Direct Anonymous Attestation Revisited

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
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)
- TPM 2.0 – October 2014
  - Elliptic curve-based DAA
  - ISO standard in 2015 (ISO/IEC 11889)
- Today: Interest in TPM revived
  - Security of mobile devices
  - FIDO authentication
Attestation Scenario

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

Issuer (TPM or Platform Manufacturer)

Privacy barrier

Verifier (Bank, eShop, Tax authority, …)
Security Requirements for Attestation

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

_Anonymity:_ signatures by an honest platform are unlinkable (without basename or different basenames).

_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.

_Revocation:_ If a TPM is compromised, signatures from the compromised keys must no longer be accepted.
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!
- other security requirements would be fulfilled
Direct Anonymous Attestation (Brickell, Camenisch, Chen - 2003)

DAA credentials are “randomizable”:

- 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
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: $Nym = f(DAA-secret) = \zeta^{DAA-secret} \mod p$, where
  - if $\zeta$ is random: published keys can be detected,
    \[ protocol \ is \ still \ anonymous \]
  - if $\zeta$ is fixed per verifier, e.g., derived from verifier's name (so-called basename): verifier can also make frequency analysis
    \[ \rightarrow \ \text{signature by the same platform w.r.t. same basename can be linked!} \]
    \[ protocol \ is \ still \ pseudonymous \]
DAA in implementation: split operation between host and TPM

proof of secret key

ensuring privacy
Direct Anonymous Attestation in TPM V2.0
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 no 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
- no security proof – Chen & Li 2013 security proof broken, current spec. not provable secure
Difficulty in Security Definitions and Proofs

- 4 parties & 4 protocols → complex protocol and thus security definition becomes complex
- After initial DAA paper (Brickell et al. 2004), a number of improved security definitions were published.
- All of them have issues, some of them severe, allowing for insecure schemes :-(

→ Need for complete security model & provably secure schemes
Simulation-Based Security Definitions

Functionality (ideal specification)

Cryptographic protocols are run between parties

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 to be realizable
Property-Based Security Definitions

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
- TPM + Host one party
- Does not cover honest TPM in corrupt Host
- Security Proof of “Pre-DAA” does not work for full 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)

**LRSW:**
- Insecure (no privacy & forgeable)
  - BCL08 & BCL09
  - CMS08a & CMS08b & CMS09
  - CPS10
    - ISO 20008-2
  - BFGSW13
  - CU15
  - CDL16a
    - Provably secure under DL, DDH & LRSW

**qSDH:**
- CF08 inefficient
- BL07 EPID (No split of TPM & host)
  - Che09
  - BL10
    - ISO 20008-2
  - CU15
  - CDL16b
    - Provably secure under DDH & qSDH

- All existing schemes are either insecure, or cannot be proven secure
  - (1, 1, 1, 1) is a valid credential on any attestation key in [CPS10] – ISO 20008 standardized!
- Revised provably secure protocols [CDL16a, CDL16b], as efficient as existing schemes – mainly details that had to be fixed
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
Comprehensive Model and Secure Protocol
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
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 $x_1, x_2$ s.t. $y = g^{x_1} h^{x_2}$ such that verifier only learns $y, g$ and $h$.

```
Prover:

random $r_1, r_2$

$t := g^{r_1} h^{r_2}$

$s_1 := r_1 - c x_1$

$s_2 := r_2 - c x_2$

Verifier:

$t = y^c g^{s_1} h^{s_2}$

PK{$(\alpha, \beta)$: $y = g^\alpha h^\beta$}
Prelimiaries – 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 $r_1, r_2 \in \mathbb{Z}_q$ and
- compute $(c, s_1, s_2) := (H(g^{r_1} h^{r_2} \| m), r_1 - c x_1, r_2 - c x_2)$

Verifying a signature $(c, s_1, s_2)$ on a message $m$:
- check $c = H(y^c g^{s_1} h^{s_2} \| m)$?

Security:
- Discrete Logarithm Assumption holds
- Hash function $H(\cdot)$ behaves as a “random oracle.”
Bi-Linear Maps

- Two groups $G, G^\dagger$ of order $q$ (elliptic curve groups)
- Bi-linear map $e: G \times G \rightarrow G^\dagger$ 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 $(g, g^a, g^b, g^c)$
  - $e(g^a, g^b) = e(g, g^{ab})$
  - $e(g^a, g^b) \neq e(g, g^c)$
Signature Scheme used to Issue Certificate to TPM

Public key of signer: $G, G^t$ of order $q$, generators $g, h, h_0, ..., h_k$, and element $y$

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

To sign $k$ messages $m_1, ..., m_k \in \mathbb{Z}_q$:

- choose random element $r, s \in \mathbb{Z}_q$
- compute $A := (g \cdot h_0^s \cdot h_1^{m_1} \cdot ... \cdot h_k^{m_k})^{1/(x+r)}$
- signature is $(A, r, s)$

Verification: $e(A, y) \cdot e(A, g)^r = e(g \cdot h_0^s \cdot h_1^{m_1} \cdot ... \cdot h_k^{m_k}, g)$

(because: $e(A, g^x) \cdot e(A, g)^r = e(A^{(x+r)}, g)$ we must have $A^{(x+r)} = g \cdot h_0^s \cdot h_1^{m_1} \cdot ... \cdot h_k^{m_k}$)
Signature Scheme used to Issue Certificate to TPM – proof of signature

Observe:

\[ d = c^e a_1^{m1} \cdots a_k^{mk} b^s \mod n \]

Let \( A' = A h^{t1} \) and \( B = g^{t1} h^{t2} \) with randomly chosen \( t1, t2 \)

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

\[
\text{PK}\{(t1, t2, t3, t4, r, s, m1, \ldots, mk) : \}
\]

\[
B = g^{t1} h^{t2} \land 1 = B^r g^{-t3} h^{t4}
\]

\[
e(A', y) \cdot e(h, y)^{-t1} \cdot e(A', g)^r \cdot e(A', g)^{-t3} = e(g \cdot h_0^s \cdot h_1^{m1} \cdots h_k^{mk}, g)
\]
Using this scheme for TPM 2.0

TPM secret key \( m_1 \)
Attributes \( m_2, \ldots, m_k \)

Proof becomes proof protocol

\[ PK\{(t_1,t_2,t_3,t_4,r,s,m_1,\ldots,m_k) : \]
\[ B = g^{t_1}h^{t_2} \land 1 = B^rg^{-t_3}h^{t_4} \land \]
\[ e(A',y)\cdot e(h,y)^{-t_1}\cdot e(A',g)^r\cdot e(A',g)^{-t_3} = e(g\cdot h_0^s\cdot h_1^{m_1}\cdot \ldots \cdot h_k^{m_k},g) \land \]
\[ Nym = H(basename)^{m_1} \} \]
How the TPM and the Host Sign Jointly (simplified)

\[ \text{PK}\{ (x, m) : y = g^x h^m \} \]
How the TPM and the Host Sign Jointly (simplified)

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

random \ r_1
\ t' = g^{r_1}

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

}\ t'

random \ r_2
\ t = t' h^{r_2}

}\ t
How the TPM and the Host Sign Jointly (simplyfied)

\[
PK\{ (x, m) : y = g^x h^m \}
\]

random \( r_1 \)

\( t' = g^{r_1} \)

t' → PK\{ (x) : y' = g^x \} →

random \( r_2 \)

\( t = t'h^{r_2} \)

t →

random \( c \)

c →
How the TPM and the Host Sign Jointly (simplyfied)

\[ t' = g^{r_1} \]

random \( r_1 \)

\[ s_1 = r_1 - c \times x \]

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

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

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

random \( r_2 \)

\[ t = t'h^{r_2} \]

\[ t' = g^{r_1} \]

\[ t = y^{c g^{s_1} h^{s_2}} \]

\[ s_2 = r_2 - c \times m \]

random \( c \)

\[ s_1 = r_1 - c \times x \]
How the TPM and the Host Sign Jointly (simplyfied)

\[ t' = g^{r_1} \]

\[ t = t'h^{r_2} \]

\[ s_1 = r_1 - c \times x \]

\[ s_2 = r_2 - c \times m \]

\[ t = y^c g^{s_1} h^{s_2} \]
Next Steps

TPM 2.0

- working on fixing security problems
- trying to unify different schemes
- spec of full schemes, i.e., also issuer, host, verifier parts.

FIDO anonymous authenticator spec

- with or without TPM 2.0
- reference implementation underway (aim at open sourcing it)
Conclusions

- 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
Thanks! Questions?

ia.cr/2015/1246
jca@zurich.ibm.com
@JanCamenisch
References


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