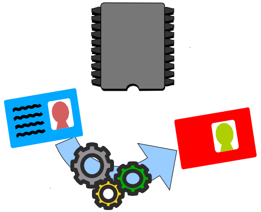
- choose random element $r, s \in Z_q$
- compute $A := (g \cdot h_0^{\,s} \cdot h_1^{\,m1} \cdot \ldots \cdot h_k^{\,mk})^{1/(x+r)}$
- signature is (A, r, s)



Verification: $e(A,y) \cdot e(A,g)^r = e(g \cdot h_0^{\,s} \cdot h_1^{\,m1} \cdot \ldots \cdot h_k^{\,mk}, g)$

(because: $e(A,g^x) \cdot e(A,g)^r = e(A^{(x+r)},g)$ we must have $A^{(x+r)} = g \cdot h_0^{\,s} \cdot h_1^{\,m1} \cdot \ldots \cdot h_k^{\,mk}$)

Signature Scheme used to Issue Certificate to TPM – Proof of Signature



Observe:

Let $A' = Ah^{t1}$ and $B = g^{t1}h^{t2}$ with random $t1, t2$

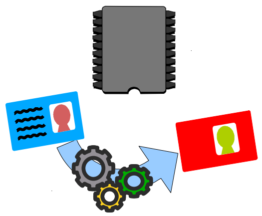
$$Ve(A,y)\cdot e(A,g)^r = e(g\cdot h_0^{s}\cdot h_1^{m1}\cdot\ldots\cdot h_k^{mk},g)$$

$$e(A',y)\cdot e(A',g)^r = e(Ah^{t1},y)\cdot e(Ah^{t1},g)^r = e(h,y)^{t1}e(h,g)^{t1\,r}e(g\cdot h_0^{s}\cdot h_1^{m1}\cdot\ldots\cdot h_k^{mk},g)$$

To prove ownership of a signature (A,e,s) on some on $m1, \ldots, mk$ execute proof protocol

$$PK\{(t1,t2,t3,t4,r,s,m1,\ldots,mk) : \quad B = g^{t1}h^{t2} \quad \wedge \quad 1 = B^r g^{-t3}h^{t4} \quad \wedge$$
$$e(A',y)\cdot e(A',g)^r = e(h,y)^{t1}\cdot e(h,g)^{t3}\cdot e(g\cdot h_0^{s}\cdot h_1^{m1}\cdot\ldots\cdot h_k^{mk},g)\}$$

Using this scheme for TPM 2.0



TPM secret key m1

Attributes m2, ..., mk

Need to include basename, so we have

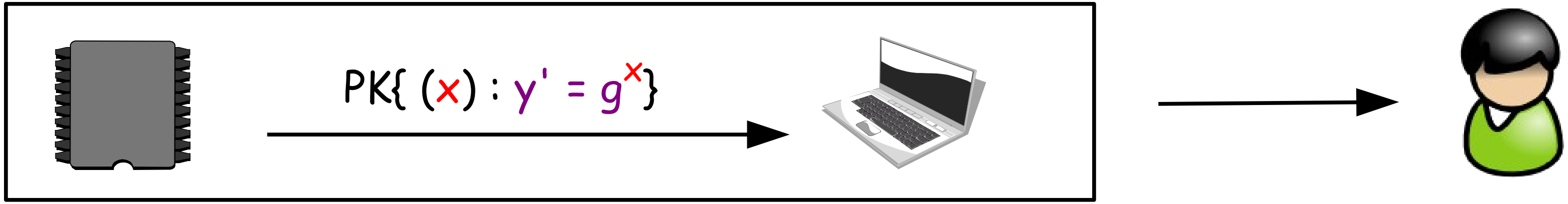
$$PK\{(t1,t2,t3,t4,r,s,m1,....,mk) : \quad B = g^{t1}h^{t2} \quad \wedge \quad 1 = B^{r}g^{-t3}h^{t4} \wedge$$

$$e(A',y) = e(h,y)^{t1} \cdot e(A',g)^{-r} \cdot e(h,g)^{t3} \cdot e(g \cdot h_0^{s} \cdot h_1^{m1} \cdot ... \cdot h_k^{mk}, g) \quad \wedge$$

$$Nym = H(basename)^{m1} \}$$

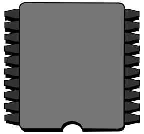
How the TPM and the Host Sign Jointly (simplyfied)

$$PK\{(x, z) : y = g^x h^z\}$$

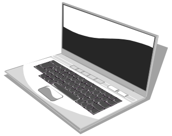


How the TPM and the Host Sign Jointly (simplyfied)

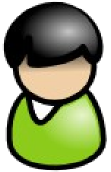
$PK\{ (x, z) : y = g^x h^z \}$



$PK\{ (x) : y' = g^x \}$ $\longrightarrow$



$\longrightarrow$



random $r1$

$t' = g^{r1}$

$\xrightarrow{t'}$

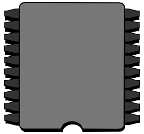
random $r2$

$t = t' h^{r2}$

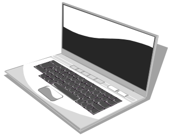
$\xrightarrow{t}$

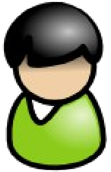
How the TPM and the Host Sign Jointly (simplyfied)

$PK\{(x, z) : y = g^{x}h^{z}\}$



$PK\{(x) : y' = g^{x}\}$





random $r1$

$t' = g^{r1}$

t'

c

random $r2$

$t = t'h^{r2}$

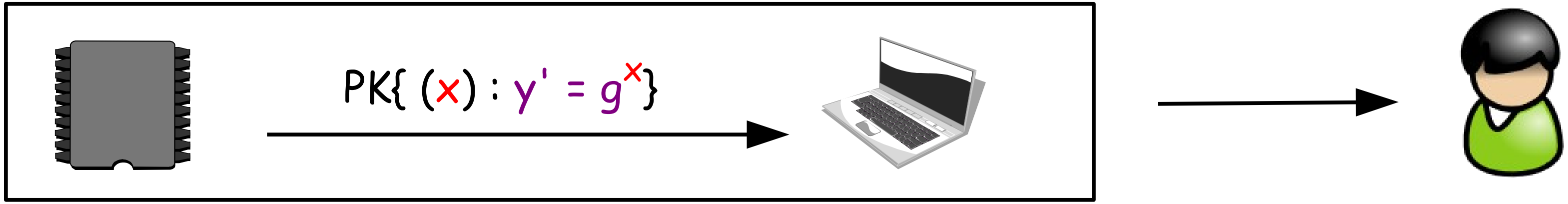
t

c

random c

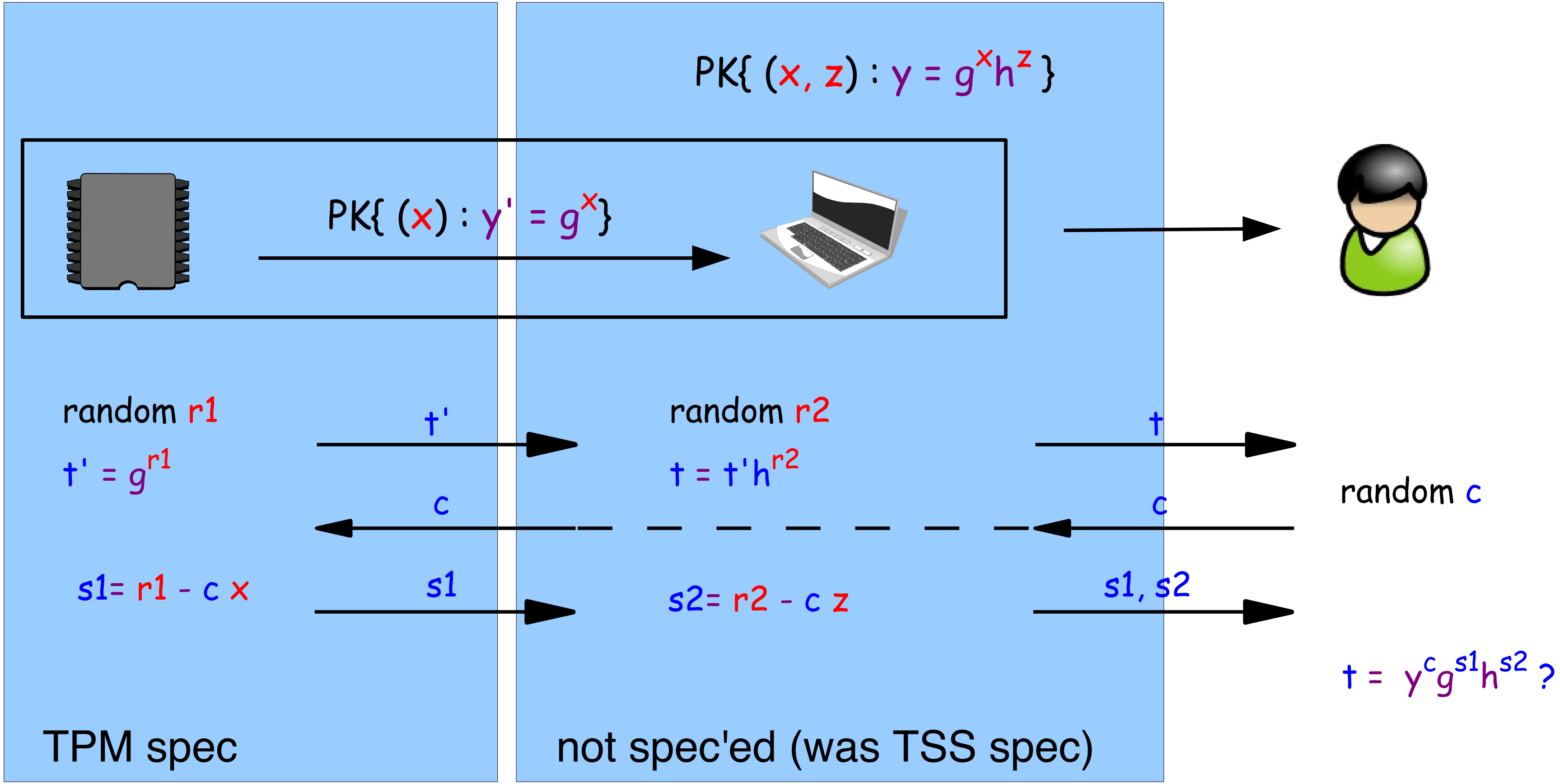
How the TPM and the Host Sign Jointly (simplyfied)

$PK\{(x, z) : y = g^{x}h^{z}\}$

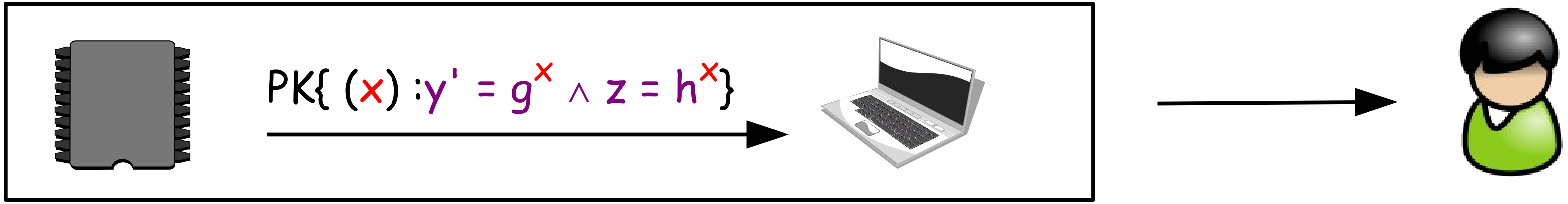


random $r1$		random $r2$		
$t' = g^{r1}$	$\xrightarrow{t'}$	$t = t'h^{r2}$	$\xrightarrow{t}$	
	$\xleftarrow{c}$		$\xleftarrow{c}$	random c
$s1 = r1 - c\,x$	$\xrightarrow{s1}$	$s2 = r2 - c\,z$	$\xrightarrow{s1,\ s2}$	
				$t = y^{c}g^{s1}h^{s2}$?

How the TPM and the Host Sign Jointly (simplyfied)



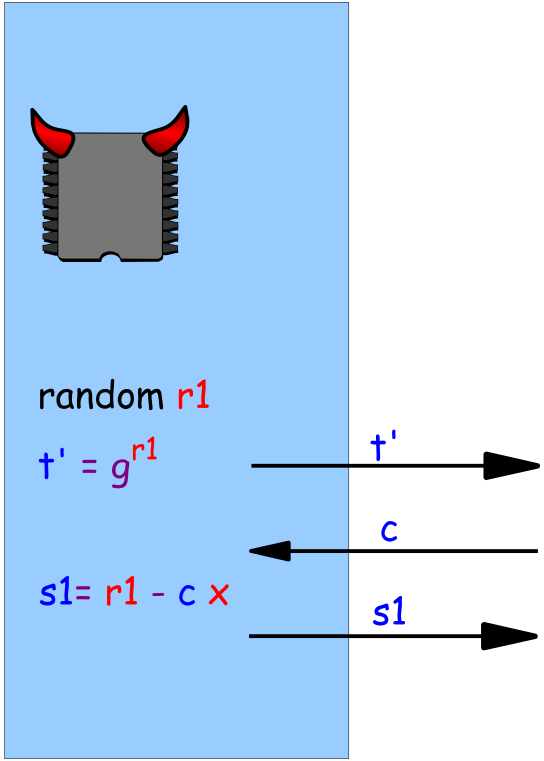
How the TPM and the Host Sign Jointly



$$PK\{(t1,t2,t3,t4,r,s,m1,\ldots,mk) : \quad B = g^{t1}h^{t2} \;\wedge\; 1 = B^{r}g^{-t3}h^{t4} \;\wedge$$
$$e(A',y) = e(h,y)^{t1}\cdot e(A',g)^{-r}\cdot e(h,g)^{t3}\cdot e(g\cdot h_0^{\,s}\cdot h_1^{\,m1}\cdot\ldots\cdot h_k^{\,mk},g) \;\wedge$$
$$Nym = H(basename)^{m1}\}$$

- TPM does proof for two bases, g and $h = H(basename)$
- TPM does not need to know about target group, all ops in $G1$
- TPM part of protocol can be extended to:
  - Include attributes
  - Key-binding for credentials

Security w.r.t. corrupted TPM – How Much Privacy & Security Can We Get?



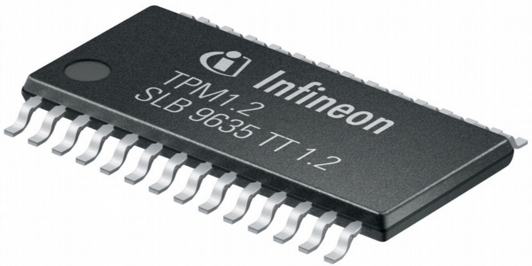
Problem: TPM could leak keys, identity, etc via its messages

Limits for Schnorr-based proofs

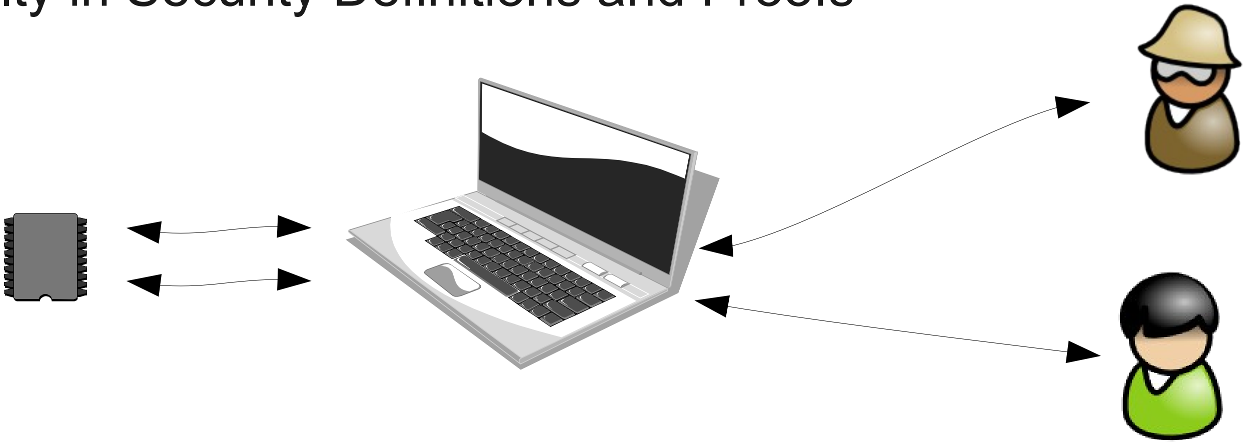
- Can randomize t and $s1$ values, but not c
  - Otherwise security cannot be proved in RO
- Thus TPM could deny certain c values!
- Way out: monitor behavior, but this is tricky
  - Need to inspect messages signed by TPM

For non-Schnorr-based Proofs → Crypto '17 paper

Security Proofs



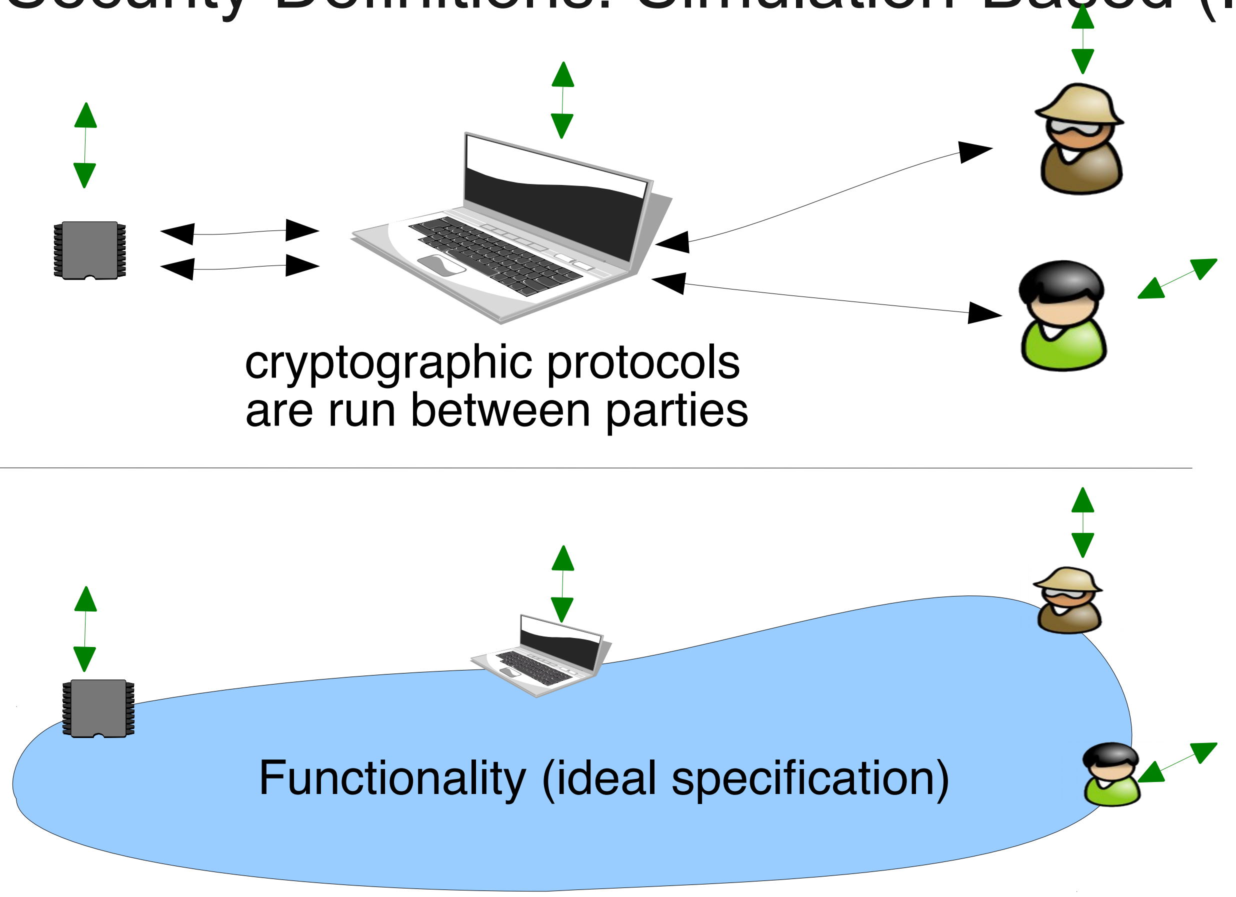
Difficulty in Security Definitions and Proofs



- 4 parties & 4 protocols → complex protocol and thus security definition becomes complex
  - Privacy: as long as Host is honest
  - Security: as long as Issuer is honest; additional but essential guarantees if TPM is honest
  - Need to consider Host and TPM as different parties
  - Malicious TPM must not be able to communicate with malicious Verifier and Issuer
- After initial DAA paper (Brickell et al. 2004), a number of improved security definitions where published.
- All of them have issues, some of them severe, allowing for insecure schemes :-(

→ Need for complete security model & provably secure schemes

Security Definitions: Simulation-Based (Ideal/Real; UC)



Interaction
with environment

secure if environment
cannot tell apart

Interaction
with environment

Existing Simulation-Based Models for DAA

Brickell, Camenisch, Chen (2004)

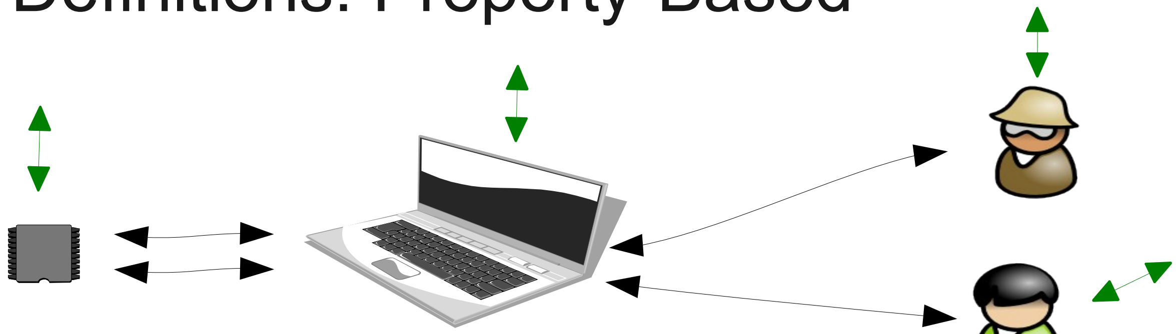
- Does not output any signature values

  → Prohibits working with signature values in practice

Chen, Morrissey, Smart (2009)

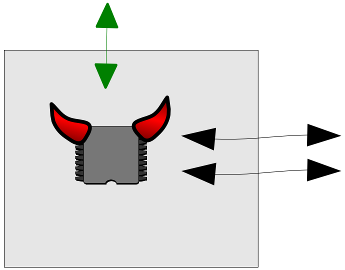
- Outputs signatures
- Signature generation too simplistically modeled → cannot be realized

Security Definitions: Property-Based



cryptographic protocols

Defines security when interacting with cryptographic protocol for each property separately.



E.g., Non-frameability: One cannot create a signature on a message that links to an honest platform's signature when the platform never signed this message.

Existing Property-Based Models for DAA

Brickell, Chen, Li (2009)

- Unforgeability not captured: trivially forgeable scheme can be proven secure
- No property for non-frameability

Chen (2010)

- Extends BCL'09 with non-frameability
- Same flaws as BCL'09

Bernhard et al. (2013)

- Discusses flaws in all previous models
- Pre-DAA: TPM + Host one party
- Does not cover honest TPM embedded into corrupt Host
- Security Proof of "Pre-DAA" does not work for full DAA

Full UC Functionality and Security Proof

Camenisch, Drijvers, Lehmann 2016 (ia.cr/2015/1246)

Comprehensive security model in UC framework (i.e., simulation based)

- Allows composition by composition theorem
- Signatures modeled as concrete values that are sent as output
- TPM and Host separate parties
- Extensive explanation on why this definition properly captures the security requirements

Provide scheme that realize the functionality

- Two provably secure instantiation (based on LRSW and q-SDH, respectively)
- As efficient as existing DAA schemes – essentially just doing a few details right

DAA Protocols for TPM2.0

TPM2.0 offers generic APIs to support various schemes, e.g.,
DAA based on LRSW (CL-signature) & qSDH (BBS+ signature)

Insecure (no privacy & forgeable)

Trivially forgeable
(but easy to fix)

Requires rewinding
& DDH oracle

Trivially forgeable
(but easy to fix)

LRSW: ~~BCL08 & BCL09~~ → ~~CMS08a & CMS08b & CMS09~~ → ~~CPS10 ISO 20008-2~~ → ~~BFG+13~~ → ~~CU15~~ → CDL16a

Provably secure
DDH & LRSW

qSDH: CF08 inefficient → ~~BL08~~ → ~~Che09~~ → ~~BL10 ISO 20008-2~~ → ~~CU15~~ → CDL16b

Unforgeability proof is wrong (and doesn‘t seem fixable)

Provably secure
DDH & qSDH
Incl. Sign.-based
revocation

Many incorrect security proofs

Provably secure schemes incompatible with current TPM2 spec

Some issues fixed in latest spec

Recent: S&P '17 - minimal changes to TPM2 spec; Crypto '17 - best privacy

Do we need all these definitions?

(1, 1, 1, 1) is a valid credential on *any* key in Chen, Page, Smart 2010

- ISO 20008 standardized!

TPM2 spec contains static DH oracle

- Larger groups and keys required (Xi et al., 2014)

TPM2 should make zero-knowledge proof

- Problem in hash computation
- Proof not zero-knowledge

Conclusions

- Try for yourself – code is open source
- Device authentication more relevant than ever
- Provable security matters – a number of standards have issues
- It often takes far longer than one would expect & still not done
- Privacy & security still to be achieved – DAA just a special case

Thanks! Questions?

ia.cr/2015/1246

jca@zurich.ibm.com

@JanCamenisch

References

Bernhard, D., Fuchsbauer, G., Ghadafi, E., Smart, N., Warinschi, B.: Anonymous attestation with user-controlled linkability. International Journal of Information Security 12(3), (2013)

Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. ACM CCS 2004.

Brickell, E., Chen, L., Li, J.: Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. International Journal of Information Security 8(5), (2009)

Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. CRYPTO 2004.

Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous Attestation Using the Strong Diffie-Hellman Assumption Revisited. TRUST 2016: 1-20

Camenisch, J., Drijvers, M., Lehmann, A.: Universally Composable Direct Anonymous Attestation. Public Key Cryptography (2) 2016: 234-264

Chen, L., Morrissey, P., Smart, N.: DAA: Fixing the pairing based protocols. ePrint Archive, Report 2009/198.

Chen, L.: A DAA scheme requiring less tpm resources. Information Security and Cryptology 2010.

Chen, L., Morrissey, P., Smart, N.: On proofs of security for DAA schemes. Provable Security 2008.

Chen, L., Page, D., Smart, N.: On the design and implementation of an efficient DAA scheme. Smart Card Research and Advanced Application 2010.

Chen, L., Li, J.: Flexible and scalable digital signatures in TPM 2.0. ACM CCS 2013.

Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. SAC 1999.

Xi, L., Yang, K., Zhang, Z., Feng, D.: DAA-related APIs in TPM 2.0 revisited. Trust and Trustworthy Computing 2014