

ZISC Information Security Colloquium

June 15, 2004

Direct Anonymous Attestation:

Achieving Privacy in Remote Authentication

Jan Camenisch

IBM Zurich Research Laboratory

[jca@zurich.ibm.com](mailto:jca@zurich.ibm.com)

Joint work with Ernie Brickell, Intel, and Liqun Chen, HP

## Overview

- What is Direct Anonymous Attestation?
- Background: Trusted computing group & TPM
- Goal: Remote Authentication of *Secure* OS
- Direct Anonymous Attestation: How it works
- Other Applications of DAA

## What Direct Anonymous Attestation is

- Direct Anonymous Attestation
  - remotely prove that a key is held in *some* hardware device
  - strong authentication combined with privacy protection
- Trusted Computing Group Standard
- Has several applications
  - use cryptographic key to authenticate as secure OS
  - (anonymous &) secure access to networks and services
  - makes key management in companies easier

## Trusted Computing Group (TCG)

- Industry standardization body for trusted computing *building blocks*
- Successor of TCPA
- Goal of Trusted Computing
  - Protect users' information & computing environment
  - Applications are provided with HW protection of crypto keys
  - Mutual Authentication of secure platforms
  - Allow companies to securely manage their IT resources
- `www.trustedcomputinggroup.org`

## Trusted Platform Module (TPM)

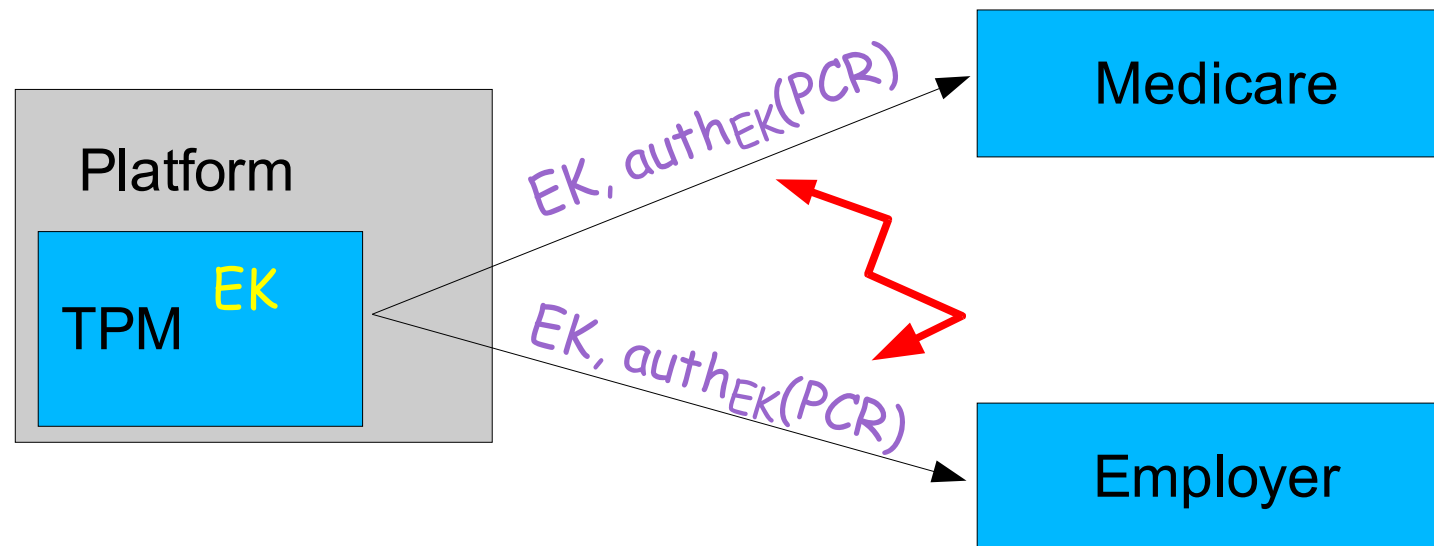
- Piece of Hardware defined by Trusted Computing Group (TCG)
- To be embedded into computing platform as root of trust
- Functionality
  - Create, use, and protect cryptographic keys
  - Sealed Storage: encrypts data such that it can only be read if platform is in same configuration
  - Measure Software on Platform, i.e., store and sign Platform Configuration Registers (PCR)
  - Attestation of PCR value to third parties, i.e., authenticate as valid TPM and then provide signatures on PCR values

## Attestation: Convincing That You Run a Secure OS



- Each TPM possess unique Endorsement Key EK (is an encryption key)
- Either verifier knows EKs of all good TPM or use certificate by CA on EK
- Verifier wants to know whether platform runs secure OS:
  - TPM could send measurements about platform (PCR values) to verifier
  - TPM needs to authenticate as a valid TPM so that verifier knows PCR is valid
  - TPM could use Endorsement Key (EK) to authenticate PCR

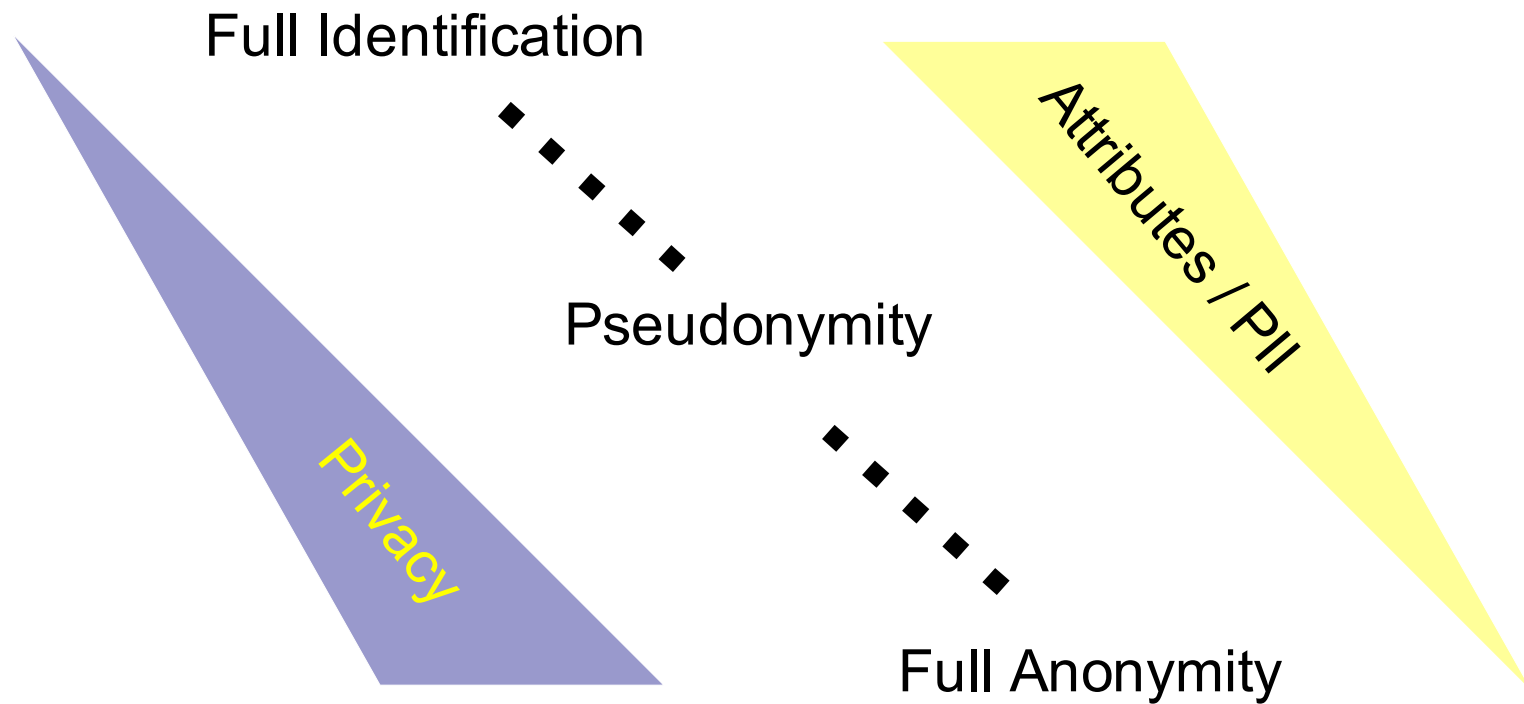
## Attestation: Convincing That You Run a Secure OS



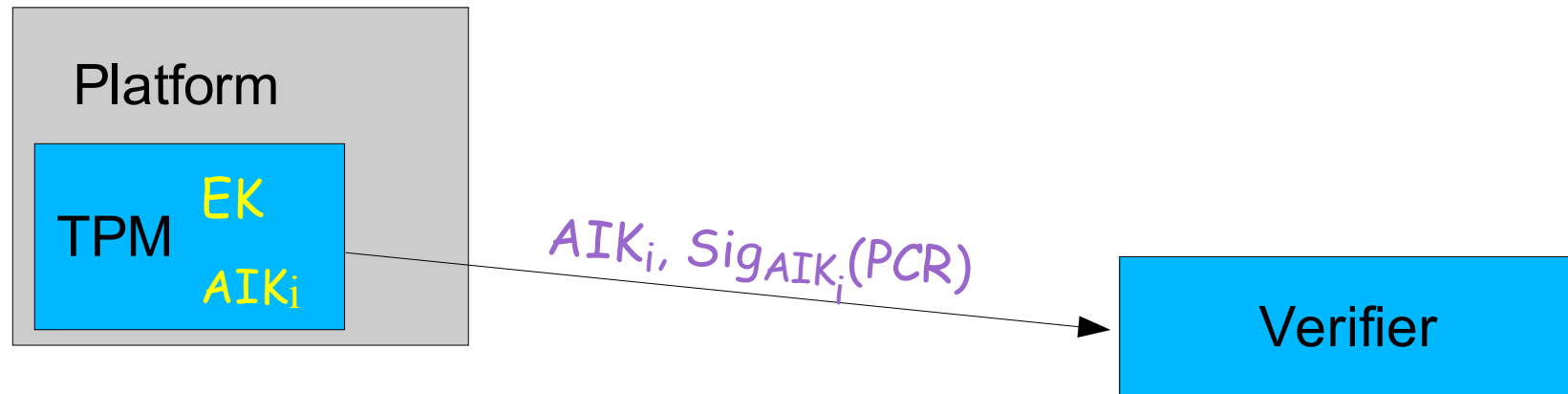
### Privacy problem:

- Two different verifier can tell that they talk to the same platform
- Actions by same users can be linked through this
- Remember Intel's Pentium III serial number

## Identification vs. Anonymity

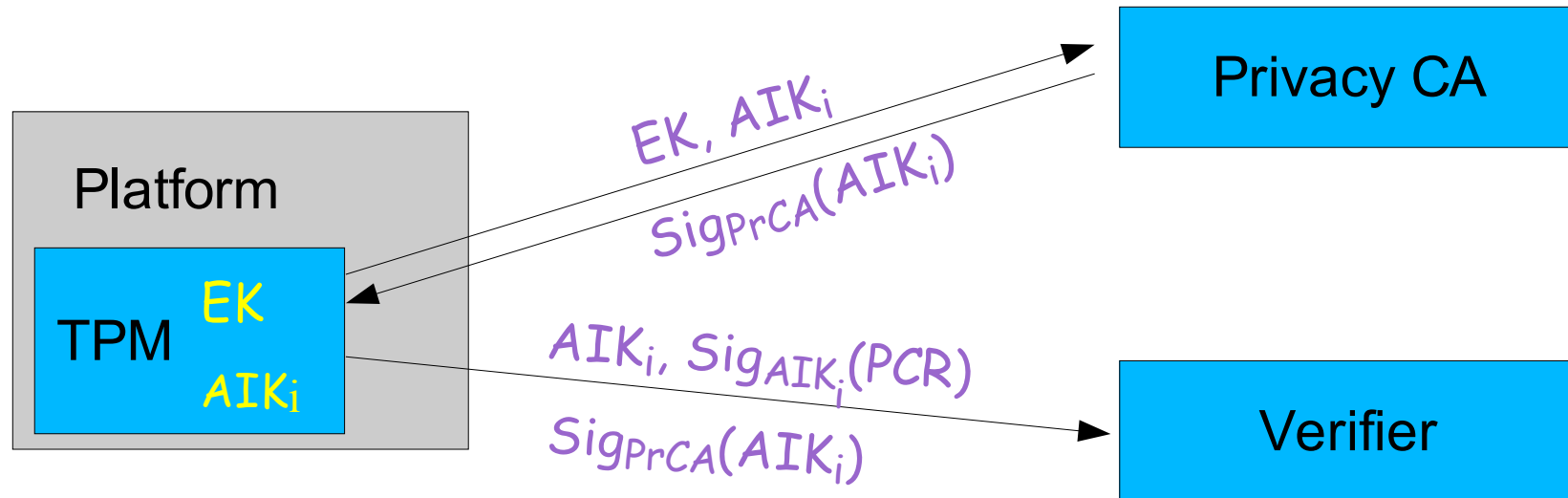


## Solution 1: the Privacy CA (TPM v1.1)



- Use / generate different keys per verifier
- Keys are called  $AIK_i$  (called Attestation Identity Key)
- $AIK_i$  is RSA signature key

## Solution 1: the Privacy CA (TPM v1.1)



- Use / generate different keys ( $AIK_i$ ) per verifier
- Keys are called  $AIK_i$  (called Attestation Identity Key)
- $AIK_i$  is an RSA signature key
- Authenticate  $AIK_i$  via Privacy CA:
  - send  $EK$  and  $AIK_i$  to Privacy CA, who checks whether  $EK$  is still good
  - obtain certificate  $SigPrCA(AIK_i)$  from Privacy CA (encrypted under  $EK$  )
  - TPM decrypts  $SigPrCA(AIK_i)$  and forwards it to verifier

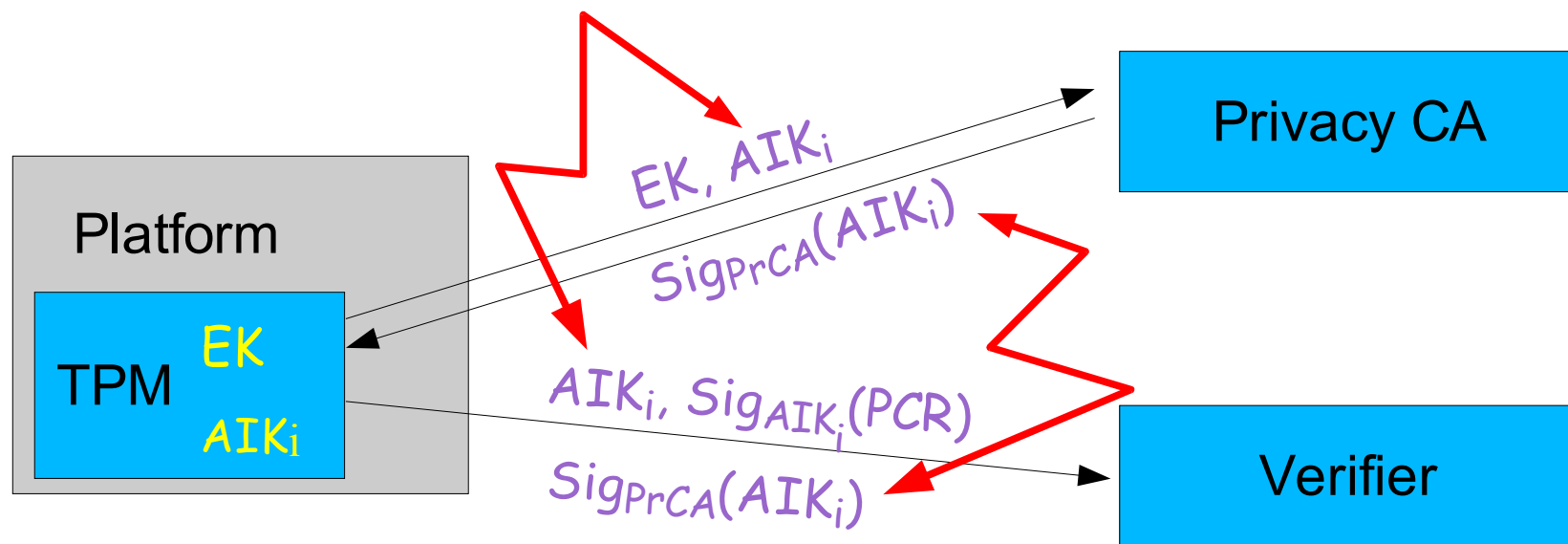
## Problem 1: the Privacy CA (TPM v1.1)

- Need to get new certificate per key: Privacy CA is a bottle neck
- Needs to be highly secured which contradicts availability
- If Privacy CA and verifier collude, they still can link
- No business model for Privacy CA

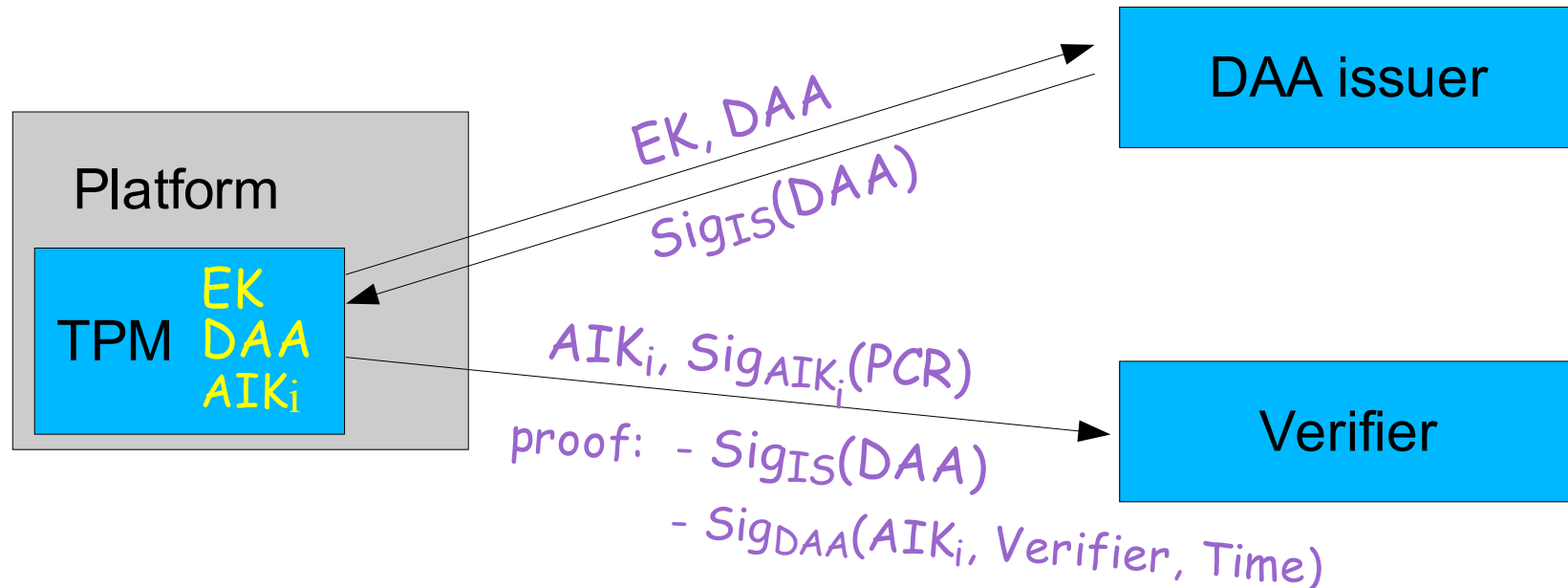
users need to trust Privacy CA not to collaborate with verifier, so Privacy CA cannot be run by Service Providers (Verifiers)

verifiers need to trust Privacy CA to only issue to valid TPM, so Privacy CA cannot be run by user/consumer organization

## Solution 1: the Privacy CA (TPM v1.1)



## Solution 2: Direct Anonymous Attestation (TPM v1.2)



Idea: do not provide certificate but use *cryptographic proof* that you have one

1. - Generate **DAA** key
  - Get signature (certificate) on **DAA** key from DAA issuer
2. - Prove that
  - a) you generated sign. by **DAA** key on **AIK<sub>i</sub>, Verifier, Time**
  - b) you possess sign. by DAA issuer on **DAA** key

## Solution 2: Direct Anonymous Attestation (TPM v1.2)

- DAA issuer and Verifier cannot link, i.e., could even be the same entity: *this solves business model problem of Privacy CA*

Certificate can  $\text{Sig}_{IS}(\text{DAA})$  could be public

- DAA certificate needs to be issued only once: *no bottleneck*
- DAA certificate can be
  - issued by manufacturer
  - by buyer of platforms (e.g., secure intranet access)

## Solution 2: Uses Camenisch-Lysyanskaya Signature Scheme

[camlys02]

Public key of DAA issuer:  $(n, a, b, d)$ , where  $n$  is an RSA modulus

Signature on message  $x$  is triple  $(c, e, s)$  such that

$$c^e = a^x b^s d \pmod n$$

For DAA :

sign public key of TPM  $DAA = a^x \pmod n$ , where  $x$  secret key of TPM

→ Scheme is provably secure under Strong RSA assumption

.... Need protocol to convince of possession of certificate on secret message

## Solution 2: Proof of Knowledge of Discrete Logarithm

[schnor91]

Prover wants to convince verifier that she *knows*  $x$  such that

$$y = a^x$$

and verifier only learns  $y$  and  $a$ :

Prover:

random  $r$

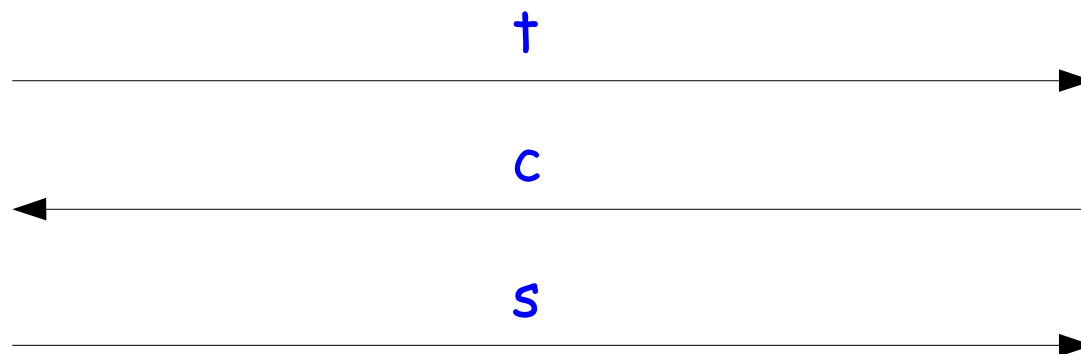
$$t = a^r$$

$$s = r - cx$$

Verifier

random  $c$

$$t = y^c a^s$$



## Solution 2: Proof of Knowledge of Discrete Logarithm

[schnor91]

Prover wants to convince verifier that she *knows*  $x_1, x_2$  such that

$$y = a^{x_1} b^{x_2}$$

and verifier only learns  $y$  and  $a, b$ :

Prover:

random  $r_1, r_2$

$$t = a^{r_1} b^{r_2}$$

$t$

Verifier

random  $c$

$c$

$$s_i = r_i - cx_i$$

$s_1, s_2$

$$t = y^c a^{s_1} b^{s_2}$$

## Solution 2: Showing DAA-Certificate

Recall:  $x$  &  $(c, e, s)$  such that  $c^e = a^x b^s d \pmod n$

† blind certificate: compute  $c' = c b^{s'}$  mod  $n$  with random  $s'$ .

$$\text{so } c'^e = a^x b^{s^*} d \pmod n$$

† send  $c'$  to verifier

† prove knowledge of  $x, e, s^*$  such that

$$d = c'^e a^{-x} b^{-s^*} \pmod n$$

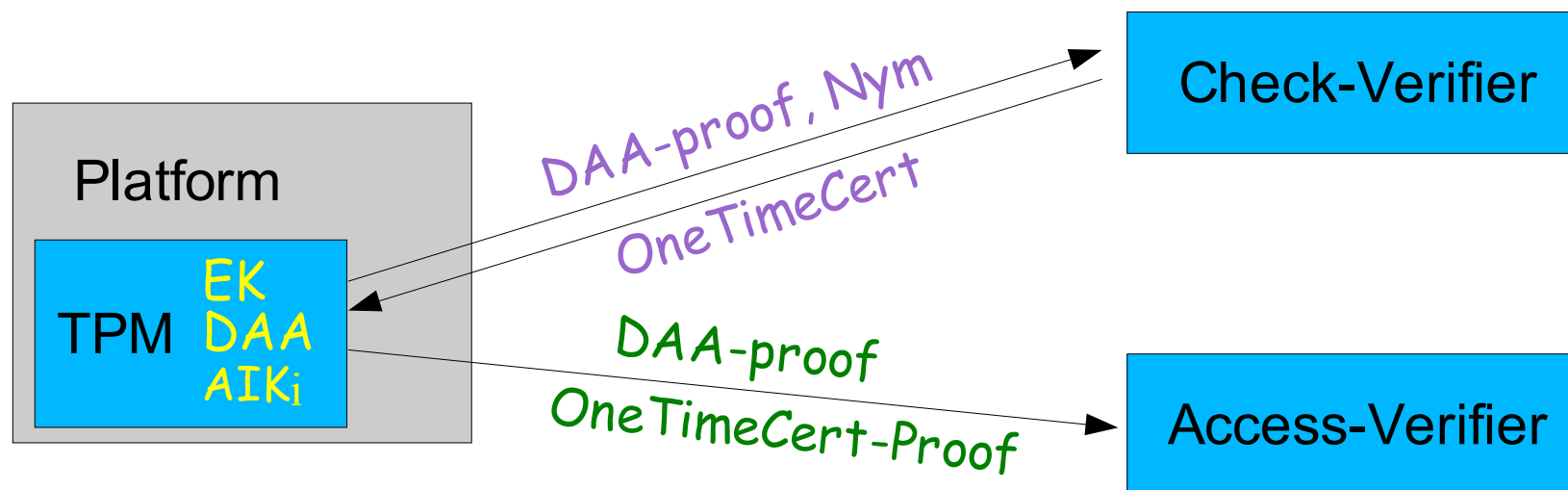
## Problem 2: Rogue TPM's

- What if DAA secret keys get extracted from a TPM?
- Verifier cannot distinguish between rogue and good TPM because of perfect anonymity!
- Verifier should be able to:
  1. detect if DAA keys found, e.g., on the Internet are used
  2. make frequency analysis on DAA keys

## Solution 3: Dealing with rogue TPM's

- TPM sends also  $Nym = f(\text{DAA-secret}) = \zeta^{\text{DAA-secret}} \bmod p$ , 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 *Named Base*): verifier can also make frequency analysis  
protocol is still pseudonymous
- **Problem 4:** Named Base solution provides less privacy because verifier can do profiling based on  $Nym$ 
  - policy on choice of  $\zeta$  is needed
  - or **Solution 4**

## Solution 4: Separating Check from Access



Idea: Separate the rogue detection from granting access

- TPM first goes to Check-Verifier who
  - uses longterm base, makes frequency & blacklist check
  - issues One-time certificate that is bound to TPM via DAA
- TPM then goes to Access-Verifier who
  - uses random base
  - grants access

## Application: Secure Log-on to Intranet

### Scenario:

- Company buys 1'000 laptops with TPM chips
- Company stores (hashes of) EKs of laptop in server which is DAA issuer
- Company distributes laptops to employees
- Employees retrieve DAA certificate from company DAA issuer (no admin involvement!)
- Company allows all platforms that have its DAA cert access to intranet

### Properties:

- Employees cannot give DAA keys to someone else
- Easy to administer: just need to add EKs to list of good platforms
- EKs are long-term (i.e., are seldom used)
- If DAA certificate expires, it can be *remotely* renewed with same security guarantees

## Application: Secure and Anonymous Access to Service

### Scenario:

- Service provider (e.g., Wallstreet journal, patent database..) sells service
- User authenticates to Service provider using DAA
- User pays & gets DAA-like certificate from service provider
- Second DAA-like certificate is bound to TPM by DAA
- Service provider allows access to all platforms that have initial DAA certificate as well as DAA-like issued by itself.

### Properties:

- Users cannot share subscriptions
- Users can access service anonymously
- Service provider is nevertheless assured that only registered users access service

## Conclusion

You can have authentication and privacy at the same time  
... was known in theory but now your computer will have it too