Privacy Preserving Protocols in e-polls

The 2007 miniWorkshop on Security Frameworks

- Privacy Protocols -

Francesco Librizzi

e-mail: librizzi@dmi.unict.it

web: http://www.dmi.unict.it/~librizzi

Overview

- Scenario;
- Introduction to Self Enforcing Privacy Protocols;
- Improvement for a Self Enforcing Privacy Protocol;
- Introduction to Trusted Computing (TC);
- DAA Protocol;
- A possible Solution: apply TC in e-voting systems.

Definitions

- " *Privacy* *is the* *right* *of discretion on personal information and private life* ";

- " *Anonymity* *is the* *right* *of discretion on personal identity* ";

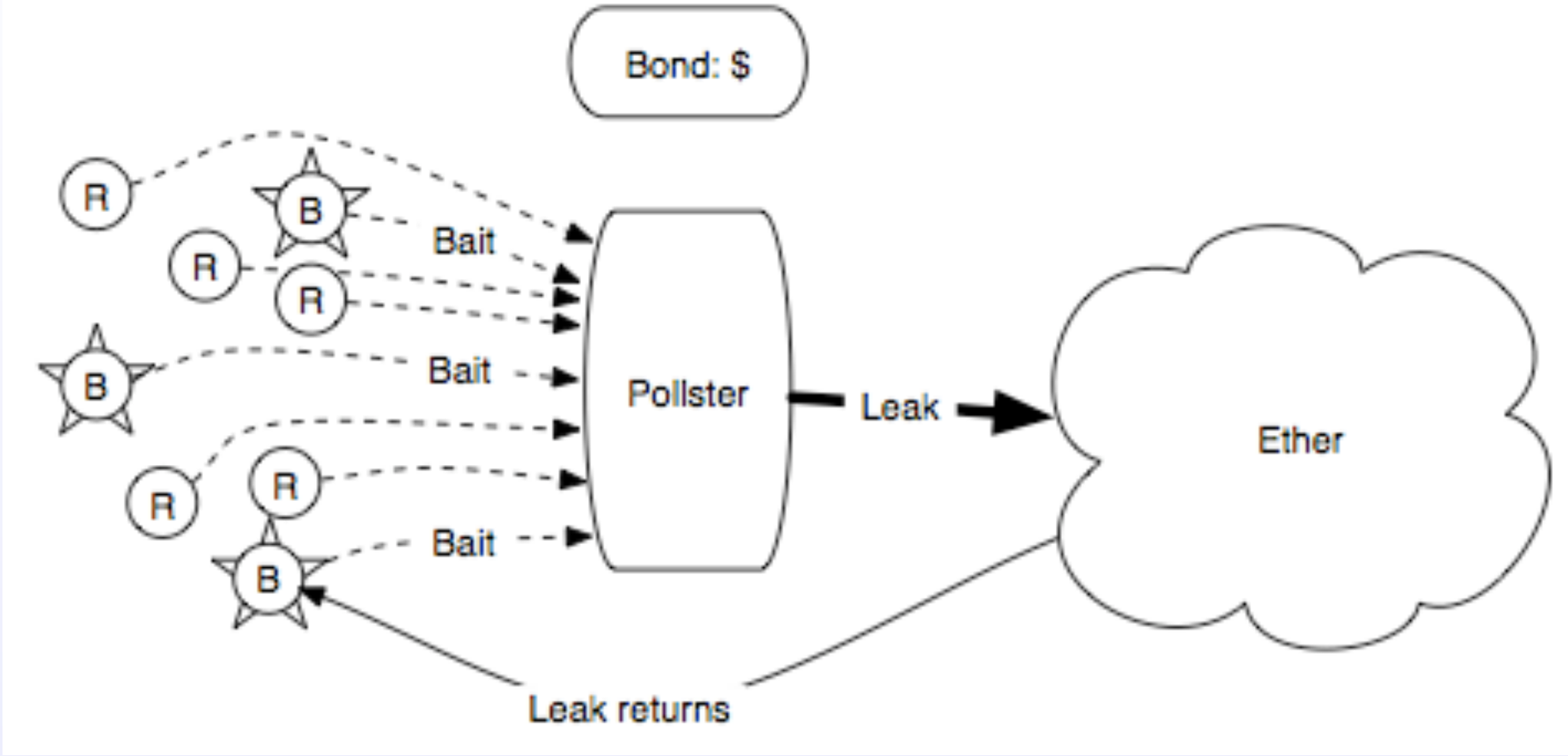
- " *A* *poll* *is a search and a process to know what people think about a topic* ".

Scenario

- *Actors*:
  - Pollster : collects data and publishes the poll results;
  - Individuals (aka *respondents*): submit data to a pollster;

- *Problems*:
  - Pollster could publish the collect data → *privacy breach*;
  - Respondents could indict the pollster for privacy breach, but *actually he behaves correctly*.

Self Enforcing Privacy Model



- *Pollster* publishes a bounty. It is used to create the baits;
- *Respondents* submit information and baits to the pollster.

Self Enforcing Privacy Protocol

Cryptographic elements:

- RSA system;
- two one-way hash functions:
  - h: $Range(h) \in \{0,1\}^l$
  - f: $Range(f) \in C$, where $C = E(x)$ and $x \in \{0,1\}^l$
- It consists of 3 steps.

The protocol (Setup step)

- *Setup*:

  1. ***P*** ⟶ ***All:***

     - $N = pq$;

     - $y \neq x^e \bmod N$;

     - *f, h: hash functions;*

The protocol (Send step)

Respondents can send two types of data:

- *Real bit*:

  2. ***I*** → ***P:*** $(I, E(r))$

     where:

     r is random and s.t. $least[\, h(I||r)\,] = b$

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

- *Bait bit:*

  2. ***I*** → ***P:*** $(I, f(s))$

     where: s is random

The protocol (Decryption step)

Decryption:

3. **P:** $D(C) = z$

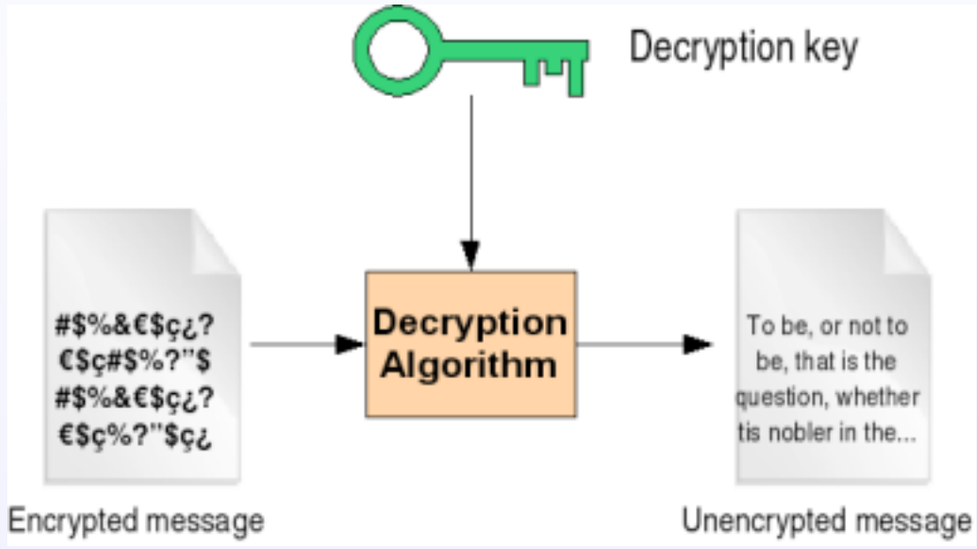
   where C is:

   - $C = E(r)$

     or

   - $C = f(s)$

and $z = r$ or $z = t$



The indictment

- *Individuals* can indict the *pollster* if $n > n_o$ valid distinct *exhibits* exist.

$<I,s,b>$ is an *exhibits*;

- An *exhibits* is valid *if and only if* :

$$least(h (I \,||\, D(f(s)))) = b;$$

- The *pollster* can contest the indictment by demonstrating that $(1/2 - w_n)n$ of alleged baits are invalid:

$$r = D (f(s)) \text{ and } least(h (I \,||\, r)) \neq b$$

Problem!

- *Respondents* ask to the *pollster* to process:

$$D(\,f(s)\,)$$

Problem!

- *Respondents* ask to the *pollster* to process:

$$D(\, f(s)\,)$$

The *pollster* is an active part in the indictment.

Problem!

- *Respondents* ask to the *pollster* to process:

$$D(\,f(s)\,)$$

The *pollster* is an active part in the indictment.

- But, if the *pollster* doesn't attend to the indictment, how does it work?
  - *It doesn't work!*

Our Solution

- We use a *PKI* and *Digital Signatures*.

- Idea:
  - Each actor must own a Key pair;
  - Each communication between *Individuals* and *Pollster* must be signed;
  - The individuals wait for an *ack* message.

The "new" protocol

- *Setup*:

  1. **P** → **All:** N, y, f, h $Sign_P(N, y, f, h)$

- *Sending Data*:

  2. *Real bit:* **I** → **P:** $(I,E(r))$ $Sign_I(I,E(r))$

  2. *Bait bit:* **I** → **P:** $(I,f(s))$ $Sign_I(I,f(s))$

  3. **P** → **I:** $ack, I, C, P, Sign_P(ack, I, C, P)$

- *Decryption:*

  4. **P:** $D(C) = z$ where $C = E(r)$ or $C = f(s)$
     and $z = r$ or $z = t$

Where's the Improvement ?

- We obtain a fairly behavior;

- In fact:
  - If the pollster commits a *privacy breach*, and he doesn't participate to the indictment
  - Then individuals can indict him, showing the pollster signs obtained during the protocol execution.

- This phase can be done without *a pollster interaction*.

Trusted Computing

- It arises from necessity to ensure physical level security.

- *Pearson* defines:

  "a trusted platform is a computing platform that has a trusted component, probably in the form of built-in hardware, which it uses to create a foundation of trust for software processes."

- The built-in hardware is the *TPM* (*Trust Platform Module*)

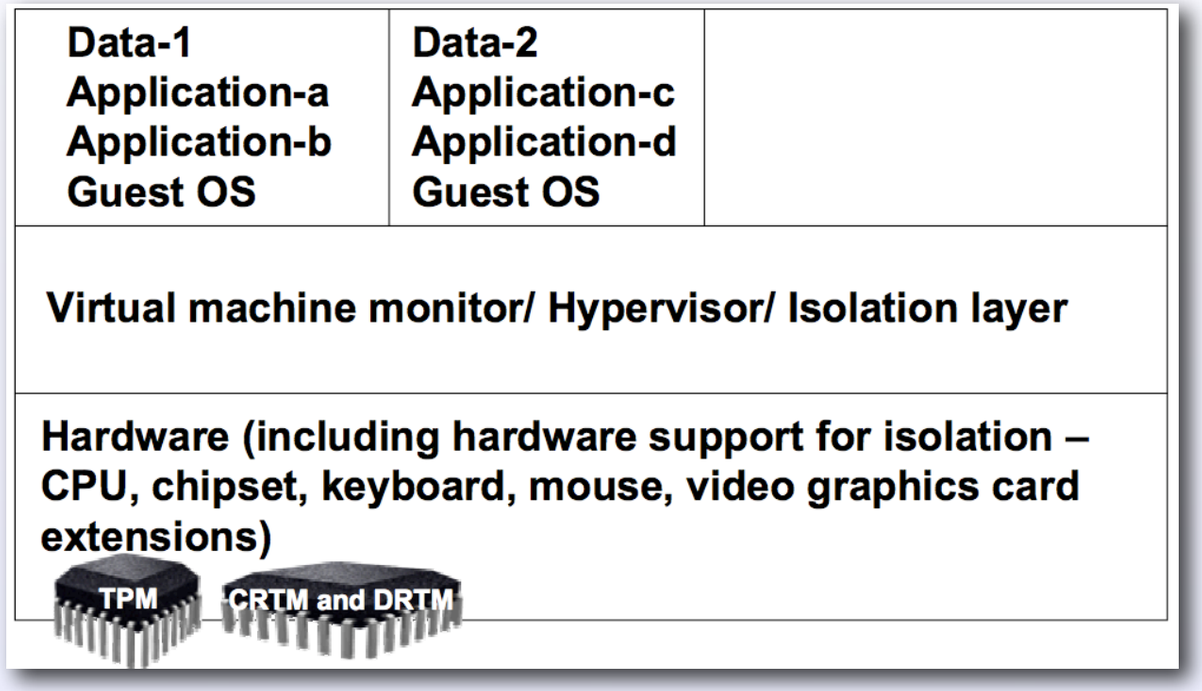
History of the TC

- *1999*: HP, Compaq, IBM and Microsoft formed the *TCPA* (*Trusted Computing Platform Alliance*) working group; TPM v1.1
- *2003*: *TCPA* was superseded by the ***TCG*** (*Trusted Computing Group*) that released the TPM v1.2 specifications.

- There are some different proposal:
  - M$: Palladium ⇒ NGSCB
  - Intel: LaGrande
  - AMD: Presidio
  - ...

What does TPM offer?

- TPM offers 5 main functions:
  - Integrity Measurement;
  - Authenticated Boot;
  - Sealing;
  - Attestation;
  - SW isolation.



- Another function is the Secure Boot!

Attestation

- *Credentials*:
  - *EK* (*Endorsement Key pair*), unique and embedded;
  - *AIK* (*Attestation Identity Key pair*), TPM generated;

- *Problem*:

  if the TPM *uses* the EK pair, then everybody could *track* its activities, *breaching its privacy*.

- *Solution*:

  Attestation, a TTP identifies the TPM and signs the AIK credentials. (*TPM v1.1*)

The DAA protocol

- *Direct Anonymous Attestation* is due to E.Brickell (Intel), J.Camenisch (IBM) and L.Chen (HP). (in TPM v.1.2)

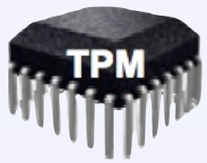
- Separates the authentication from the credential issue;

- Two Phases: *Join*, *DAA-Sign;*

- Uses several Cryptographic techniques: *Interactive Proofs (IP), Group Signature Schemes, ...*

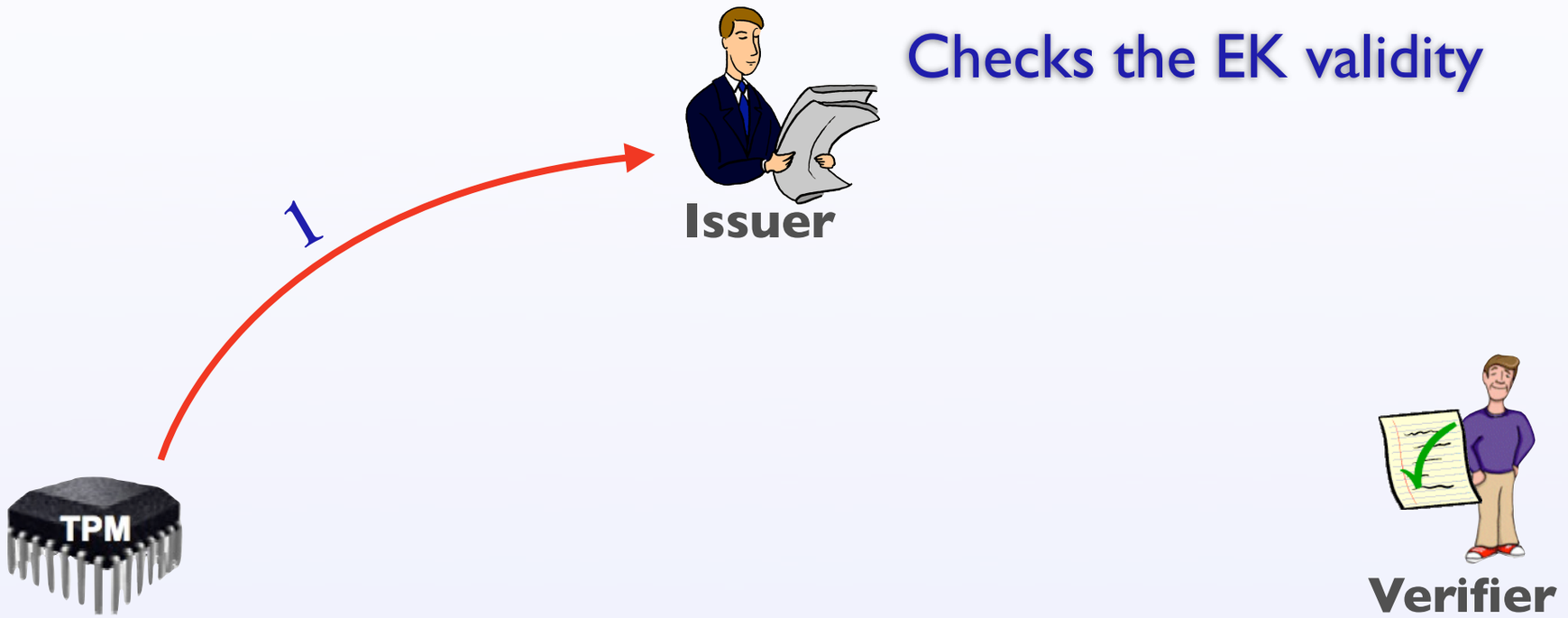
DAA (Join)

Issuer



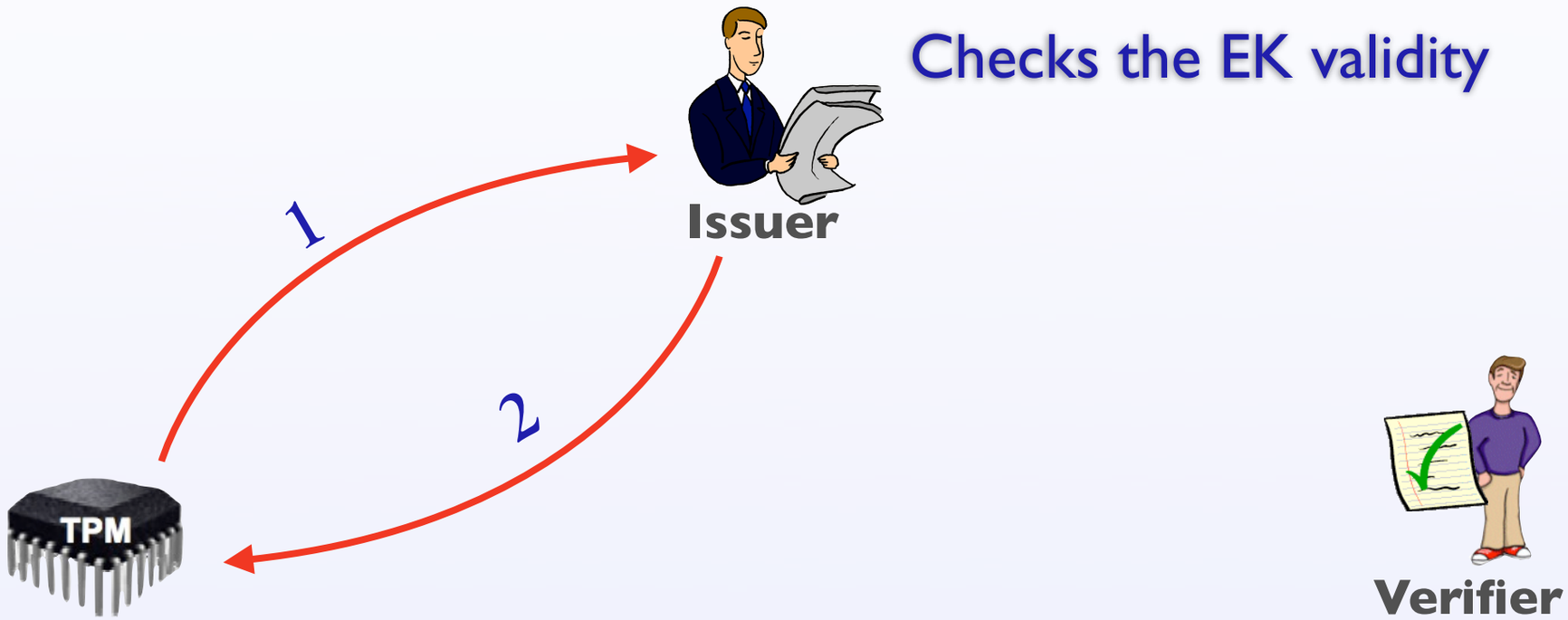
Verifier

DAA (Join)



1. TPM → Issuer : *EK public Key*

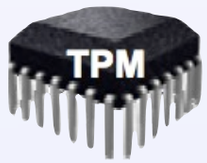
DAA (Join)



1. TPM → Issuer : *EK public Key*
2. Issuer → TPM : *Attestation (anonymous)*

DAA (Sign)

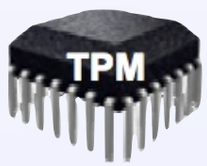
Issuer



Verifier

DAA (Sign)

Issuer



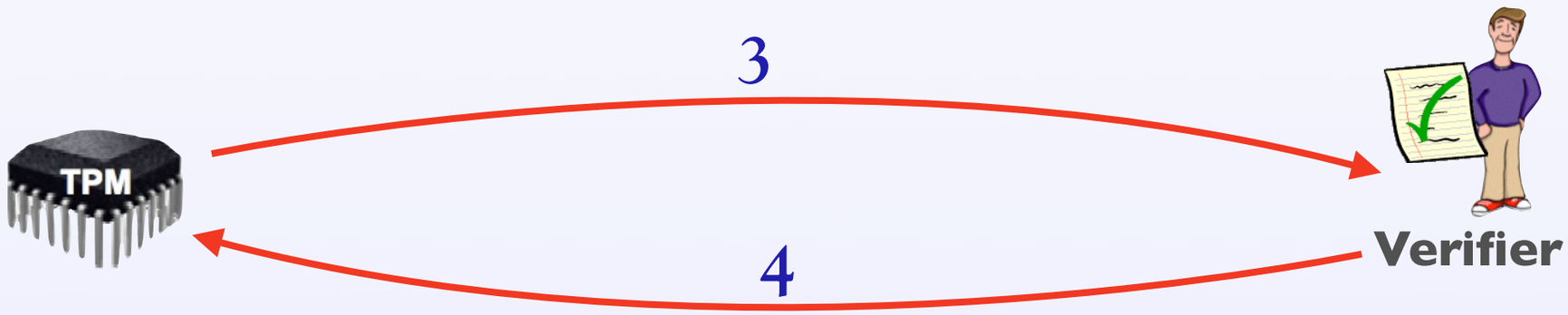
3

Verifier

3. TPM → Verifier : *AIK public Key, Attestation Cert. , Pseudonym, IP(Attestation, Pseudonym)*

DAA (Sign)

Issuer



3. TPM → Verifier : *AIK public Key, Attestation Cert. , Pseudonym, IP(Attestation, Pseudonym)*

4. Verifier → TPM : *Sign(AIK)*

Problem!

- *Pseudonymous* is calculated as follows:

$$\boldsymbol{N_v = \zeta^w}$$

where:

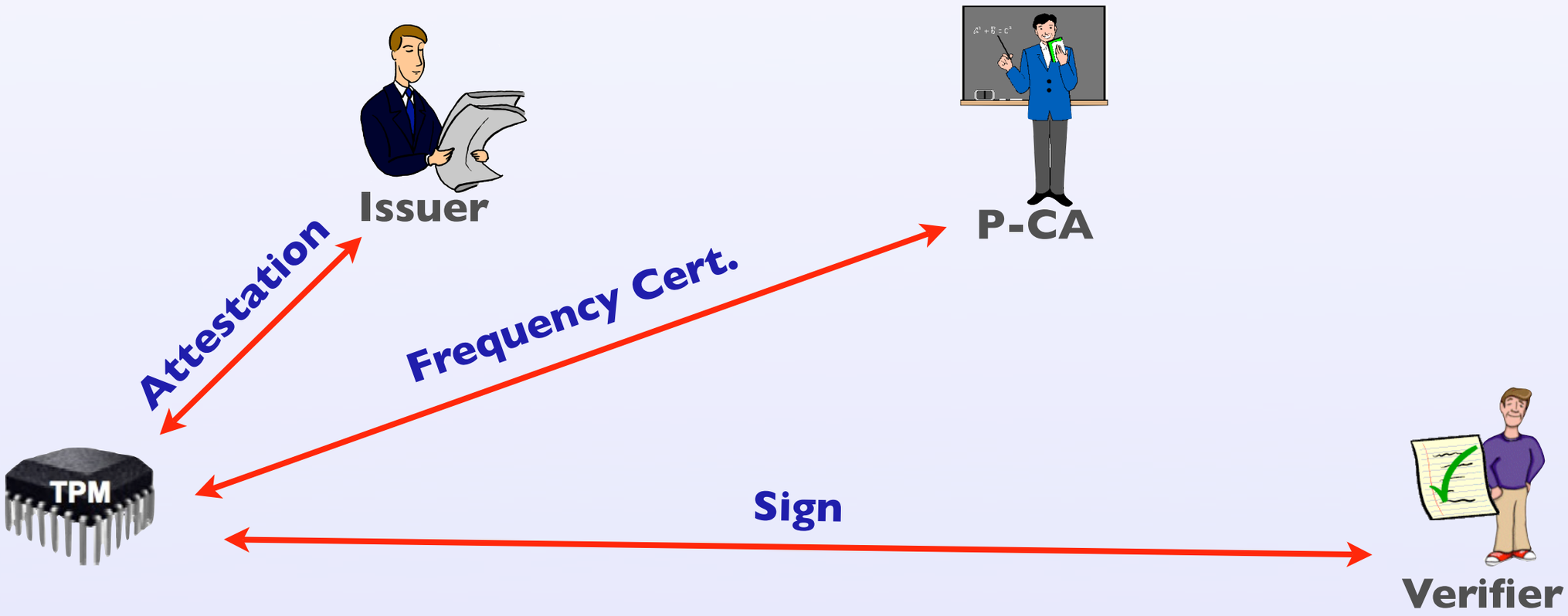
- ζ is the *Verifier name hash*;
- and $\boldsymbol{w}$ is a *secret value* generated during the *Join phase*.

- If the TPM executes only once the *Join phase* and if it always contacts the same *Verifier*, then:

we can track the TPM activities → *breach privacy*

Modified DAA

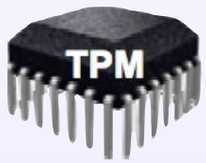
- New entity P-CA (Privacy - Certificate Authority);
- New certificate: frequency certificate;
- New schema:



Modified DAA (Attestation)

Issuer

P-CA



Verifier

Modified DAA (Attestation)

Checks the EK validity

Issuer

P-CA

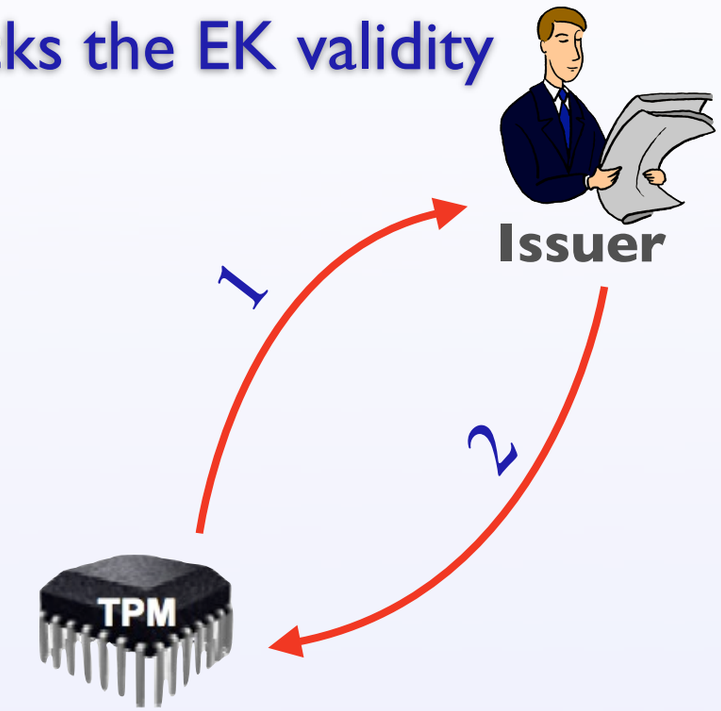
1

Verifier

1. TPM → Issuer : *EK public Key*

Modified DAA (Attestation)

Checks the EK validity

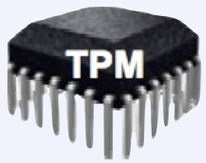


1. TPM → Issuer : *EK public Key*
2. Issuer → TPM : *Attestation (anonymous),* K_t

Modified DAA (Freq. Cert.)

Issuer

P-CA



Verifier

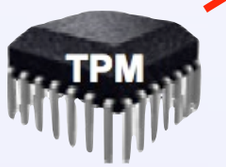
Modified DAA (Freq. Cert.)

Issuer

P-CA

Checks the Attest. Cert. and the pseud.

3



Verifier

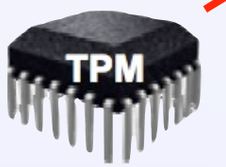
3. TPM → P-CA : *Attestation Cert., Pseudonymous*

Modified DAA (Freq. Cert.)

Issuer

P-CA

Checks the Attest. Cert. and the pseud.



3

4

Verifier

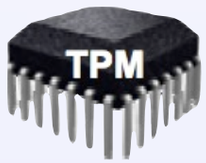
3. TPM → P-CA : *Attestation Cert., Pseudonymous*

4. P-CA → TPM : *Frequency Certificate*

Modified DAA (Sign)

Issuer

P-CA

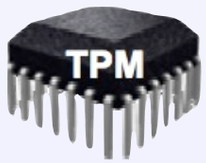


Verifier

Modified DAA (Sign)

Issuer

P-CA



5

Verifier

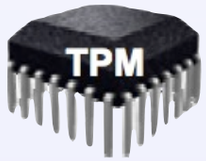
IP(Attest.,Freq.Cert.)
if OK then Sign

5. TPM → Verifier : *Attestation Cert., Freq. Cert., AIK*

Modified DAA (Sign)

Issuer

P-CA



5

6

Verifier

IP(Attest.,Freq.Cert.)
if OK then Sign

5. TPM → Verifier : *Attestation Cert., Freq. Cert., AIK*

6. Verifier → TPM : *Sign(AIK)*

TC in e-voting systems!

DAA protocol solves the problems of:

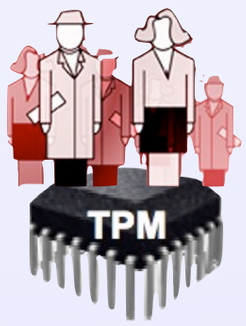
- authentication
- and anonymity;

But, in e-voting systems there are the same problems.

- Idea:

Use DAA protocol in e-voting system

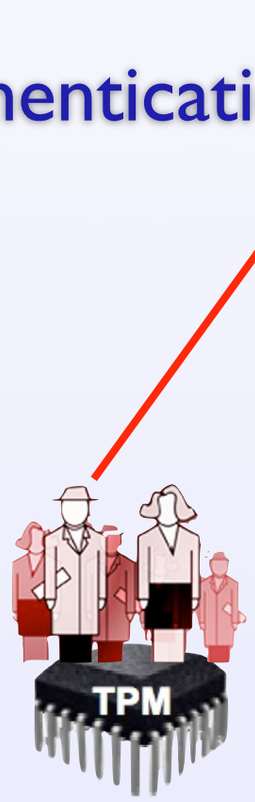
Draft Solution



Draft Solution

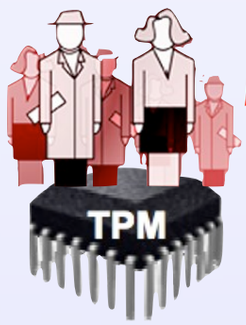
Checks the citizen qualification

Authentication



Draft Solution

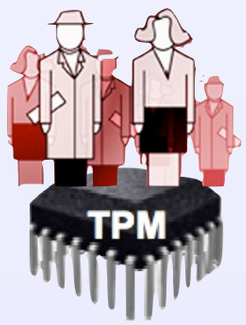
Anonymous Attestation



Draft Solution

Checks the Attestation and Pseudonymous

Vote Request



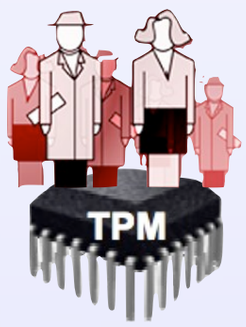
Draft Solution

Frequency Cert.

Draft Solution

Voting

Draft Solution



Ack

Conclusions

- We studied two different scenarios that have some similarities, and we think that the TC technologies could be applied in e-voting systems.

- The first Protocol introduced allows respondents to obtain both the privacy for submitted data and a fairly behaviour between respondents and pollster. Moreover with our improvement the protocol always works fine.

- The draft solution here introduced is only a sample, in fact there are several improvements to do to obtain the e-voting constraints.

Future Works

- Could DAA protocol be applied in Self Enforcing Protocol to obtain the anonymity?

- How could we assure that a vote is securely counted during the poll?

- How could we assure that a vote is counted only once?

- ...

References

[1] - P.Golle, F. McSherry and Ilya Mironov. Data Collection With Self-Enforcing Privacy. In CCS'06 October 30- November 3, 2006.

[2] - E.Brickell, J.Camenisch and L.Chen. Direct Anonymous Attestation. May 18, 2004.

[3] - Jan Camenisch. Better Privacy for Trusted Computing Platforms. IBM Research, Zurich Research Laboratory, Switzerland.

Questions?