
ACTIVIDENTITY™

The Open Protocol for Access Control Identification and Ticketing with PrivacY

For Secure Contactless Transactions
and
Enabling Logical and Physical Access Convergence

Actividentity

October 2010

ACTIVIDENTITY

OPACITY

The Program

What is OPACITY… The program.

Open **P**rotocol for **A**ccess **C**ontrol **I**dentification and **T**icketing with privac**Y**

- A program to design and promote a new protocol suite to allow the removal of usage restrictions on contactless transactions
  - A protocol suite
  - A set of specifications
  - A reference implementation
  - A standardization initiative
  - A review program
  - A proof of concept
  - A statutory invention registration
  - A proof of concept
  - A product

What is OPACITY… The program.

A protocol Suite:

- generic, standard, authentication and key agreement protocols specially optimized for fast contactless transactions.

- a stable target (20/30 years) for Physical Access Control Systems (PACS).

What is OPACITY… The program.

A protocol suite:

Designed to protect information over the air

- End to end secure channel protection for digital transactions
- With security enhancements, such as forward secrecy and full privacy, sensitive or privileged information can now be exchanged over the air with assurance.

Designed for performance

- Optimized for performance. (Elliptic curve, persistent binding, single command)

Designed for easy integration

- Simple cryptographic key management
- Easy integration (one command one response)

Designed to be easily certified and last

- NSA Suite-B, NIST 800-56A compliant (FIPS 140-2 approved protocol)

What is OPACITY… The program.

- **A set of specifications,**

The specification fully describes the protocol constructs but also its standard, ISO compliant implementation at the card-edge on a User smart card and on a Secure Authentication Module (SAM) smart card.

The specification fully describes the protocol constructs but also its standard, ISO compliant implementation at the card-edge on a User smart card and on a Secure Authentication Module smart card.

The specification itself is covered by an Apache 2.0 license so that no Intellectual Property claims can be made on it and is provided royalty free to the community.

See http://www.apache.org/licenses/LICENSE-2.0.

See the current OPACITY specifications.
Opacity-protocol-specification-3-5.doc.

What is OPACITY… The program.

- **A reference implementation,**

The reference implementation provides an immediate and unambiguous guidance on interpreting the specification.

  - For the first time, and uniquely a reference implementation includes applet code,
  - For the first time, a reference implementation covers the concept of a SAM, which is a smartcard chip, issued with the same infrastructure than user card and with the same level of certification.
  - For the first time, in compliance with NIST guidance, ALL the crypto on the terminal side is implemented in the FIPS boundary of the SAM reference, which considerably facilitates and speeds up integration. Also the resulting level of Assurance and Certification of the integration is much higher.
  - The reference implementation has been successfully tested on cards from Oberthur, Gemalto, G&D, NXP, Infineon.

The reference implementation is covered by an Apache 2.0 license so that no Intellectual Property claims can be made on it and is provided royalty free to the community*. See "http://www.apache.org/licenses/LICENSE-2.0"*

What is OPACITY… The program.

- **A review program,**

The specifications and/or the reference implementation have been reviewed by personnel of:

US NIST, US Dod, BT, and members of the PACS community such as NEDAP, BORER, TDS, Hirsch/SCM…

What is OPACITY… The program.

- **A Standardization initiative:**

Various contribution were made under the program.

- ISO 24727-3 (SMAV3)
- ISO 24727-6 (the final registration of the protocol as an ISO standard is in progress and should be completed by end of August).
- ANSI-GICS part1/part2: the next revision of PIV targeted for publication in December 2010.

- **A Proof Of Concept**

Includes the complete provisioning of an OPACITY PIV 2-7 card as well as physical and logical access usage.

- **A few download sites are being identified:**
  - ActivIdentity
  - SmartCard Alliance
  - Muscle ORG.

What is OPACITY… The program.

- **A Statutory Invention Registration**

A special type of patent, which protects the protocol but excludes royalties, has been filed and protects the community using OPACITY from anyone trying to patent the protocol.

- **A product**, with supporting Actividentity applets, middleware and CMS with FIPS certification targeted for completion in December 2010 on multiple card platforms.

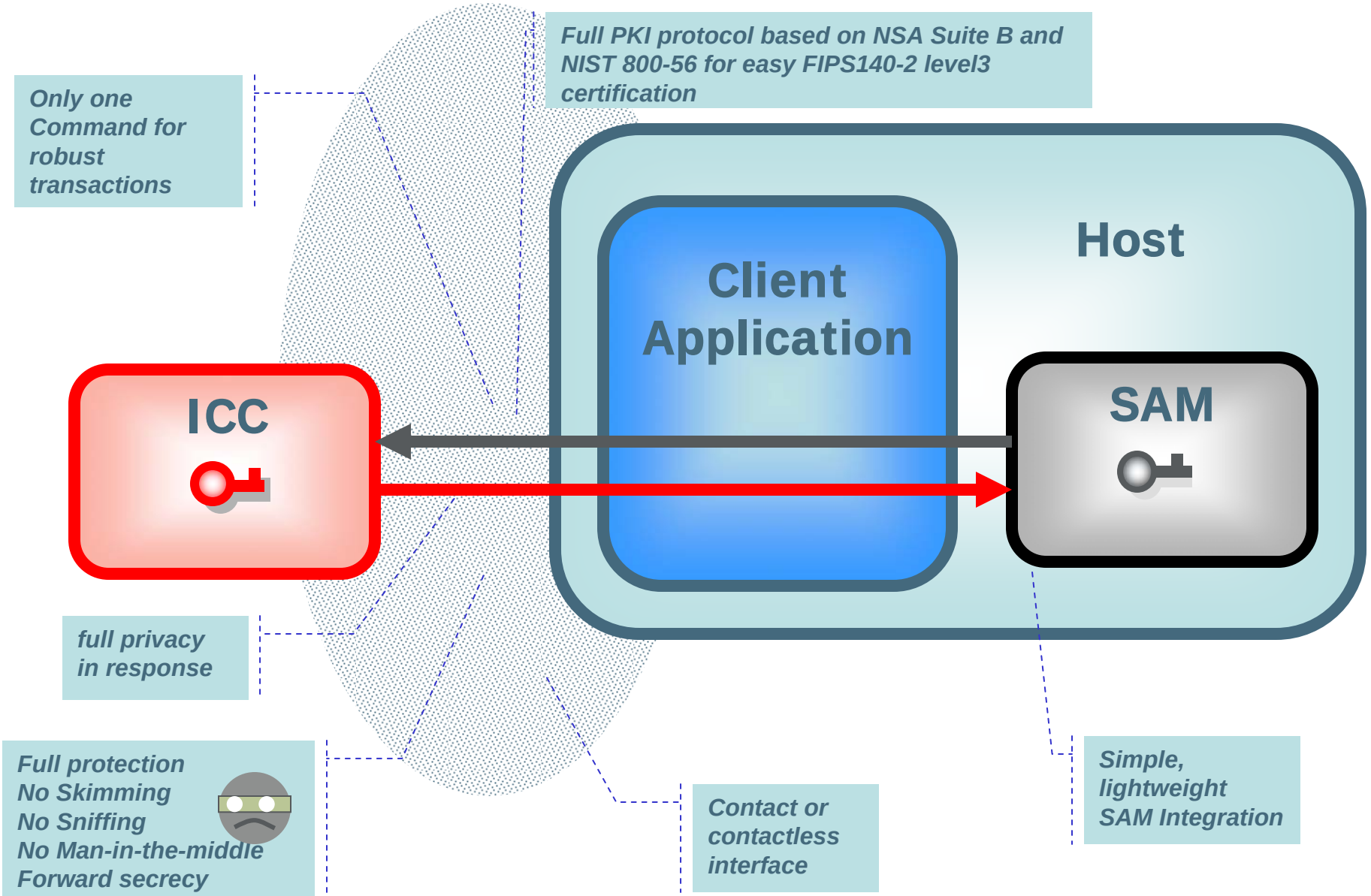
Status of the OPACITY Program

- **Specification and standardization**
  - OPACITY Protocol Specification 3.5.
  - Compliance with the NIST SP 800-56A protocol recommendations has been reviewed and green-lighted by NIST
  - Based on optimizations of NIST SP 800-56A compliant ECC-based authentication protocols contributed by Actividentity to ISO 24727-3 in 2008.
  - Written into current draft of GICS part 2.
  - First protocol registered to ISO24727-6 (SAI Global - ISO registry)
- **Publication**
  - Published to SCA as “first” industry contribution
  - http://www.smartcardalliance.org/pages/smart-cards-contributions-opacity
- **Reference implementation**
  - Current is Revision 7 with Apache license 2.0
  - Currently distributed on http://Sourceforge.net/projects/opacity
- **Proof of Concept**
  - ActivIdentity Windows7 mini-driver part of the OPACITY POC
  - Actividentity prototype PACS
  - Communicated to a few PACS vendors.

OPACITY

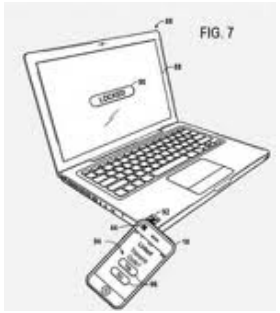
Technology and Solutions

Opacity protocol Overview

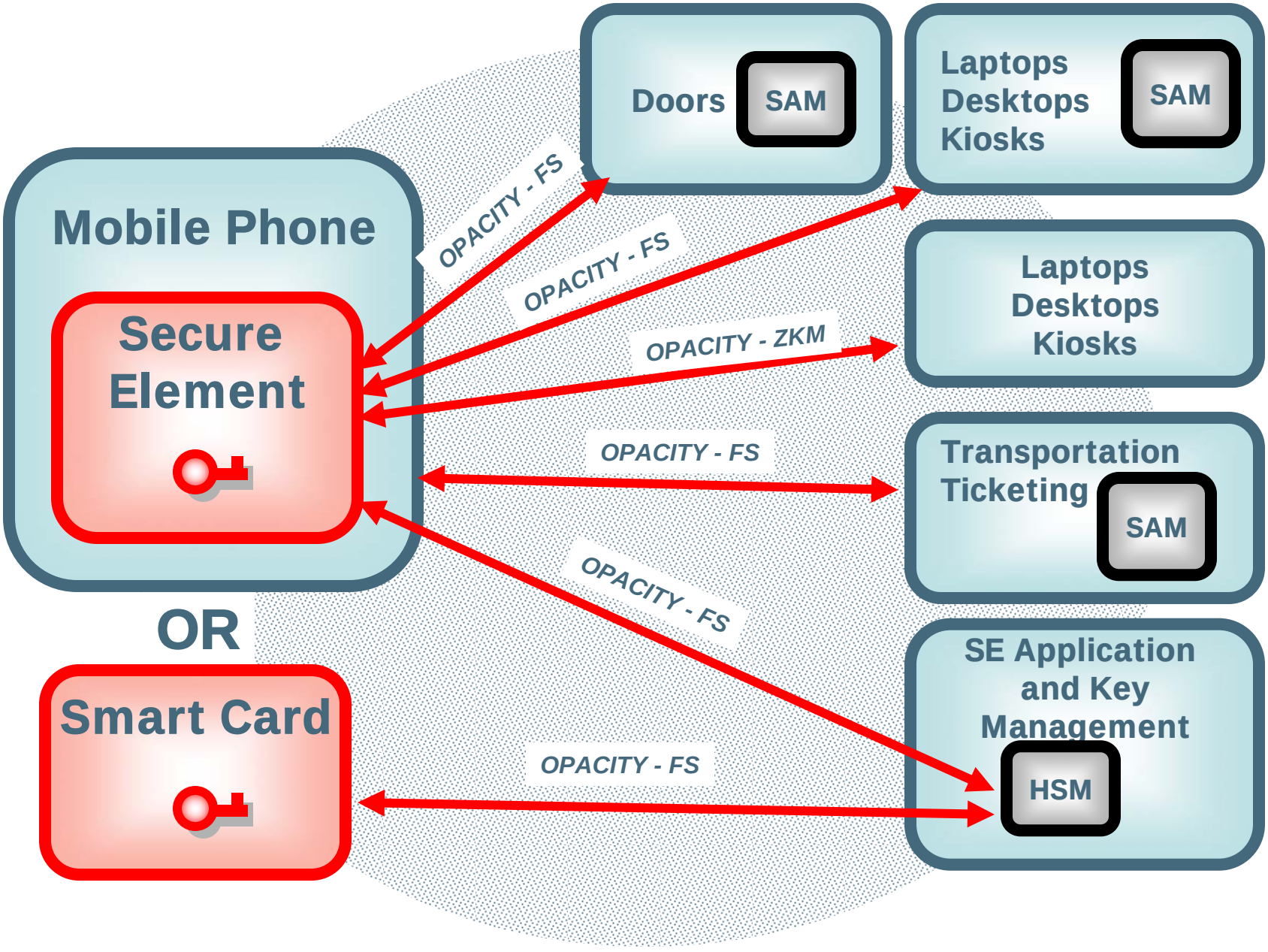


Use Cases – what can we do with OPACITY

- Strong Authentication and Secure channel to the door or door controller for physical access
  - Return encrypted code in one fast contactless transaction
  - Use of end-to-end secure messaging to transport PIN or biometrics or PACS credential via a contactless communication
- Strong Authentication and Secure channel for transit applications and ticketing
- Strong Authentication and Secure channel to remote services via trusted terminals (desktops, laptops, tablets, phones and kiosks)
  - Use of secure messaging to provide an end-to-end protected path for identity proofing and cloud service transaction signing
- Strong Authentication and Secure channel to desktops, laptops and kiosks for logical access
  - Use of secure messaging to provide an end-to-end protected path for document or transaction decryption and signature using a secure element or a smart card



Opacity – Use cases



OPACITY Simple Key Management: the CVCs

OPACITY is a PKI based Key establishment and Authentication protocol that relies on Elliptic Curve Cryptography ECDH and ECDSA as well as AES (128/256) and SHA256.

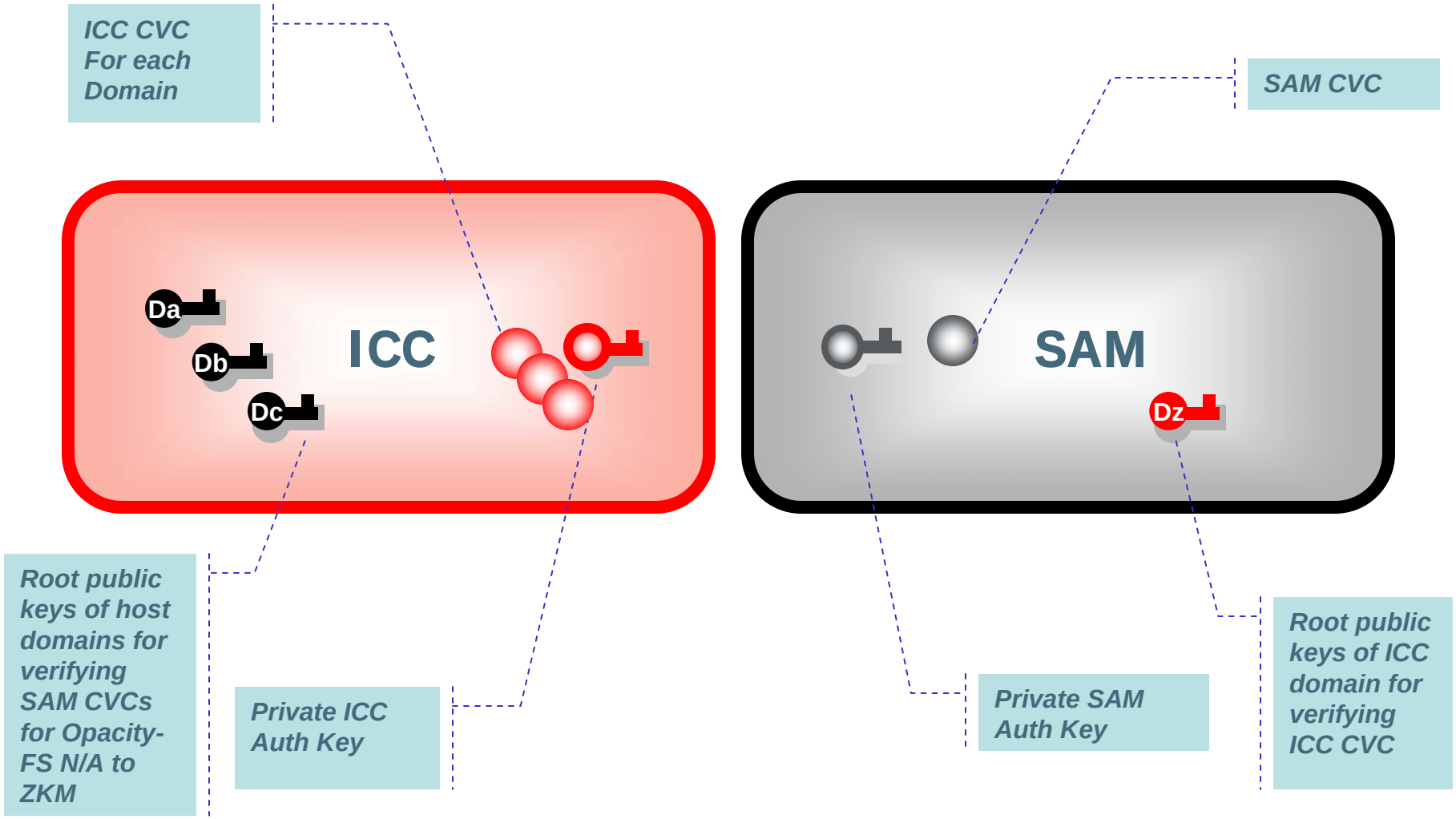
OPACITY is optimized for secure contactless transaction such as Physical Access.

In order to combine very fast performance and very high security OPACITY uses **Card Verifiable Certificates**, ie, CVC, which are tiny certificates defined per ISO7816-8 annex B, and specialized as in EN 14890-1 section 14. The CVCs are created and securely injected on smartcards by a Card Management System operating with a Digital Signatory based on a Hardware Security Module or a CA. (CVCs are not self-signed!) .

OPACITY simple key management principles:

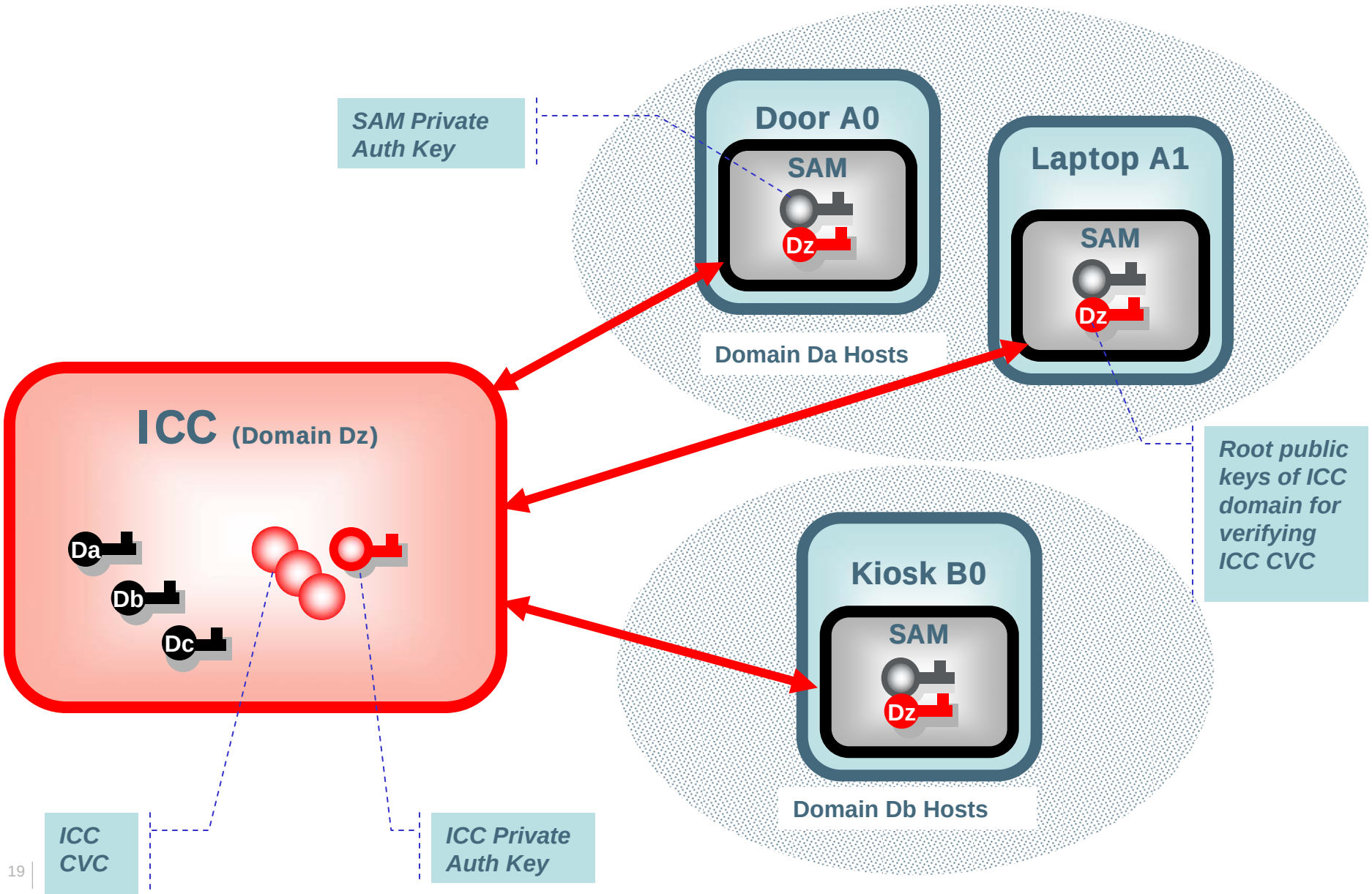
- Secret or private keys are never shared (keys are always unique to a device).
- No Master key is needed.

Opacity – Simple and efficient Key Management



No Distribution of Secret keys

Opacity – Cross Domain PKI



OPACITY Modes: ZKM and FS

OPACITY supports two main protocols:

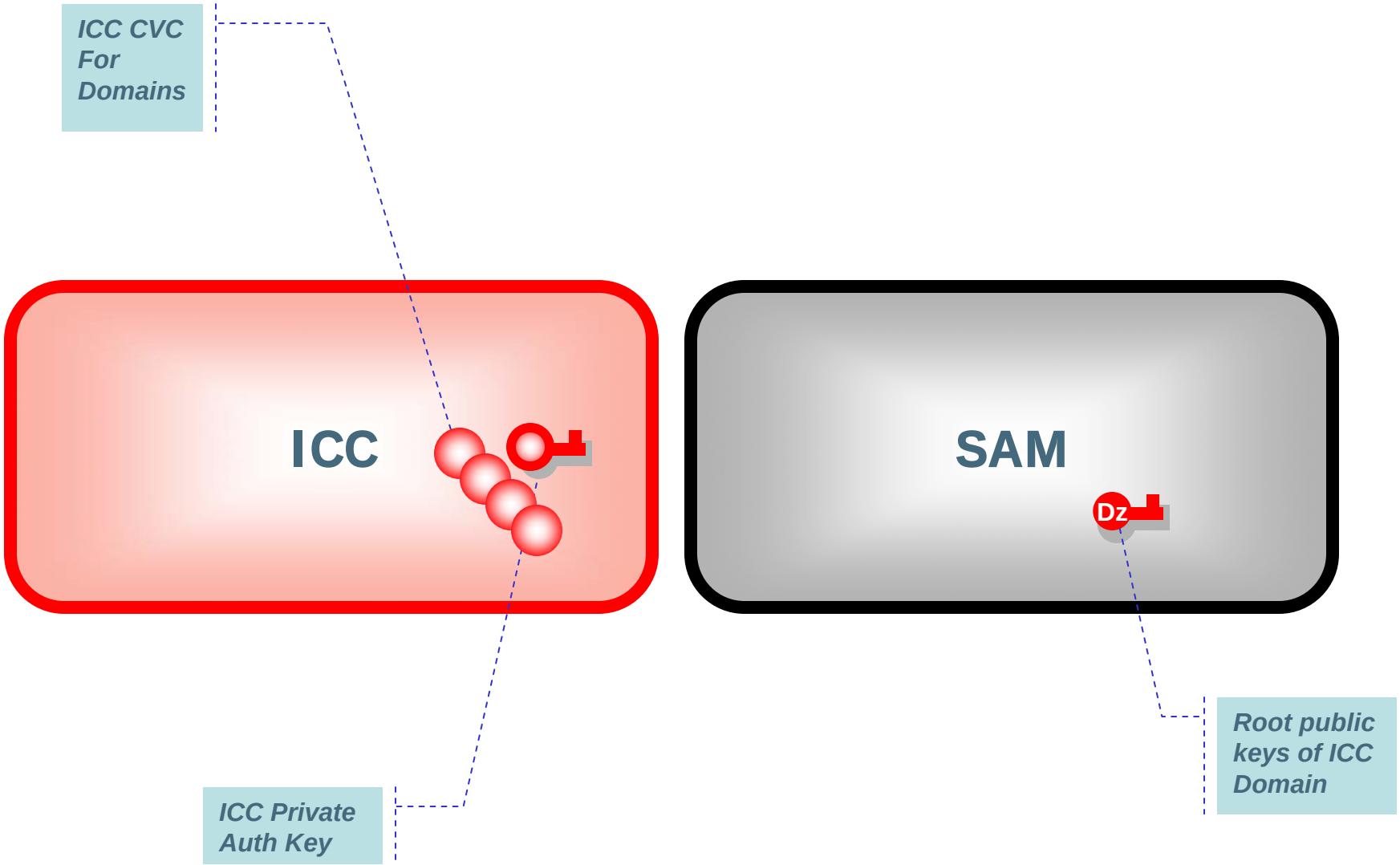
- Zero Key Management (ZKM aka. SMAv3) that does not require ANY secrets to be stored on a terminal.
- Forward Secrecy (FS), that provides enhanced privacy protection and durable confidentiality and very high level of Assurance, but require a static PKI credential stored and operated by the terminal.

An other mode, Forward Privacy (FP), in addition provides durable privacy.

The OPACITY reference implementation provides an Secure Authentication Module (SAM) applet. A SAM is a secure ICC or a smartcard device embedded into a Terminal, such as a Door Controller.

The ZKM/SMA v3 protocol does not require a SAM since it does not require any static secrets to be operated on the Terminal. Instead it only requires the root Public Key of the CVC Digital Signatory to be protected by the Terminal (in contrast other protocols require MASTER keys and private keys to be distributed to ALL terminals)

Opacity – Zero Key Management (SMA) option



OPACITY Implementation: SAMs

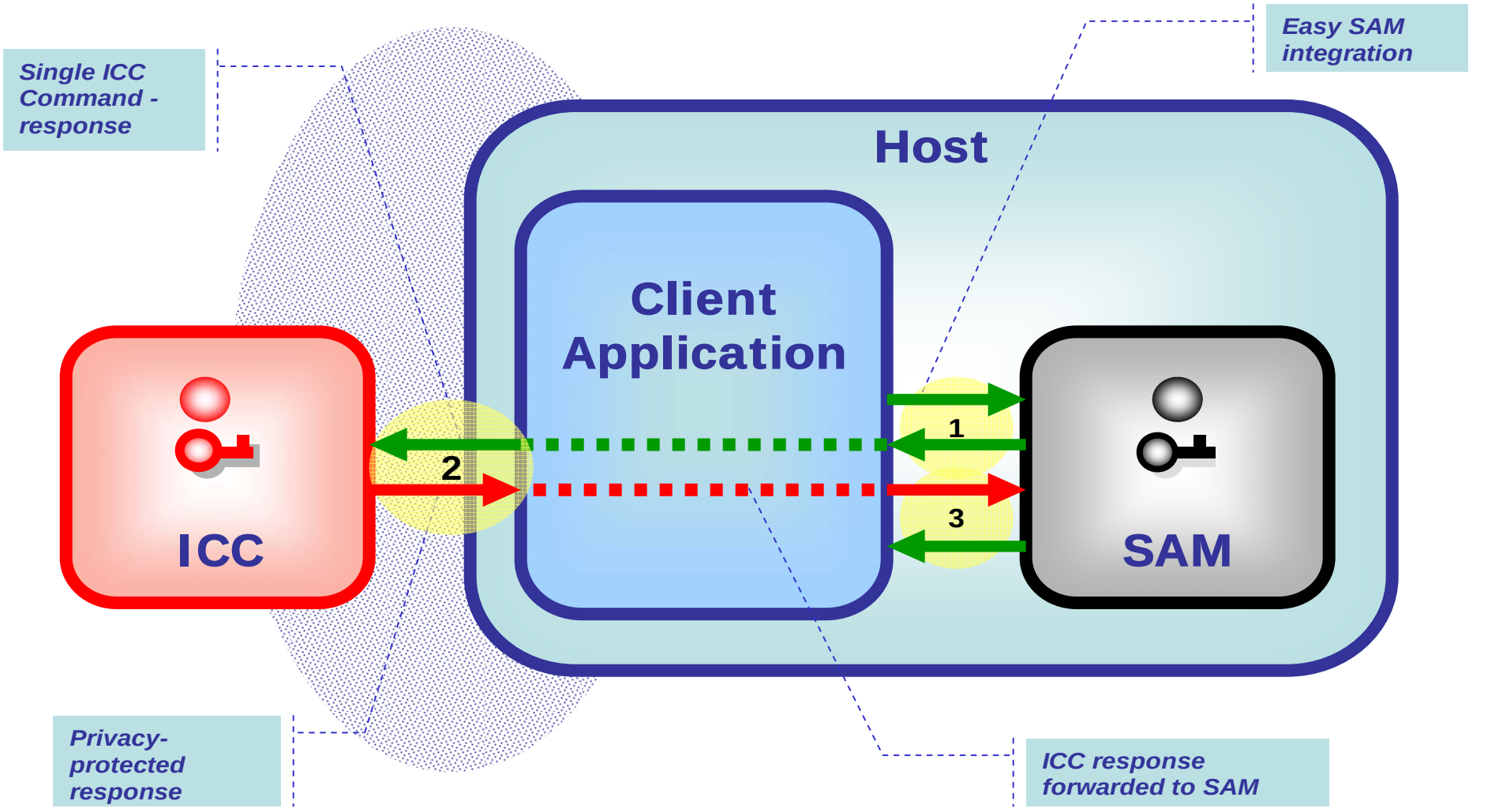
- Nevertheless, implementing advanced and **CERTIFIED** cryptography in an embedded device such as a door controller in software is both tricky, long, expensive and frustrating because the level of certification reached can never go beyond fips140-2 level **1.**

- For those who are still interested in a software implementation, the Opacity reference implementation provides a soft SAM library.

- But, in order to facilitate a quick and cheap integration into embedded devices the OPACITY reference implementation provides an applet that embeds ALL terminal cryptographic functions into a hardware SAM which can reach a FIPS140-2 level3 certification level !

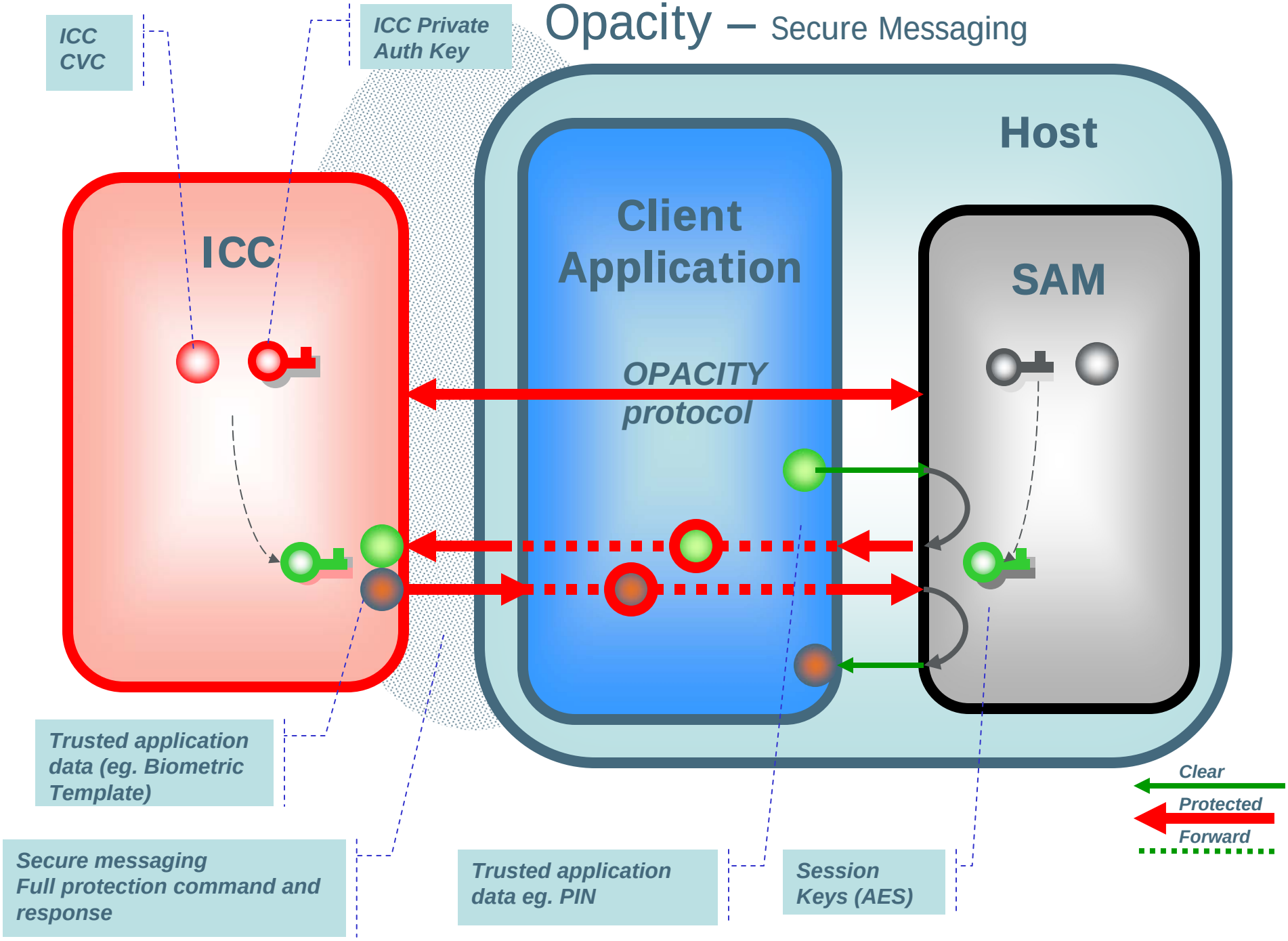
- In addition, the CMS and Key Management infrastructure that is deployed to issue smartcards to Users can also be leveraged to issue SAMs in a cheap, assured and scalable manner.

Opacity – Simple Command Flow

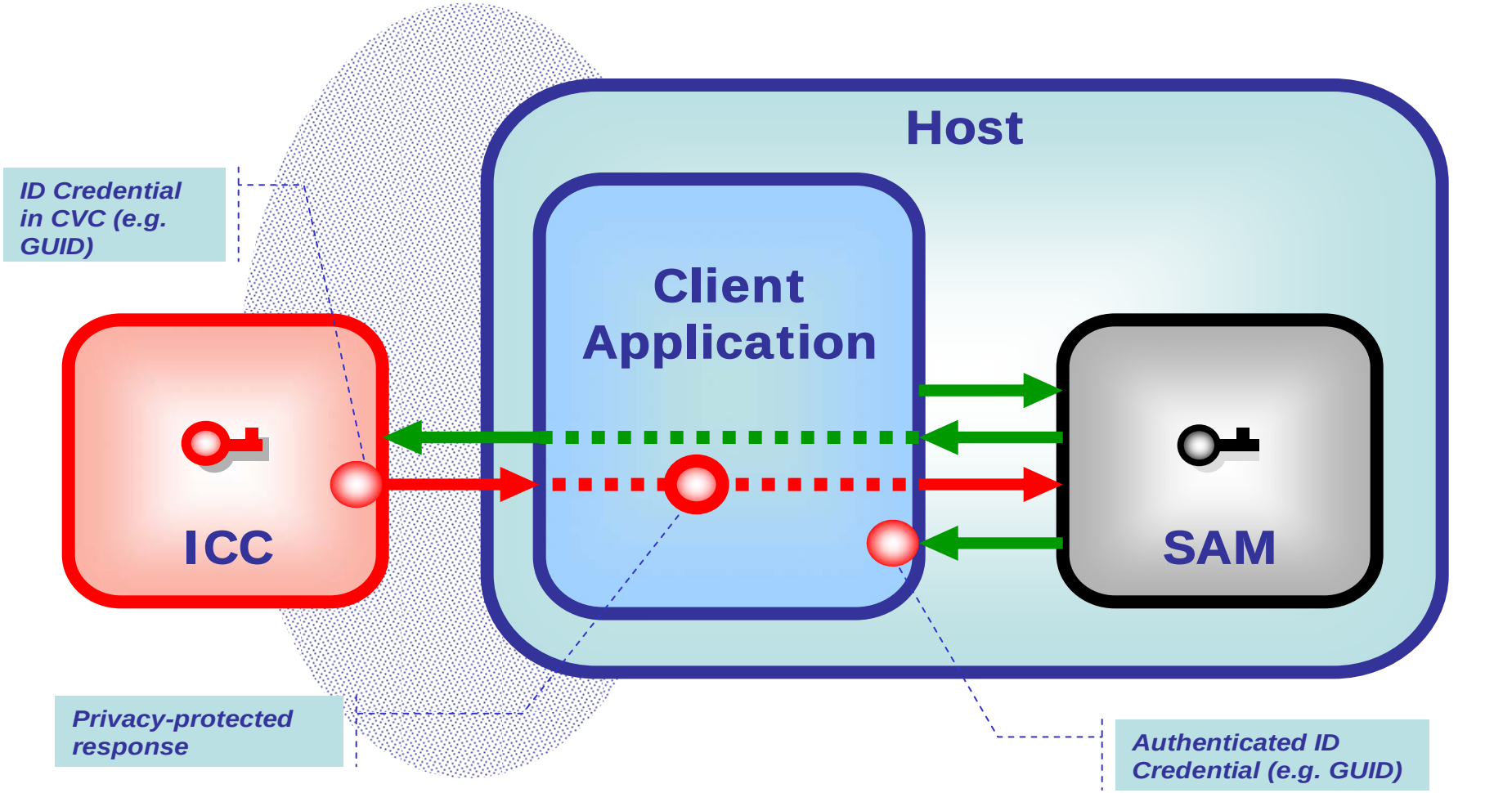


1. Generate SAM key pair 2. Authenticate SAM to ICC 3.Authenticate ICC to SAM

clear protected forward


Opacity – Secure Messaging
ICC
CVC
ICC Private
Auth Key
Host
ICC
Client
Application
SAM
OPACITY
protocol
Trusted application
data (eg. Biometric
Template)
Secure messaging
Full protection command and
response
Trusted application
data eg. PIN
Session
Keys (AES)
Clear
Protected
Forward

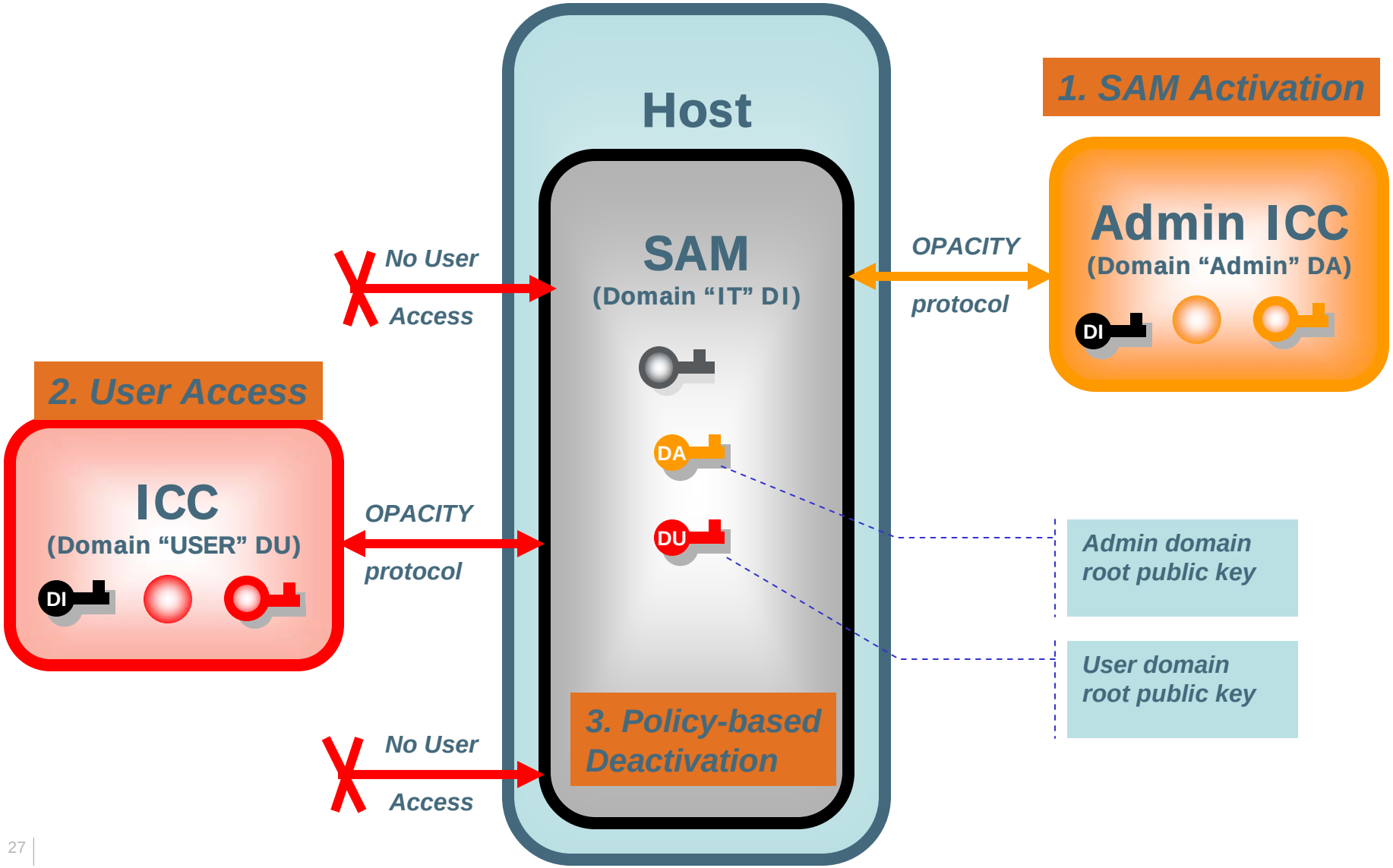
Opacity – Secure Credential Transfer



Other OPACITY Concepts: SAM Activation

- SAM activation:
  - SAM operations are protected by an Access Control Rule.*(Only an authorized system can operate the SAM so that if a Terminal/SAM is stolen, nothing is gained by the attacker).*
    - Admin operations are protected by GlobalPlatform Secure Channels.
    - Usage operations are protected by a PIN based or an OPACITY based authentication by the Control Panel to the SAM. (*The Control Panel is usually in a secured location*)
  - The SAM ACR must be fulfilled by the PACS control panel, ie, the panel “activates” the SAM. If a SAM is de-powered it loses its activation state.

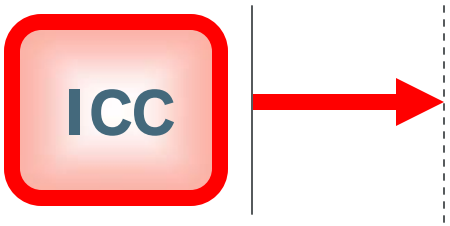
Opacity – SAM Activation



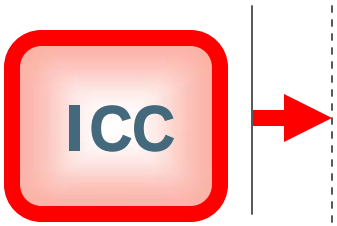
Other OPACITY Concepts: Persistent Binding

- Persistent Binding (PB):
  - The OPACITY pre-session keys are cached by the User applet and the Terminal applet.
  - This allows even faster transactions after the first time a card sees a terminal.
  - The time to leave of persisted pre-session keys is policy dependant.

Opacity – User experience

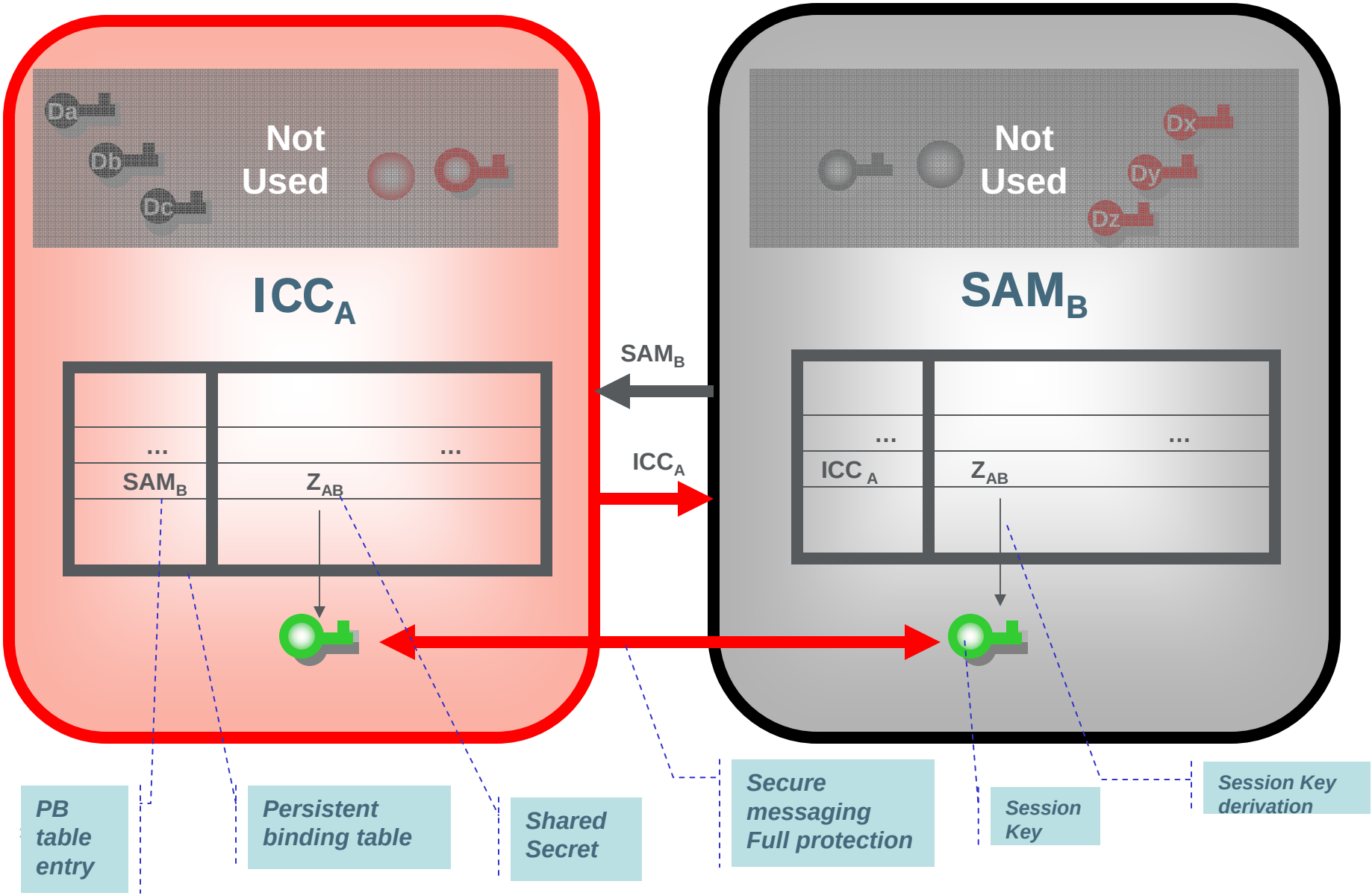


Time to open door or computer logon (W/O PB)

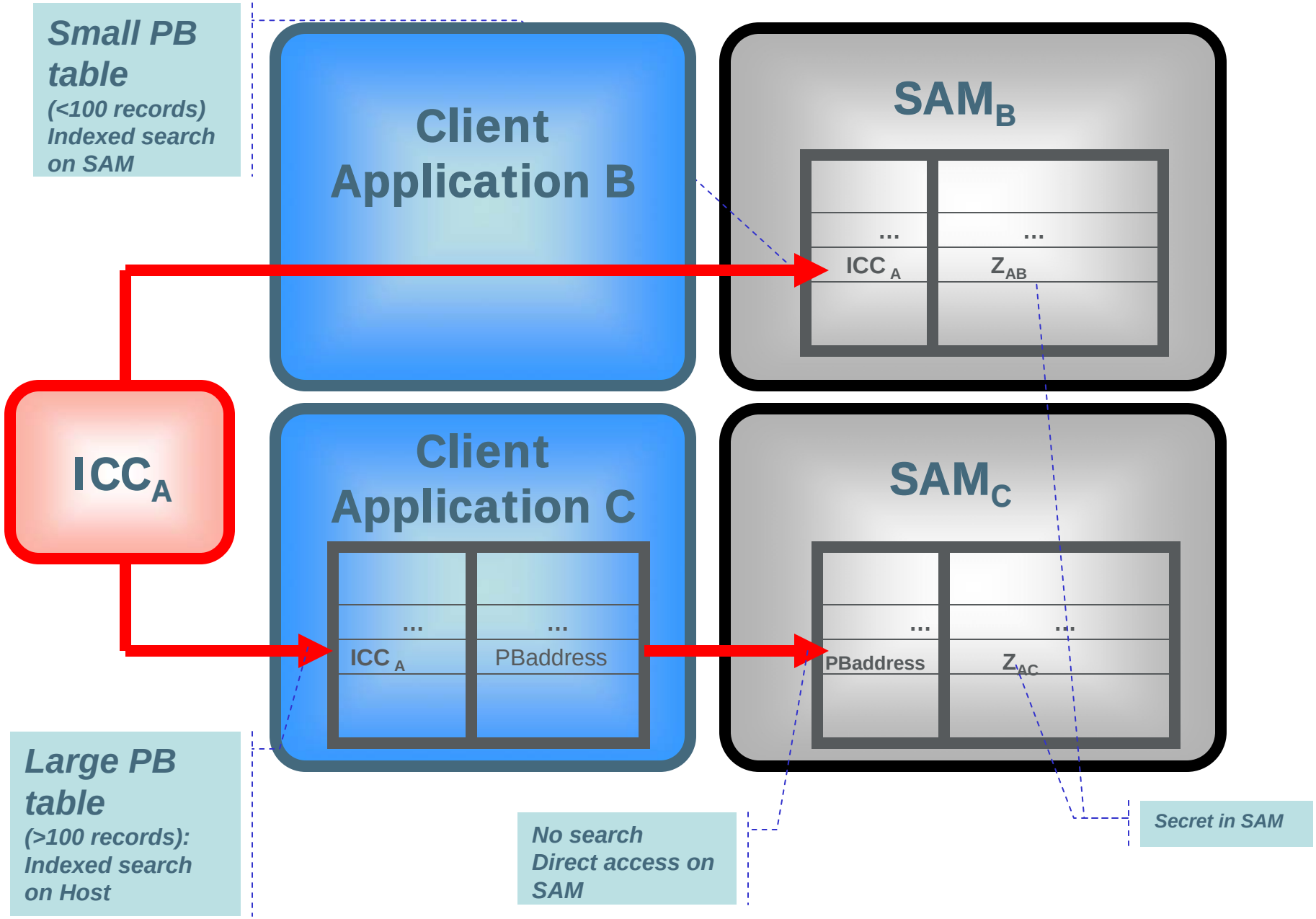


Time to open door or computer logon (W/ PB)

Opacity – Persistent Binding

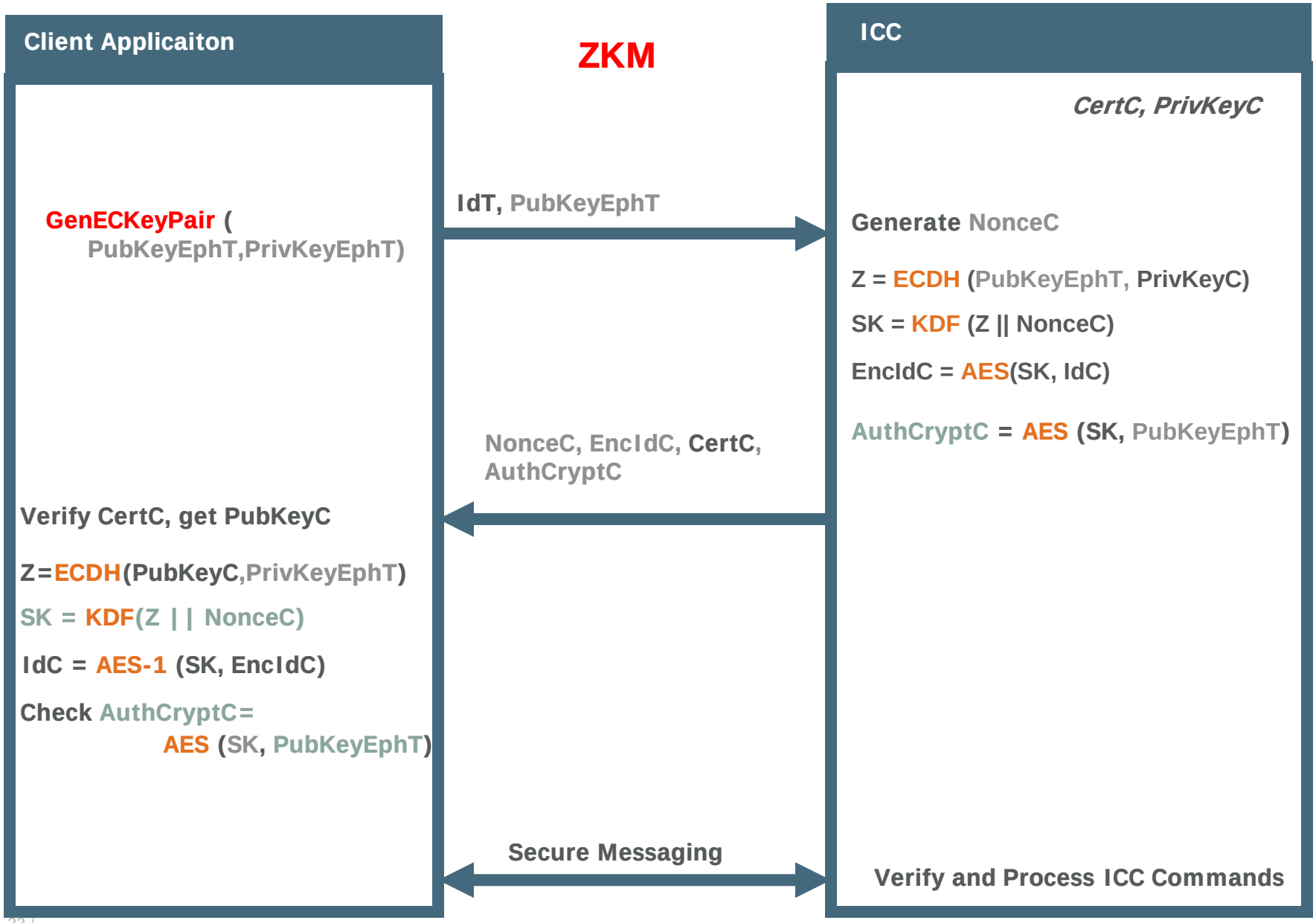


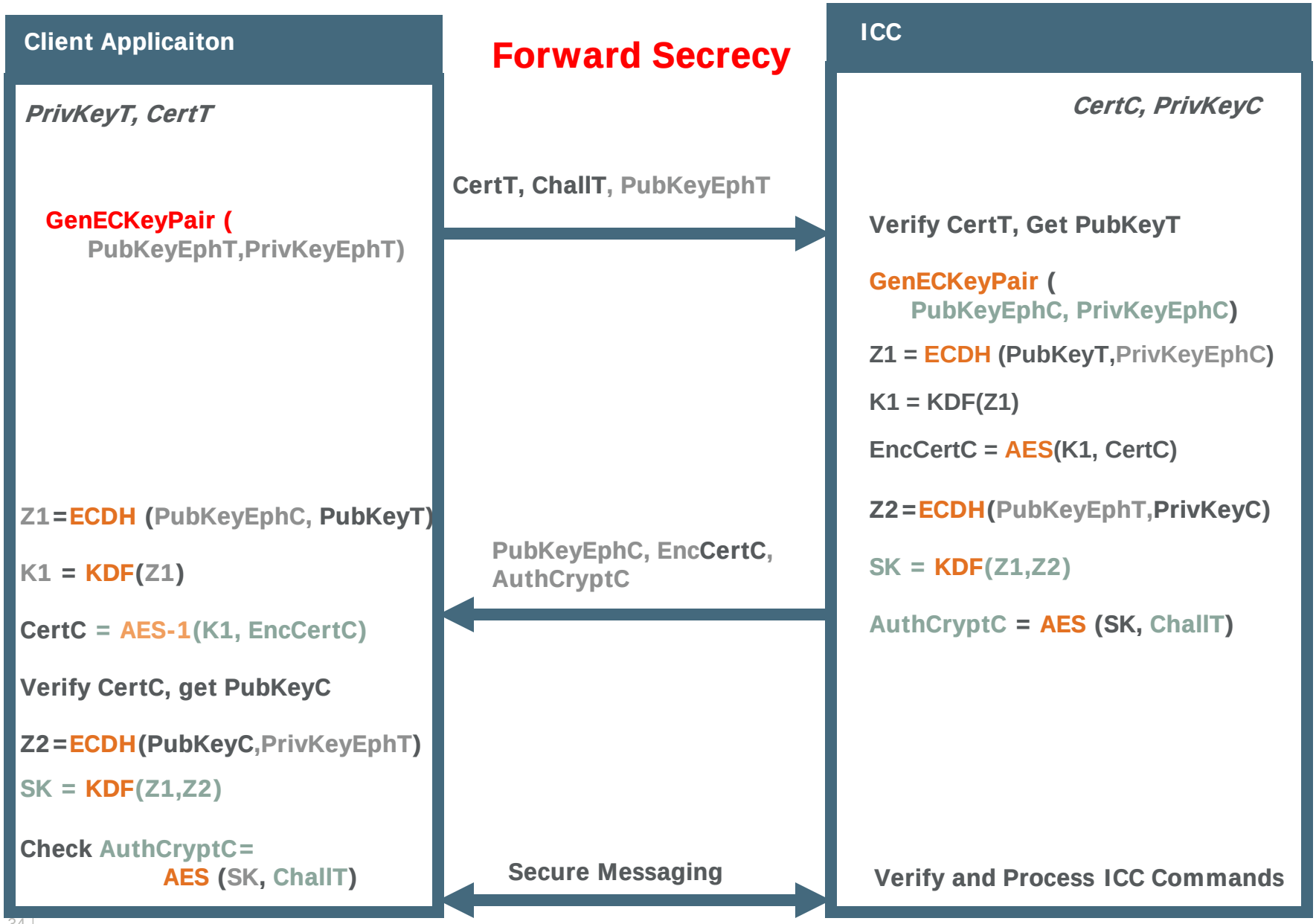
Opacity – Scalable Persistent Binding



OPACITY

The Protocol


Client Applicaiton
ZKM
ICC
CertC, PrivKeyC
GenECKeyPair (PubKeyEphT,PrivKeyEphT)
IdT, PubKeyEphT
Generate NonceC
Z = ECDH (PubKeyEphT, PrivKeyC)
SK = KDF (Z || NonceC)
EncIdC = AES(SK, IdC)
AuthCryptC = AES (SK, PubKeyEphT)
NonceC, EncIdC, CertC, AuthCryptC
Verify CertC, get PubKeyC
Z=ECDH(PubKeyC,PrivKeyEphT)
SK = KDF(Z | | NonceC)
IdC = AES-1 (SK, EncIdC)
Check AuthCryptC= AES (SK, PubKeyEphT)
Secure Messaging
Verify and Process ICC Commands


Client Applicaiton
Forward Secrecy
ICC
PrivKeyT, CertT
CertC, PrivKeyC
CertT, ChallT, PubKeyEphT
GenECKeyPair (PubKeyEphT,PrivKeyEphT)
Verify CertT, Get PubKeyT
GenECKeyPair (PubKeyEphC, PrivKeyEphC)
Z1 = ECDH (PubKeyT,PrivKeyEphC)
K1 = KDF(Z1)
EncCertC = AES(K1, CertC)
Z2 = ECDH(PubKeyEphT,PrivKeyC)
SK = KDF(Z1,Z2)
AuthCryptC = AES (SK, ChallT)
PubKeyEphC, EncCertC, AuthCryptC
Z1 = ECDH (PubKeyEphC, PubKeyT)
K1 = KDF(Z1)
CertC = AES-1(K1, EncCertC)
Verify CertC, get PubKeyC
Z2 = ECDH(PubKeyC,PrivKeyEphT)
SK = KDF(Z1,Z2)
Check AuthCryptC= AES (SK, ChallT)
Secure Messaging
Verify and Process ICC Commands

Questions and Answers

- elesaint@actividentity.com


ACTIVIDENTITY™