Trusted Computing: Introduction & Applications

Lecture 5: Remote Attestation, Direct Anonymous Attestation

Dr. Andreas U. Schmidt

Fraunhofer Institute for Secure Information Technology SIT,
Darmstadt, Germany

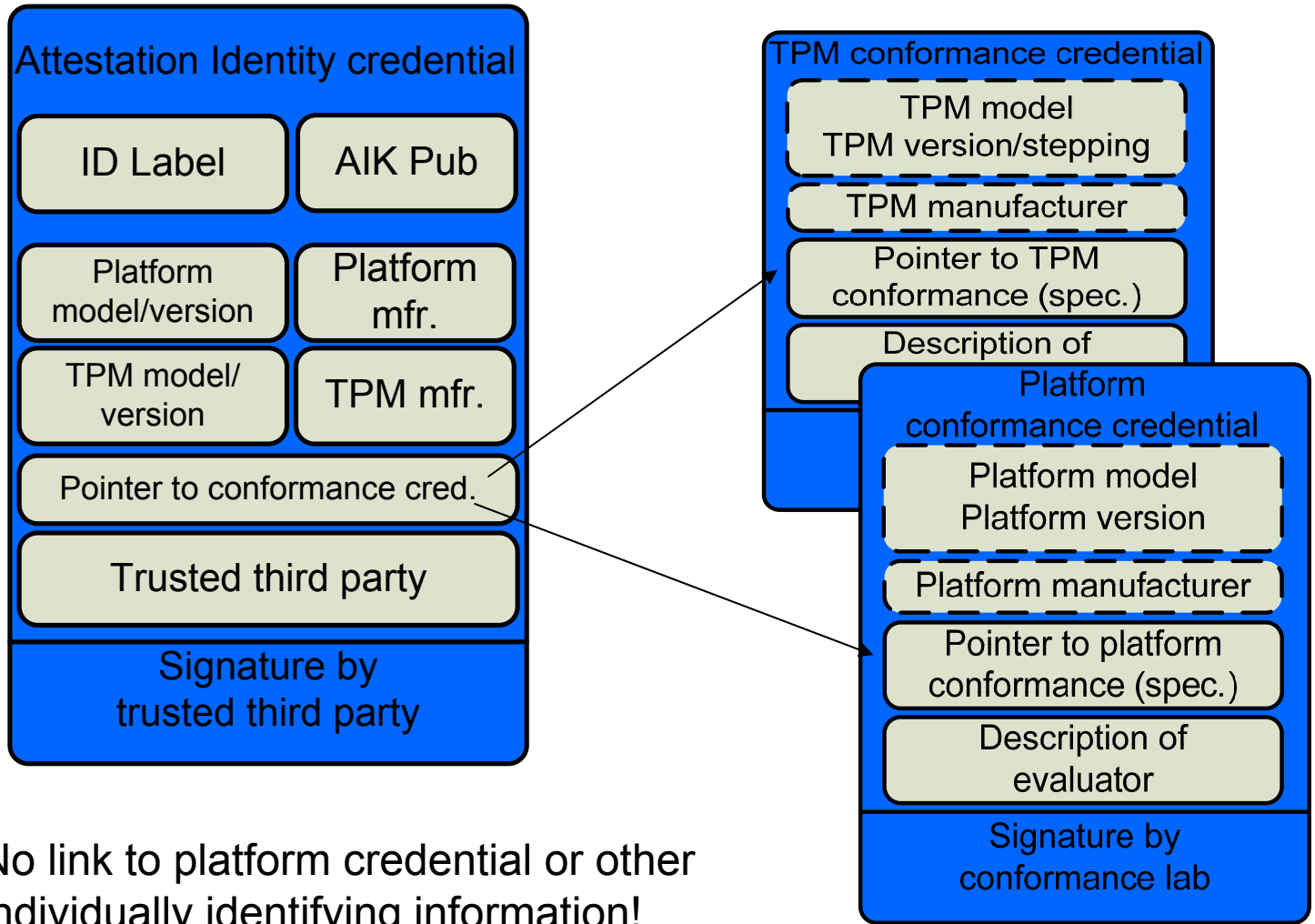
Literature

1. C. J. Mitchell (ed.), *Trusted Computing*. IEE Press, 2005.
2. Eimear Gallery, Graeme Proudler Lecture at Royal Holloway: http://www.isg.rhul.ac.uk/files/IY5608_-_Lecture_3_Roots_of_Trust.pdf
3. Sven Lachmund: Talk at the CAST-Workshop Trusted Computing, Darmstadt, May 2007
4. J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In EUROCRYPT 2001, vol. 2045 of LNCS, pp. 93–118. Springer Verlag, 2001.
5. E. Brickell(IBM), J. Camenisch(Intel), and L. Chen (HP), *Direct Anonymous Attestation,*ACM Conference on Computer and Communications Security (CCS), pp. 132--145, October 2004.
6. Jan Camenisch and ElsVan Herreweghen Design and Implementation of the idemix Anonymous Credential System CCS'02, November 18–22, 2002, Washington, DC, USA.
7. L. Chen https://www.trustedcomputinggroup.org/news/presentations/051012_DAA-slides.pdf
8. J. Camensich http://www.zisc.ethz.ch/events/ISC2004Slides/folien-jan-camenisch.pdf
9. Oded Goldreich: Zero-knowledge 20 years after its inverntion http://www.wisdom.weizmann.ac.il/~oded/PS/zk-tut02v4.ps Foundations of Cryptography (Fragments of a Book, published by Cambridge University Press) http://theory.lcs.mit.edu/~oded/frag.html

AIKs as TP identities

- The endorsement key is not used to identify a trusted platform.
- A platform may have multiple TP identities, where a TPM identity or TP identity is synonymous with an Attestation Identity Key (**AIK**). Such an identity must be:
  - Statistically unique;
  - Difficult to forge; and
  - Verifiable to a local or remote entity.
- An TP identity/attestation identity key:
  - Guarantees that certain properties hold for the platform associated with the identity:
- A TPM uses attestation identities when proving that it is a genuine TPM (conformant to TCG specifications), without identifying a particular TPM.
- Allows linkage of behaviour to previous usage of a platform using that same identity.
- Privacy Certification Authority (Privacy-CA; P-CA) attests that an ID/AIK belongs to a TP.

AI credential



No link to platform credential or other individually identifying information!

AIKs

- An AIK is a RSA 2048-bit key pair.
- The private key from an AIK pair can be used to digitally sign data generated by the TPM (exclusively):
  - PCR information;
  - Non-migratable keys generated by the TPM.
- The public key from this AIK pair is used to verify digital signatures generated by the TPM.
- A TPM may have an unlimited number of AIKs.
- As AIK generation is time-consuming, ver. 1.2 of the spec contains an (optional) pre-fetch mechanism

AIK generation

Stage 1

- A TPM_MakeIdentity command is called on the TPM:
  - The properties of the key pair to be generated are specified as input.
  - The digest of the ID Label chosen by the TPM owner and an identifier for the PCA chosen to certify the new identity is also input.
- The TPM generates an attestation identity key pair.
- The TPM creates an identity-binding:
  - Takes:
    - The public key from the newly generated AIK.
    - The digest of the ID Label chosen by the TPM owner and the identifier for the PCA chosen by the owner to attest to the new identity.
  - Computes a digital signature (generated using the private AIK) over the above data.

AIK generation

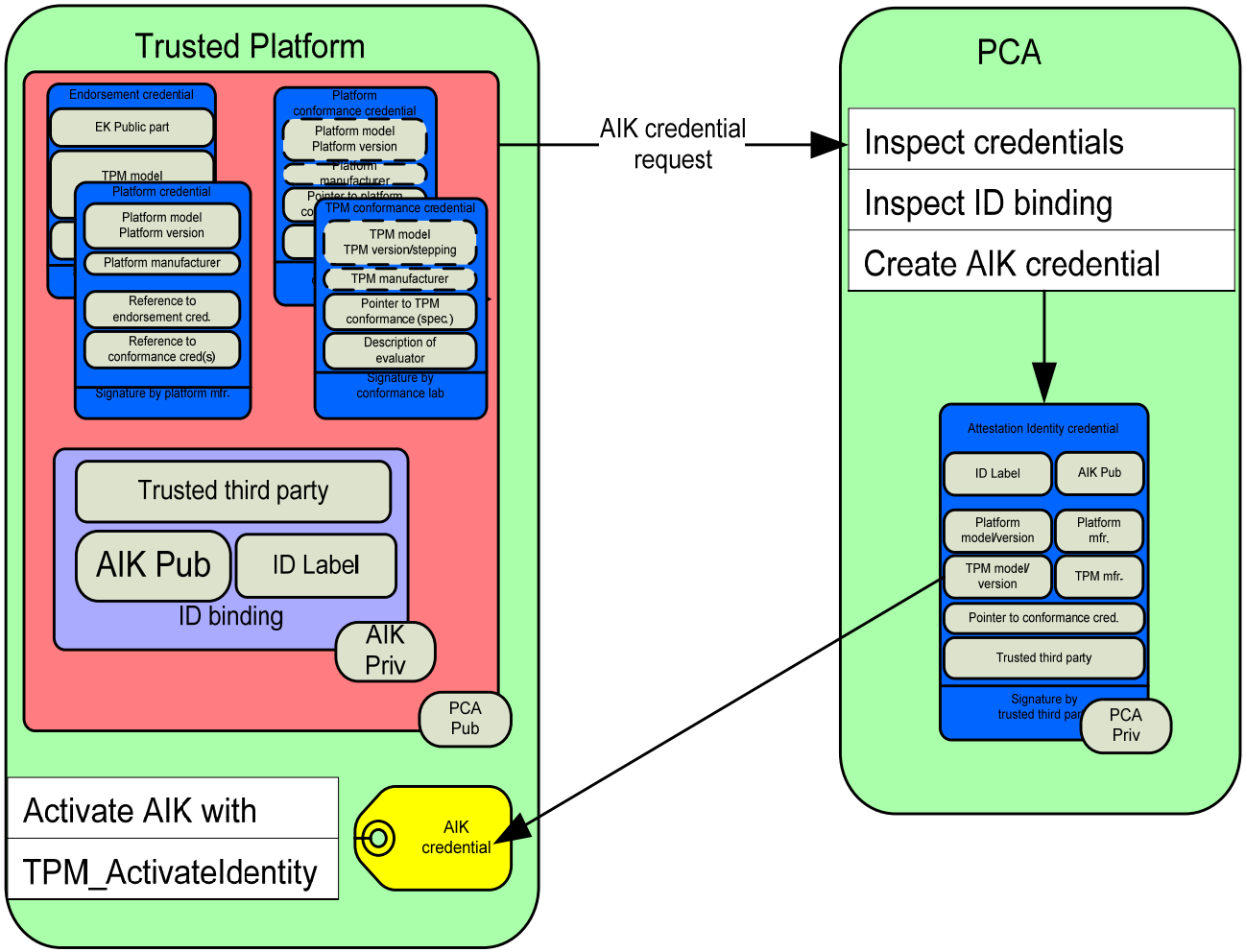
Stage 2:

- The TSS_CollateIdentityRequest is called in order to assemble all data needed by a Privacy-CA. This includes:
  - The data linked to the identity-binding (i.e. the public key from the newly generated AIK, the ID label, and the identifier for the P-CA);
  - The identity-binding;
  - The TP credential set – the endorsement credential, the platform credential and any conformance credentials.

Stage 3:

- The data described above is encrypted under the public key of the chosen P-CA and sent to the P-CA.

Requesting an AIK credential



AIK credential generation

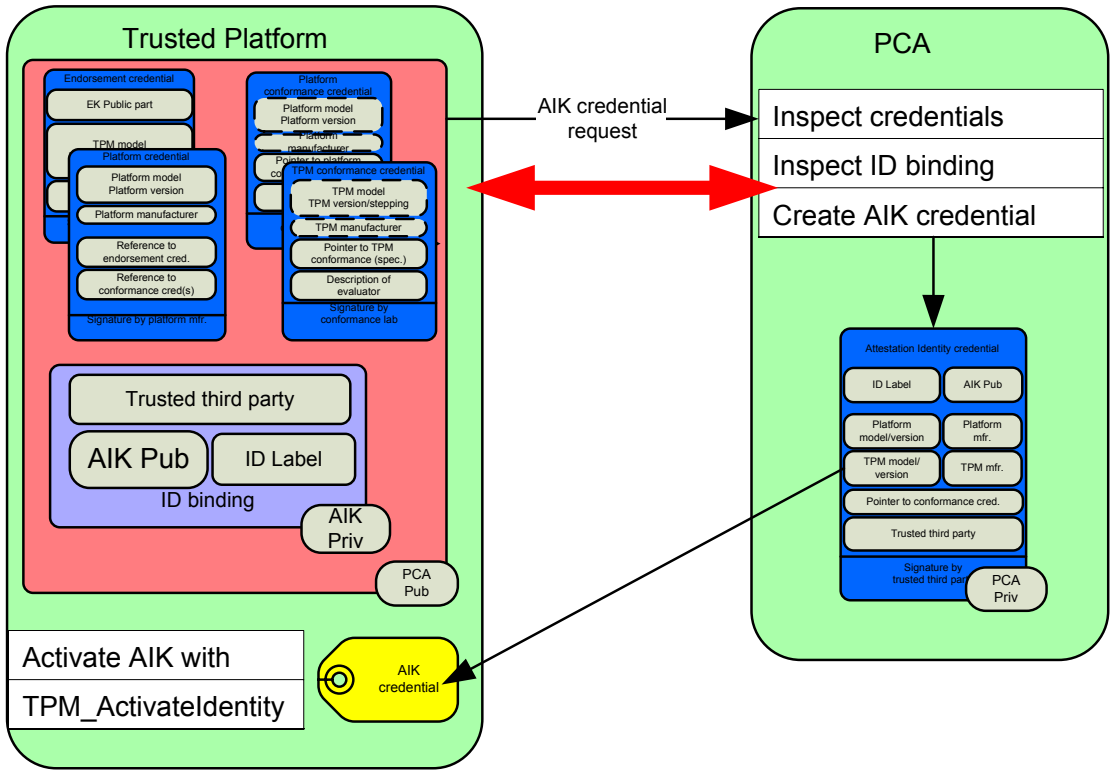
Stage 4:

- The PCA decrypts the message received.
- The PCA inspects the credentials and deduces whether the platform described in them is a genuine TP.
- The PCA inspects the identity-binding and verifies that it is consistent with the supplied information:
  - It ensures that the message was destined for it (and not another PCA);
  - It verifies the signature using the public AIK supplied.
- The PCA has no way of knowing, however, that the public AIK belongs to the genuine TPM described in the credentials.
- So, the PCA then:
  - Generates an attestation identity credential;
  - Encrypts it using a symmetric key;
  - Encrypts the symmetric key such that it can only be decrypted by the legitimate TPM.
  - The PCA also sends an encrypted hash of AIK Pub from the signed identity-binding such that it can only be decrypted by a legitimate TPM.
  - Encryption completed using the public EK in the credentials received.

Filling the gap

An intermediate handshake would allow the PCA to assure itself of the claimed identity

Could be based on a challenge/response,
For instance sending a Nonce+AIK digest encrypted with
EK Pub

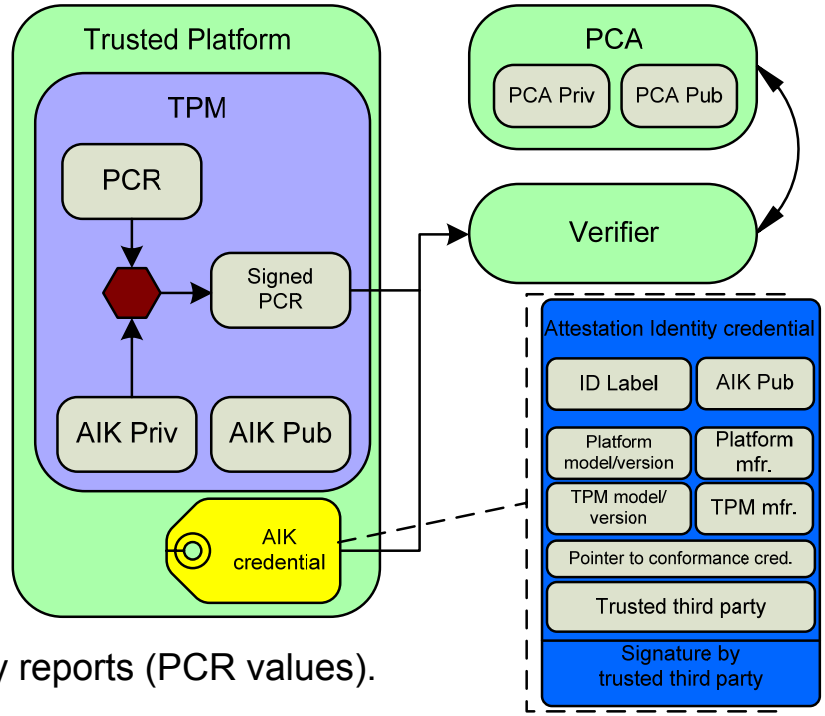


Activating the Identity for usage

- **Stage 5:**
- The TPM decrypts the data (excluding the encrypted
- attestation identity credential).
- If the data was intended for the TPM – the decrypted data contains a hash of a public key belonging to an attestation identity key pair of the TPM.
- If a match is found – the TPM releases the PCA symmetric key to the host platform.
- This is all accomplished using the TPM_ActivateIdentity command.
- **Stage 6:**
- Only if stage 4 has been successful will the attestation identity credential be decrypted using the symmetric key generated and supplied by the PCA.

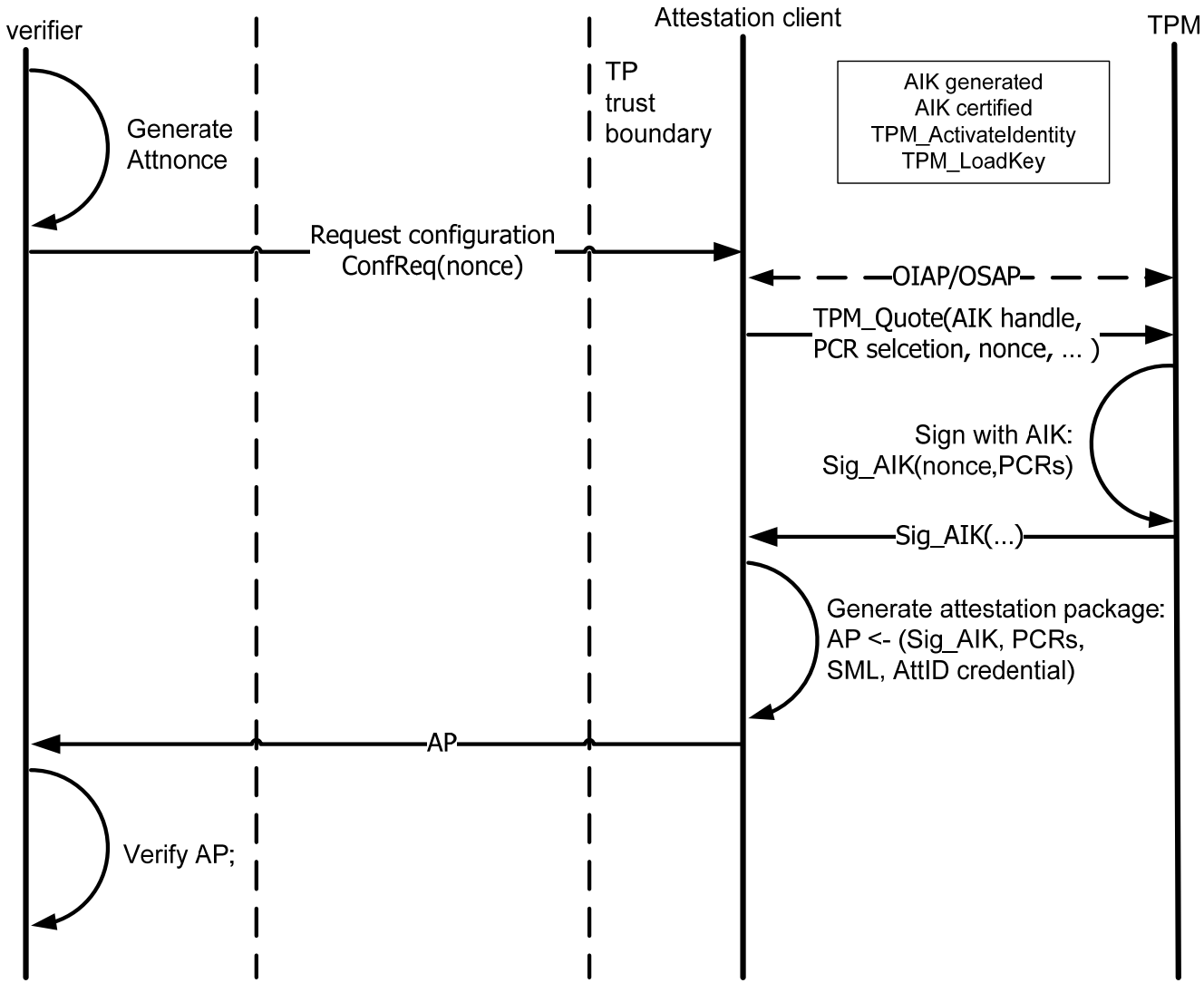
AIK credential usage - attestation

- Attestation allows a requesting party to verify
  - It is a valid TP I'm talking to
  - The specific state of the platform



- Attestation identity keys are used to sign integrity reports (PCR values).
- The recipient can then evaluate:
  - How trustworthy **the information is** using the signature of the platform on the measurements and the platform identity certificate.
  - How trustworthy **the software configuration** of the platform is using the reported measurements.
- A validation entity (VE) certifies integrity measurements, i.e. measured values and measurement digests, which correspond to correctly functioning or trustworthy platform components, for example embedded data or program code, in the form of validation certificates.
- In order to evaluate how trustworthy **the software configuration** of the platform is, the reported measurements are compared against the expected integrity metrics retrieved from certificates generated by VEs.

Attestation protocol



Pitfalls and problems

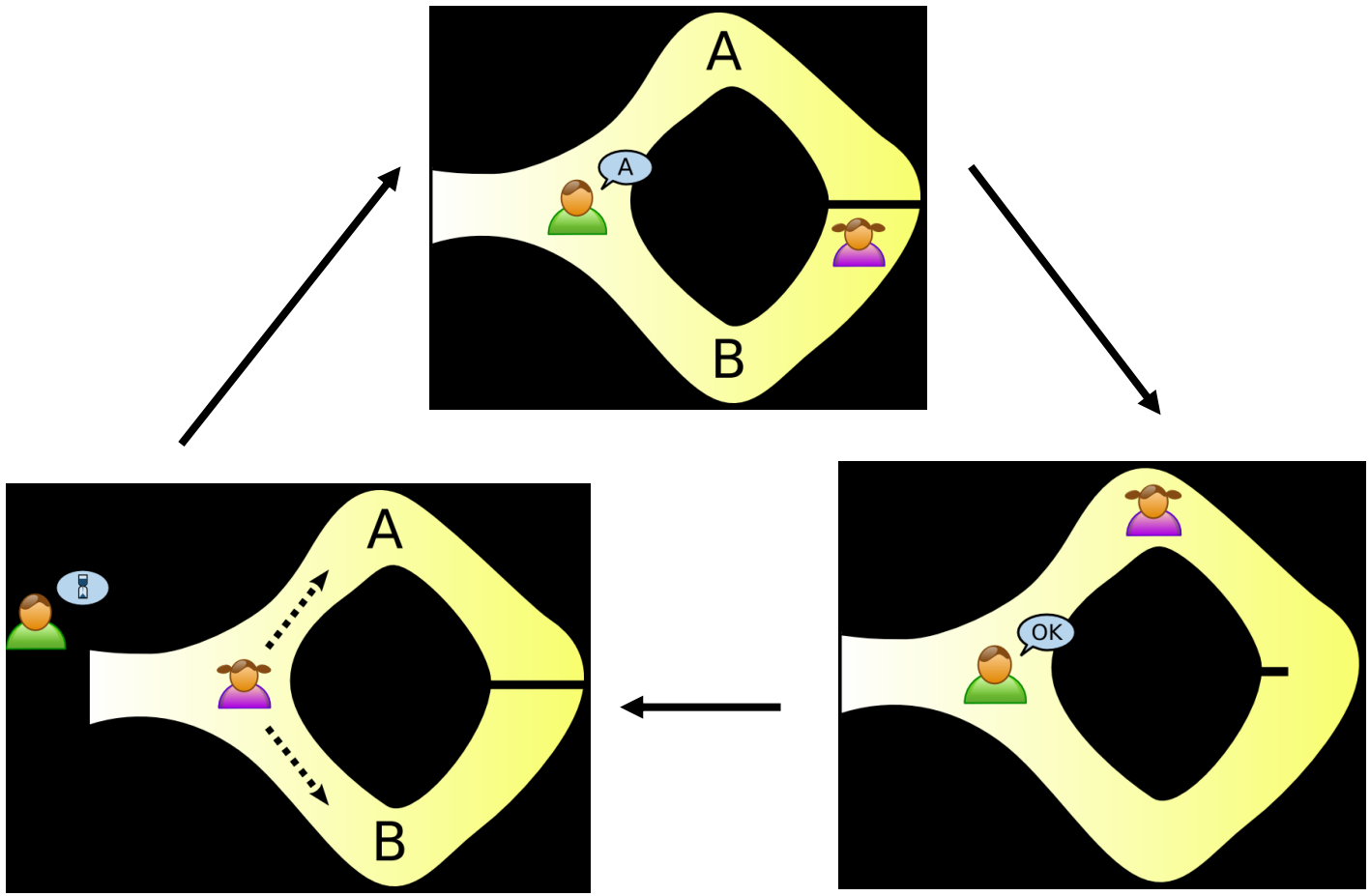
- Missing protocol step in AI credential generation
- Complexity/scalability
  - Complex SML – hard and costly to verify
  - Possible Trade-off between completeness and performance
  - Complex, open systems require large verification database
  - Load-time vs. Run-time problem – OS should be trustworthy and perform run-time checks
- Privacy/PCA
  - Complex SML potentially reveals platform identity
  - PKI infrastructure necessary
  - Point of weakness as they are capable of:
    - User/TPM activity tracking; and/or of
    - Making unwanted disclosures of platform information.
  - Lack of business model
- CA issues credentials
  - Validation, Conformance, Platform, Attestation Identification
  - Important role
    - If they fail, security features can be circumvented
    - E.g. if malfunctioning HW/SW is certified
    - E.g. if they are paid to certify a component
  - Not an easy job
    - Complexity of today's HW/SW
    - Insufficient ability to validate source code

Direct Anonymous Attestation - DAA

- Added in ver. 1.2 of the spec
- DAA removes the necessity to disclose the public value of the endorsement key to a PCA
- DAA is based on a family of cryptographic techniques known as *zero knowledge proofs*.
- DAA allows a TPM to convince a remote 'verifier' that it is indeed valid without the disclosure of the TPM public endorsement key, thereby removing the threat of a TTP collating data which may jeopardize the privacy of the TPM use.
- DAA uses a group signature scheme and ensures that the issuer is not able to identify the signer of a DAA signature.
  - Unforgeability of DAA-certificates relies on Strong RSA Assumption
  - Unlinkability of DAA-certificates/signatures relies on DDH Assumption
- Issuer and verifier are able to detect "broken TPMs" (Rogue Tagging)
- DAA involves heavy (and complex) computations and not all TPMs implement it, but it can be outsourced to the TSS.
- Does not remove privacy issues associated with a complex SML.

Commonly explained by the cave story

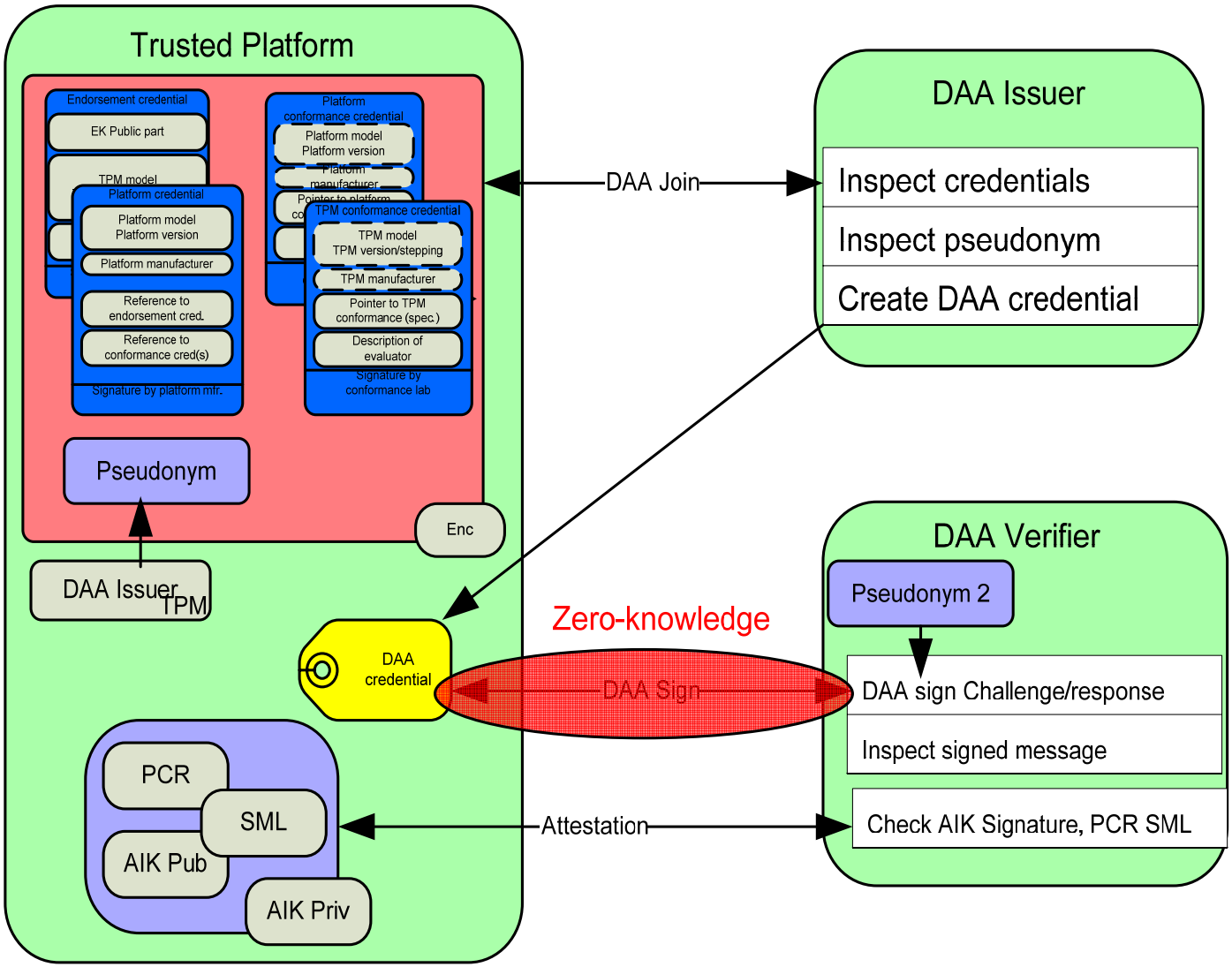
[J.J. Quisquater and L. Guillou, How to explain zero-knowledge protocols to your children, *Advances in Cryptology - Crypto '89*, Springer-Verlag (1990), 628-631.]



DAA Building Blocks

- The DAA scheme is composed of:
  - Actors: TP, DAA Issuer, DAA Verifier (challenger);
  - Protocols: DAA-Join, DAA-Sign.
- DAA-Join:
  - The platform proves to the DAA Issuer that it has a TPM-controlled, non-migratable secret *f* (generated from a seed and DAA Issuer information) by providing a *pseudonym* of the form N_1^f (N_1^f is derived from the DAA Issuer's name and is an element of a suitable group).
  - The DAA Issuer then provides to the platform a credential in the form of a Camenisch-Lysyanskaya signature on *f*.
- DAA-join is performed only once and is semantically similar to enrolment with an Identity provider
- DAA-Sign:
  - The platform can sign a message using the DAA credential and a pseudonym N_2^f chosen by the DAA Verifier. The choice of this pseudonym determines the anonymity properties of the attestation process.

Stages of DAA (simplified)



- 1st Extension: (does not break the current TPM Specification)
  - Better privacy by combining DAA with a Privacy-CA which issues “one-time” DAA-certificates (Camenisch, ESORICS 2004)
- 2nd Extension: (breaks the current TPM Specification)
  - Ensure anonymity on untrusted TPMs (Camenisch, unpublished)

Implementations

- DAA Test Tools from IBM Implement TSS part of DAA
- Mario Strasser: Software-based TPM Emulator for Linux implement TPM DAA commands https://developer.berlios.de/projects/tpm-emulator/
- IBM‘s idemix system is a forerunner and extension of the ideas: a privacy-enhanced PKI