ABC4TRUST

ABC4Trust on Smart Cards

Pascal Paillier – CryptoExperts

Summit Event, Brussels – Jan 20, 2015

Embedding Privacy-ABCs on Smart Cards

pascal.paillier@cryptoexperts.com

Smart card reference implementation

ABC4Trust Card Lite

- supports device-bound U-Prove and Idemix
- and virtually any discrete-log based pABC system :)
- is free and open source on GitHub

Version 1.2 is based on MultOS ML3 dual-interface cards with $\simeq$64 kB of non-volatile memory, $\simeq$ 1kB of RAM and SLE78 Infineon processor

Why use Smart Cards?

Without smart cards

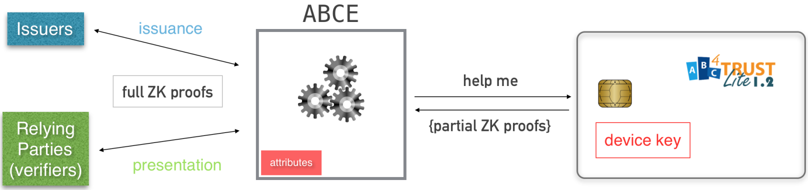
- credentials, attributes, secrets: all in one place
- untrusted computing environment (trojans, viruses, etc)
- identity theft

With smart cards

- stores/uses the user's private key securely
- trusted computing environment
  - logical security can be demonstrated
  - tamper-resistance achievable (side-channels, faults)
  - security certification (common criteria, EMVCo)

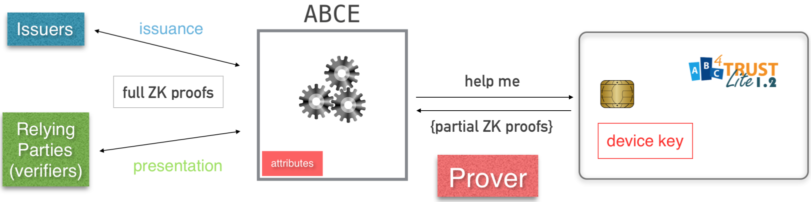
What does it do?

- contains the user's private key x
- operates crypto operations on x delegated by the ABC Engine
- + stores blobs for the ABCE, + backup-able, + PIN-protected



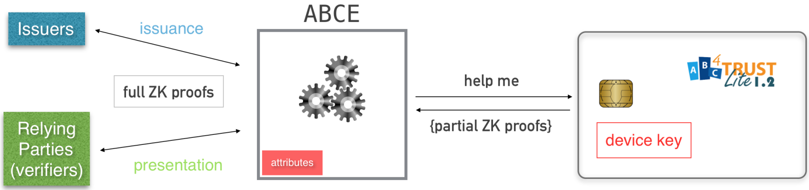
What does it do?

- contains the user's private key x
- operates crypto operations on x delegated by the ABC Engine
- $+$ stores blobs for the ABCE, $+$ backup-able, $+$ PIN-protected



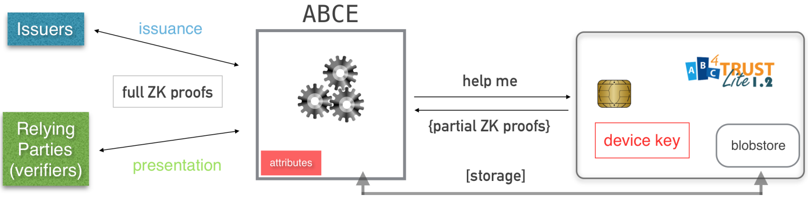
What does it do?

- contains the user's private key x
- operates crypto operations on x delegated by the ABC Engine
- + stores blobs for the ABCE, + backup-able, + PIN-protected



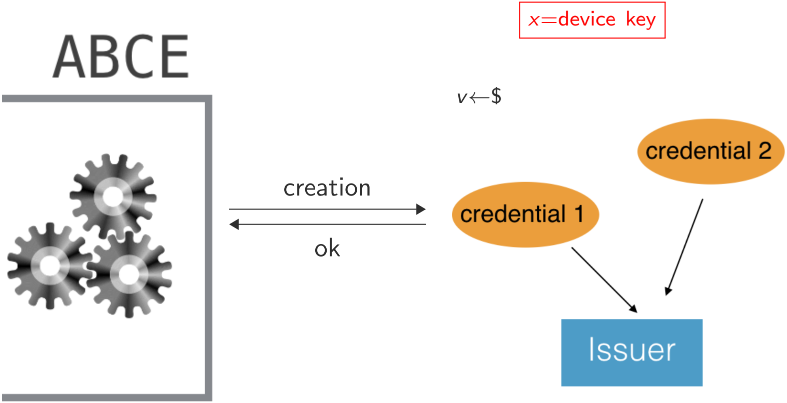
What does it do?

- contains the user's private key x
- operates crypto operations on x delegated by the ABC Engine
- $+$ stores blobs for the ABCE, $+$ backup-able, $+$ PIN-protected



How does it work? (basic operations)

The card uses an internal representation of pABC objects and roles



x=device key

ABCE

$v \leftarrow \$$

creation

ok

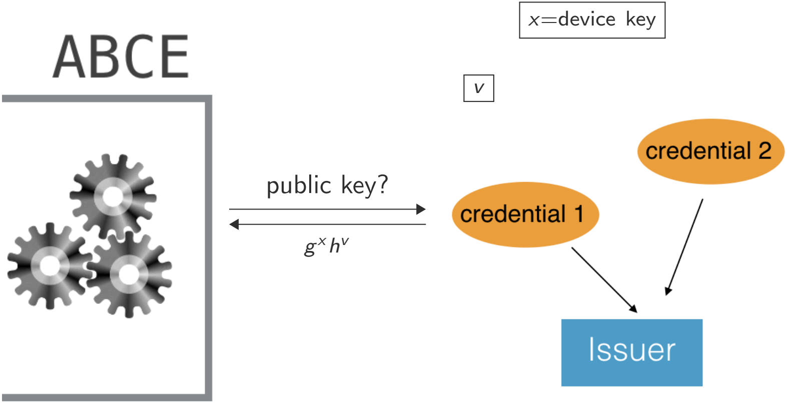
credential 1

credential 2

Issuer

How does it work? (basic operations)

The card uses an internal representation of pABC objects and roles



x=device key

v

ABCE

public key?

$g^x h^v$

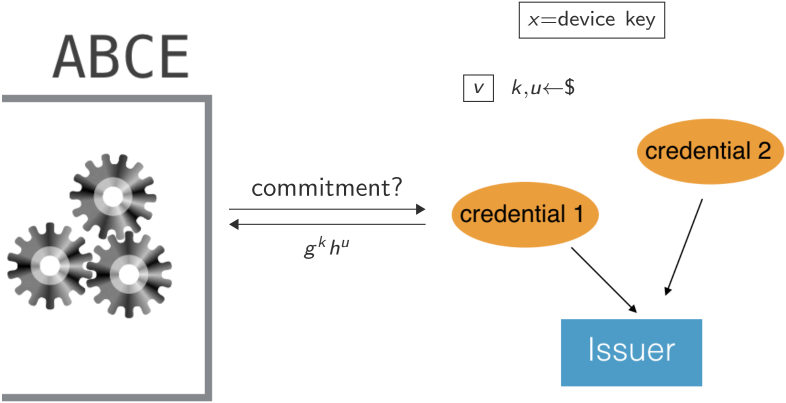
credential 1

credential 2

Issuer

How does it work? (basic operations)

The card uses an internal representation of pABC objects and roles



$\boxed{x\text{=device key}}$

ABCE

$\boxed{v}$ $k,u \leftarrow \$$

commitment?

$g^k h^u$

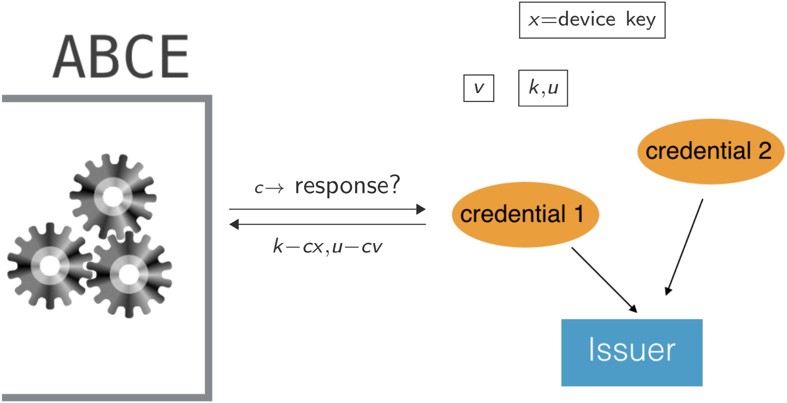
credential 2

credential 1

Issuer

How does it work? (basic operations)

The card uses an internal representation of pABC objects and roles



ABCE

x=device key

v

k,u

credential 2

$c \rightarrow$ response?

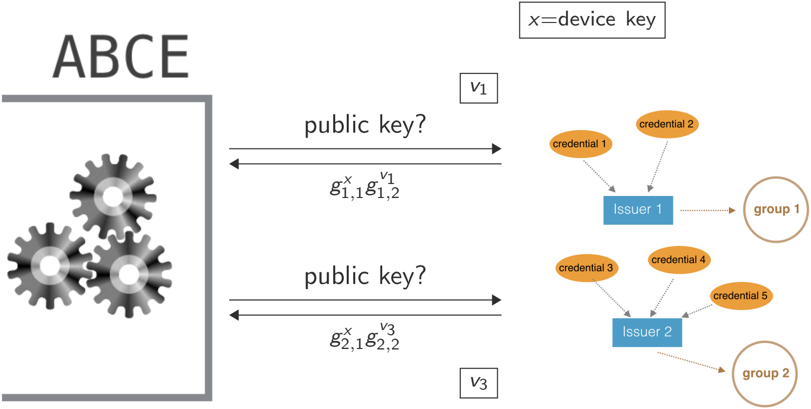
credential 1

$k-cx, u-cv$

Issuer

How does it work? (integrated sessions)

Issuers rely on their own algebraic setting of groups and generators.



x=device key

ABCE

v_1

public key?

$g_{1,1}^{x} g_{1,2}^{v_1}$

public key?

$g_{2,1}^{x} g_{2,2}^{v_3}$

v_3

credential 1

credential 2

Issuer 1

group 1

credential 3

credential 4

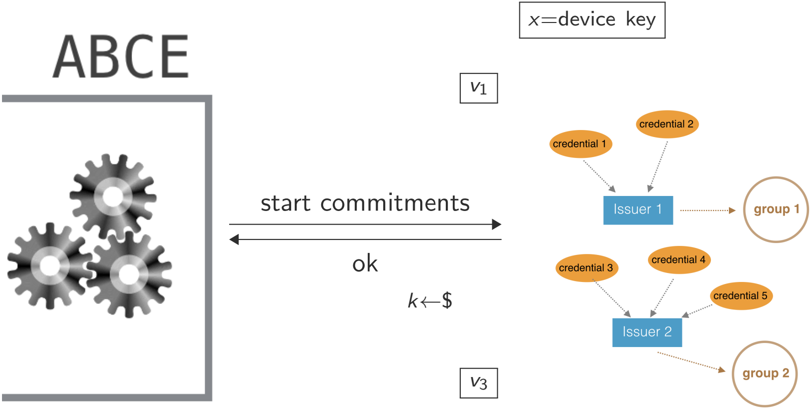
credential 5

Issuer 2

group 2

How does it work? (integrated sessions)

Issuers rely on their own algebraic setting of groups and generators.



x=device key

v_1

start commitments

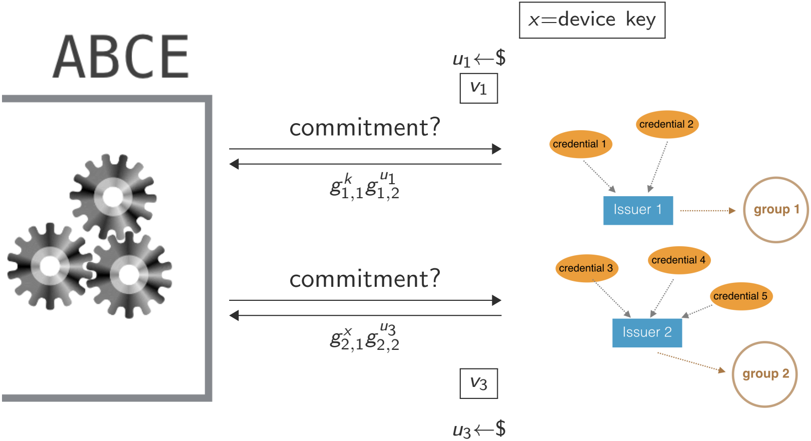
ok

$k \leftarrow \$$

v_3

How does it work? (integrated sessions)

Issuers rely on their own algebraic setting of groups and generators.



ABCE

$\boxed{x\text{=device key}}$

$u_1 \leftarrow \$$

$\boxed{v_1}$

commitment?

$g_{1,1}^{k} g_{1,2}^{u_1}$

commitment?

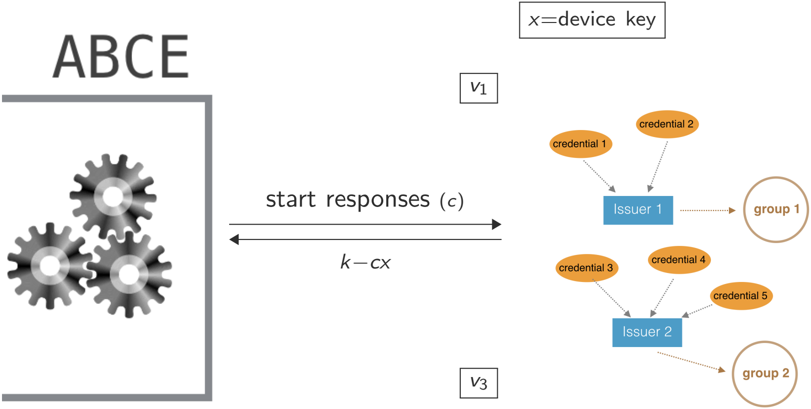
$g_{2,1}^{x} g_{2,2}^{u_3}$

$\boxed{v_3}$

$u_3 \leftarrow \$$

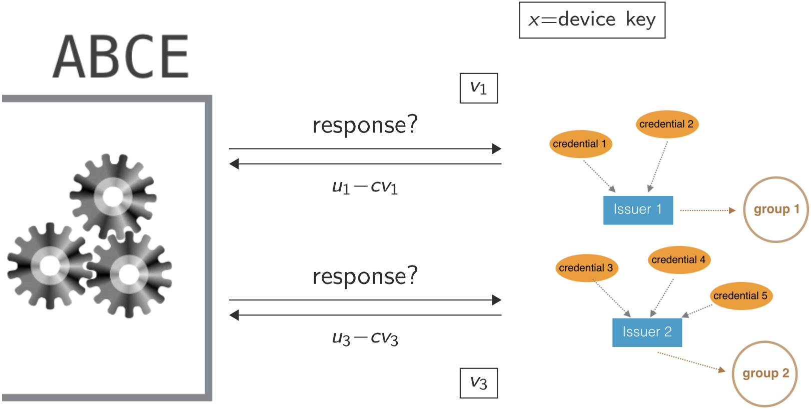
How does it work? (integrated sessions)

Issuers rely on their own algebraic setting of groups and generators.



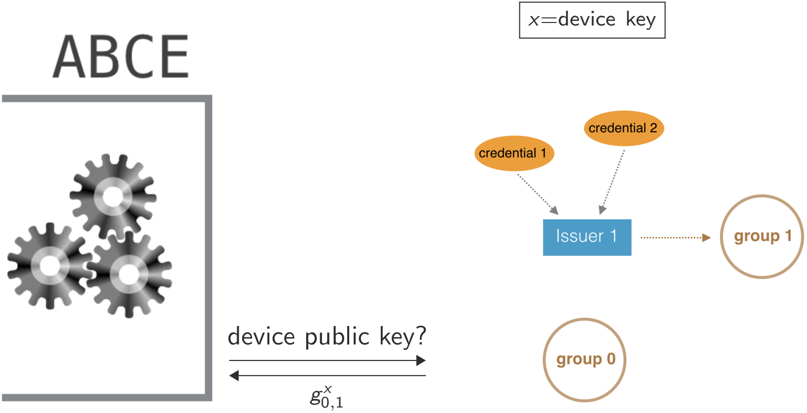
How does it work? (integrated sessions)

Issuers rely on their own algebraic setting of groups and generators.



How does it work? (pseudonyms)

There is a group 0 specifically for pseudonyms.



x=device key

ABCE

credential 1

credential 2

Issuer 1

group 1

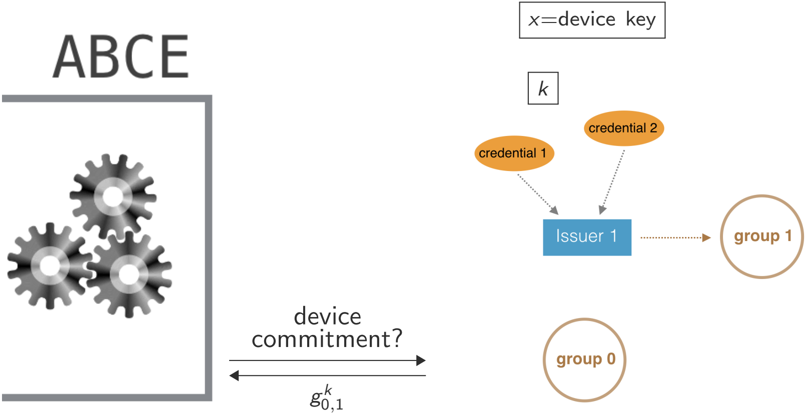
group 0

device public key?

$g_{0,1}^{x}$

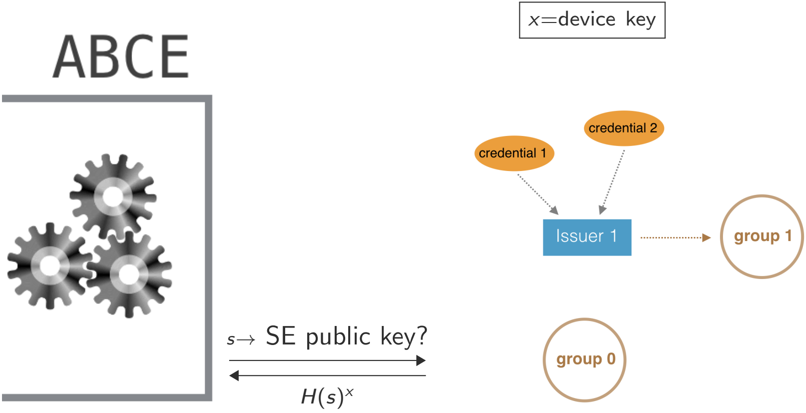
How does it work? (pseudonyms)

There is a group 0 specifically for pseudonyms.



How does it work? (SE pseudonyms)

Group 0 is also used for scope-exclusive pseudonyms.



ABCE

$s \rightarrow$ SE public key?

$H(s)^x$

x=device key

credential 1

credential 2

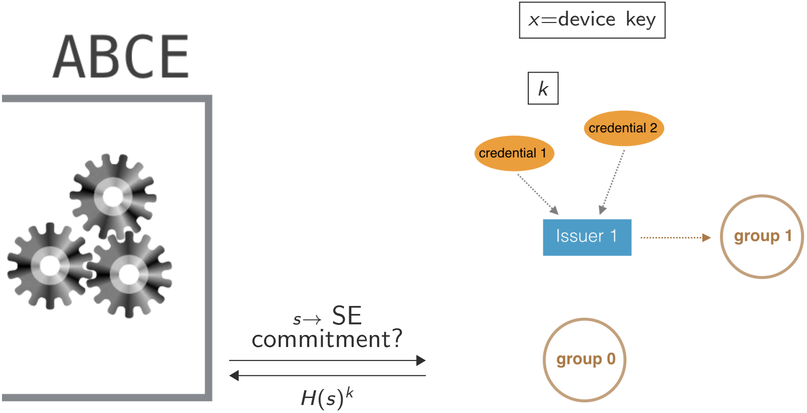
Issuer 1

group 1

group 0

How does it work? (SE pseudonyms)

Group 0 is also used for scope-exclusive pseudonyms.



ABCE

x=device key

k

credential 1

credential 2

Issuer 1

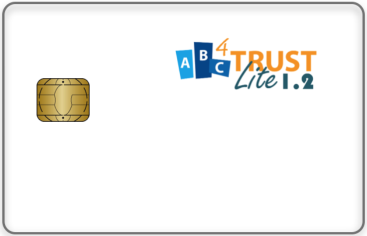
group 1

group 0

$s \rightarrow$ SE commitment?

$H(s)^k$

Conclusion



- Smart card implementation that supports device-bound privacy-ABCs in a maximally flexible way
- Protects user private key from logical and physical attacks
- Open source application based on MultOS, freely available on Github (C code)
- Fully documented and standardization-ready application