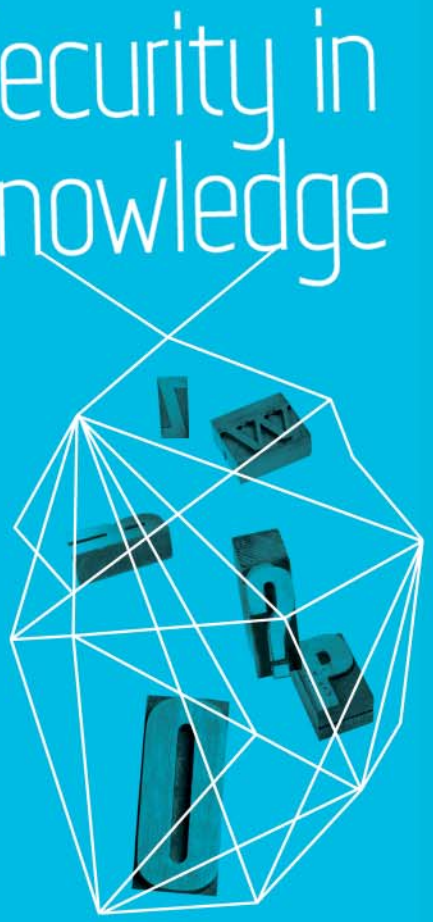
CRYPTOGRAPHY AS A SERVICE

Peter Robinson
RSA, The Security Division of EMC



RSA CONFERENCE
EUROPE 2013

Session ID: ADS-R01
Session Classification: Advanced

Introduction

- Deploying cryptographic keys to end points such as smart phones, virtual machines in The Public Cloud, and smart grid equipment is risky.
- This presentation proposes a Cryptography as a Service (CaaS) model which allows operations to be performed without exposing cryptographic keys and recommends how to overcome the pitfalls associated with this technology.

Acknowledgements

- I acknowledge the combined efforts of the many people who have progressed the CaaS technology area in RSA to where it is today.
- In particular I thank: Steve Schmalz, Jaimee Brown, Eric Young, Stefan Pingel, Sean Parkinson, Sandra Carielli, Ken Ray, Didi Dotan, Mike Shanzer, Ingo Schubert, Ari Juels, Karen Reinhardt, David Healy, and Nikos Triandopoulos.

Pre-Requisites

- This session is classified as **Advanced** and assumes you have an understanding of:
  - Security concepts such as Entropy.
  - Cryptographic Algorithms such as RSA, ECDSA, ECDH, AES, SHA256, HMAC/SHA256, and PRNGs.
  - Transport Layer Security (TLS).
  - Hyper Text Transfer Protocol (HTTP).
  - Cryptographic Message Syntax (CMS) / PKCS #7.

Agenda

- Problem Domain and Examples.
- CaaS Definition.
- End Point Authentication.
- Crypto Algorithms and Security Operations.
- System Impact of Moving to CaaS.
- CaaS Usage.

Problem Domain and Examples

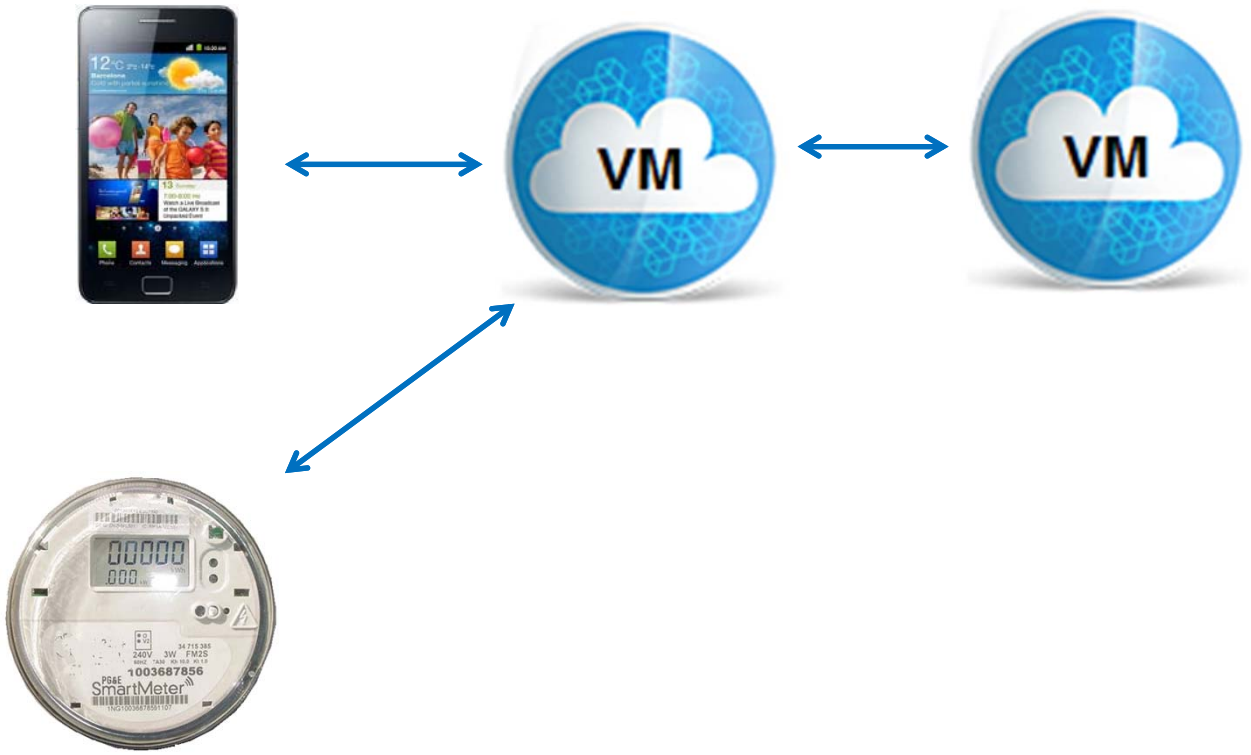
#RSAC

RSA CONFERENCE EUROPE 2013

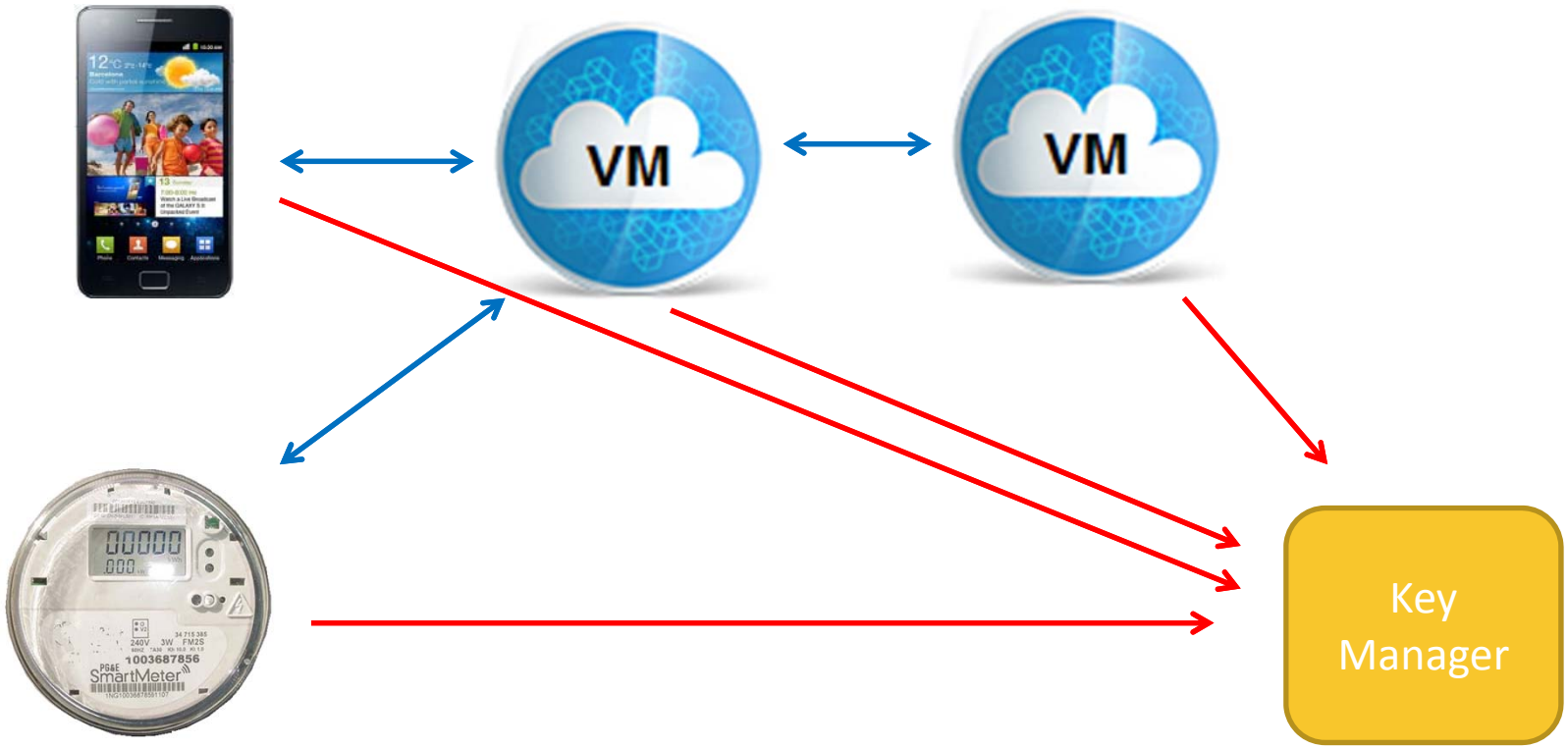
Vulnerable End Points

- Poor entropy: Bad place to generate keys.
- Heightened probability of compromise.

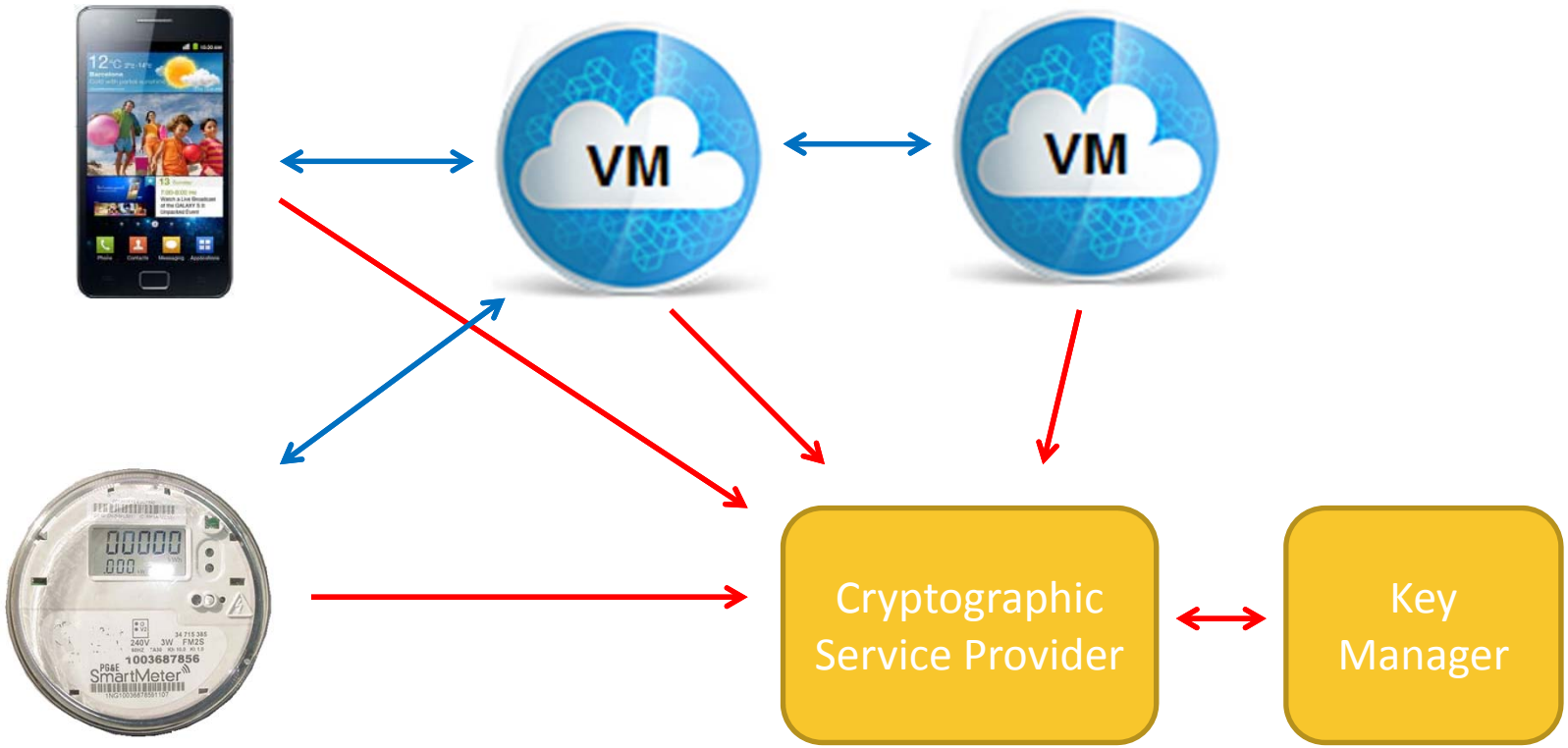
End Point Key Gen and Crypto



End Point Cryptography



Cryptography as a Service



Example: Smart Phone App

- Sign out-going email.
- Decrypt in-coming email.
- View and create protected information:
  - Stored off-device: Documents stored in the cloud.
  - On-device: Address book.

Example: Smart Phone App

- Sign out-going email:
  - Email signing private key.
- Decrypt in-coming email:
  - Email decryption private key.
- View and create protected information:
  - MAC and symmetric encryption: Symmetric keys.
  - Asymmetric signing and decryption: Private keys.

Example: Smart Phone App

- Compromise scenarios:
  - Attacker steals smart phone and attempts to compromise data or keys on the phone.
  - Attacker steals a back-up of the phone.
  - Malware App steals data or keys.
  - Malware App compromises or biases entropy sources which results in more easily guessed keys.

Example: Smart Phone App

- Multiple devices:
  - If you have multiple devices, the problems are multiplied.

Example: Virtual Machine

- HTTPS server.
- Audit logging.
- Secure Messaging / Transactions.

Example: Virtual Machine

- HTTPS server:
  - Signing or decryption private key (depending on TLS cipher suite).
- Audit logging:
  - Symmetric keys for MAC and symmetric encrypt, or asymmetric keys for signing and enveloping.
- Secure Messaging or Transactions:
  - Symmetric or asymmetric keys for signing and encryption.

Example: Virtual Machine

- Compromise scenarios:
  - Attacker copies the VM.
  - Attacker steals a back-up of the VM.
  - Attacker uses a misconfiguration or vulnerability to gain access to the VM.
  - Attacker compromises or biases entropy sources which results in more easily guessed keys.

Example: Smart Grid

- Status information from Smart Grid equipment.
- Verification of software updates.
- Verification and decryption of control messages.

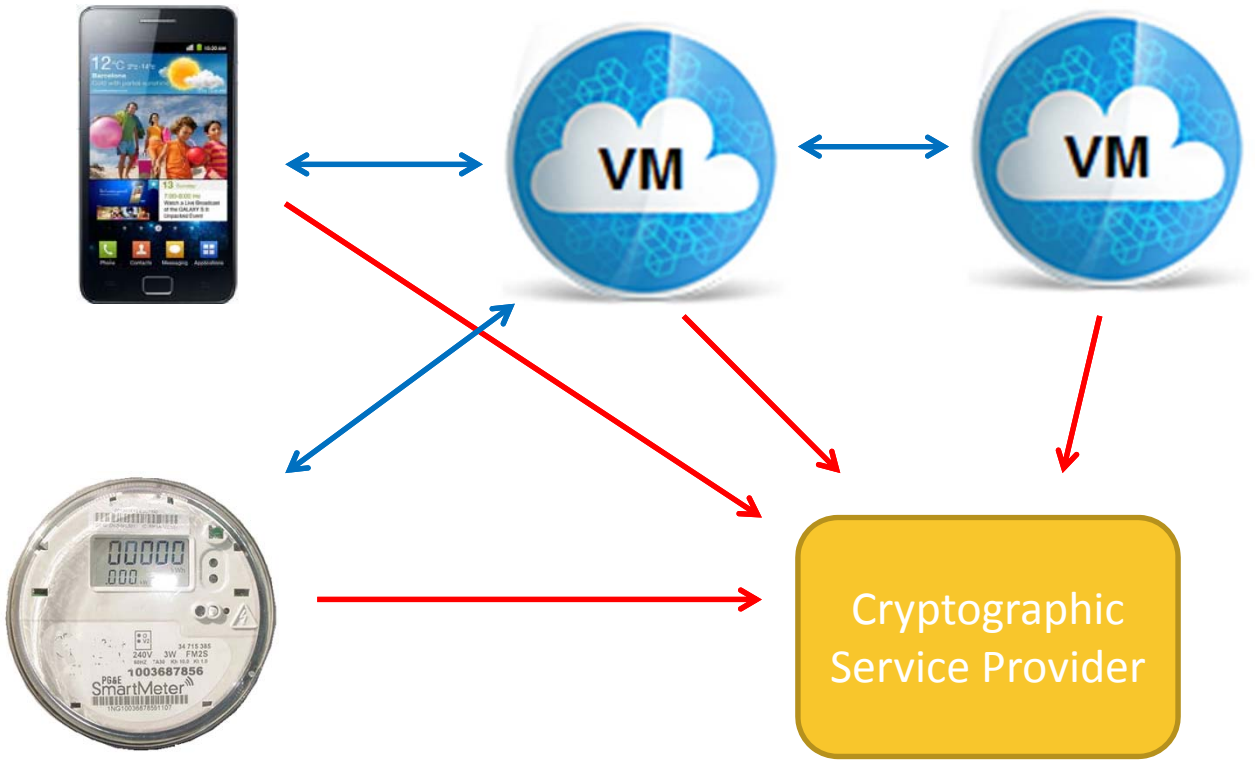
Example: Smart Grid

- Status information from Smart Grid equipment:
  - Private key for signing and symmetric key for encryption.
- Verification of software updates:
  - Public key for verification.
- Verification and decryption of control messages:
  - Public key for verification and symmetric key for decryption.

Example: Smart Grid

- Compromise scenarios:
  - Attacker obtains access to the smart grid device and attempts to compromise data or keys on the device.
  - Attacker uses a misconfiguration or vulnerability to take control of the device.
  - Attacker uses knowledge of the device's entropy sources to guess keys.

Cryptography as a Service

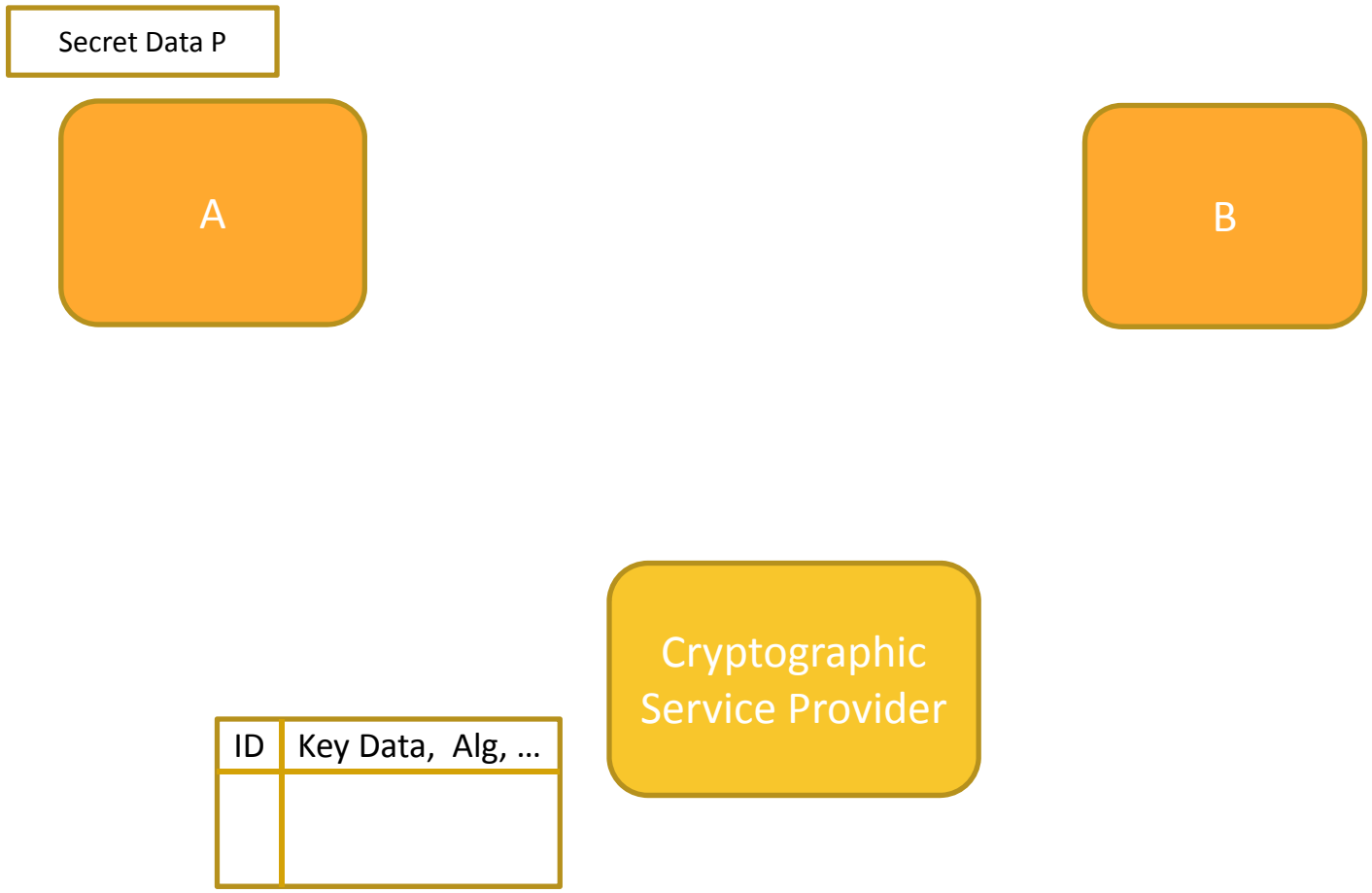


CaaS Definition

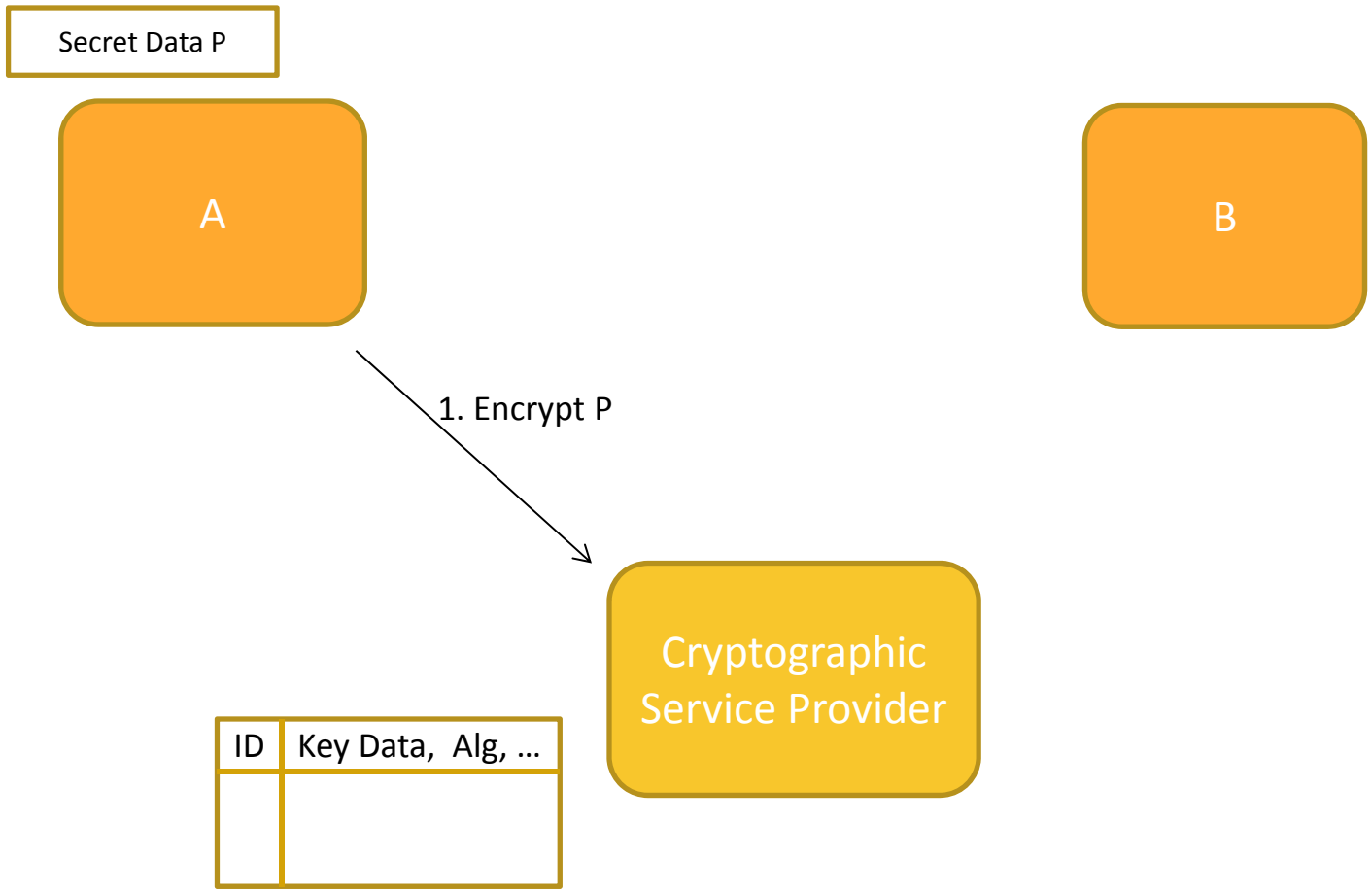
Cryptography as a Service

- Keyed cryptographic operations, such as encryption and decryption, are performed by a CaaS provider on behalf of a device via web services APIs.
- The cryptographic keys used to perform these operations are stored within the CaaS provider, so devices do not possess these keys at any time.

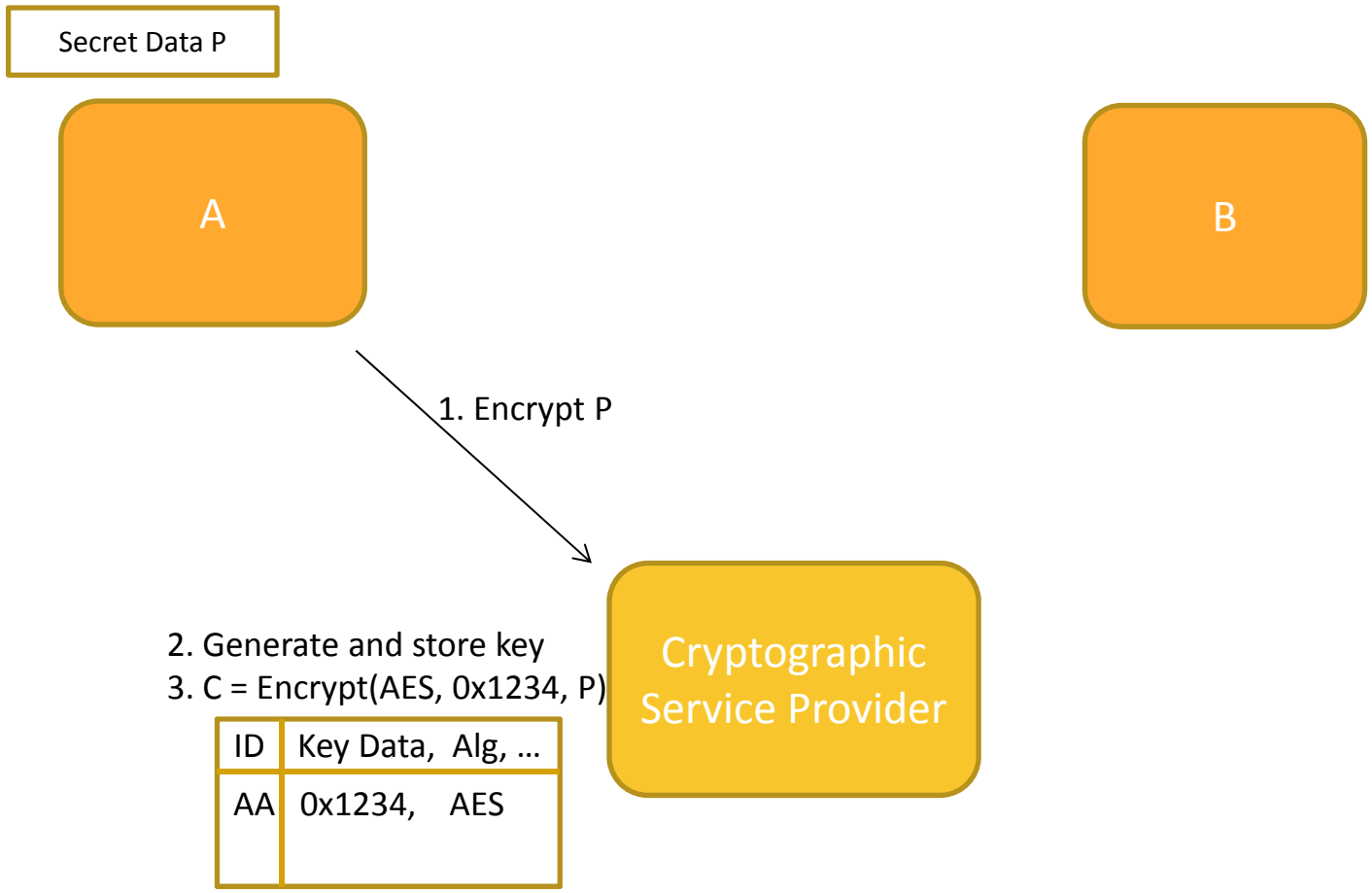
Cryptography as a Service



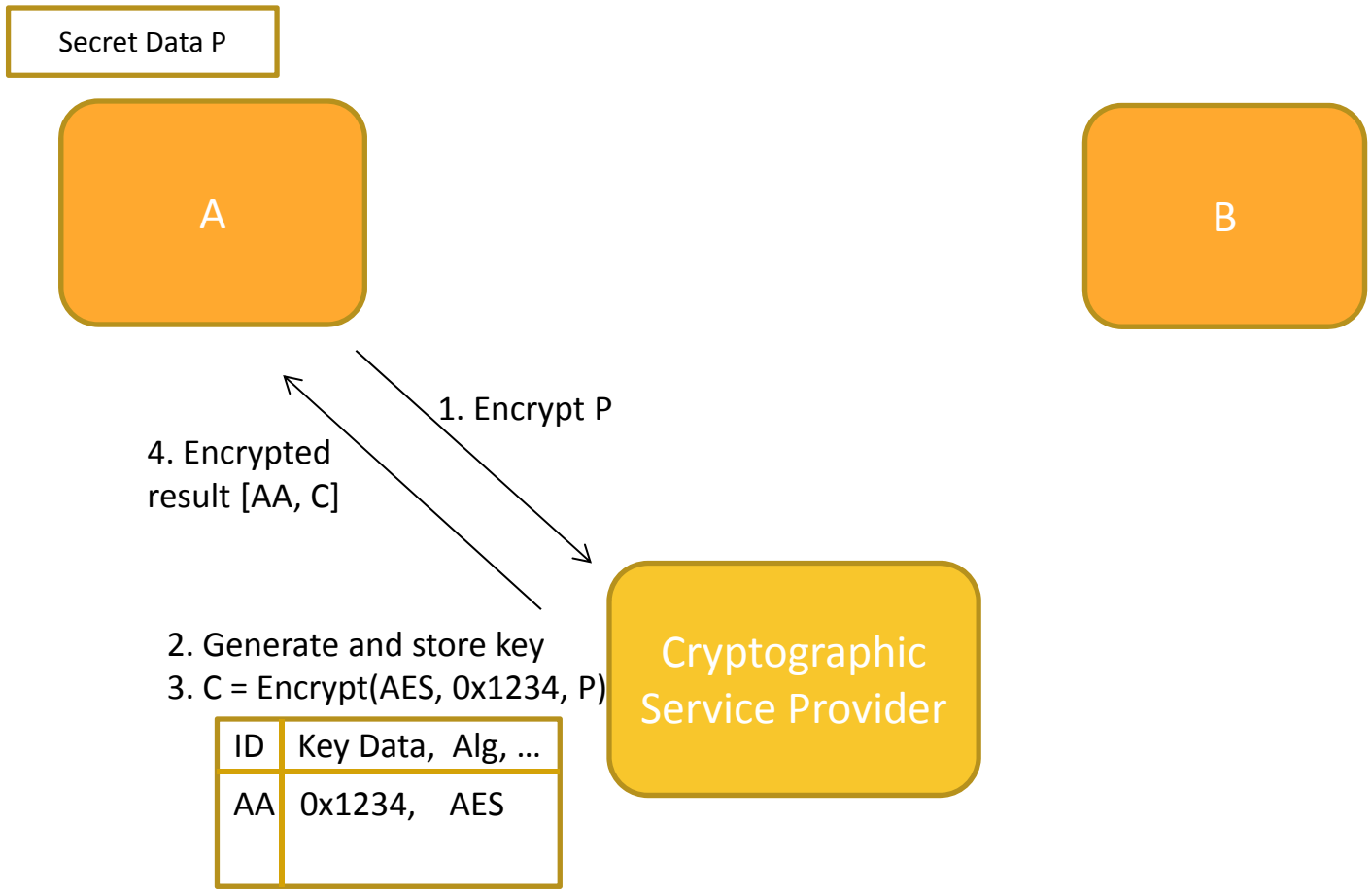
Cryptography as a Service



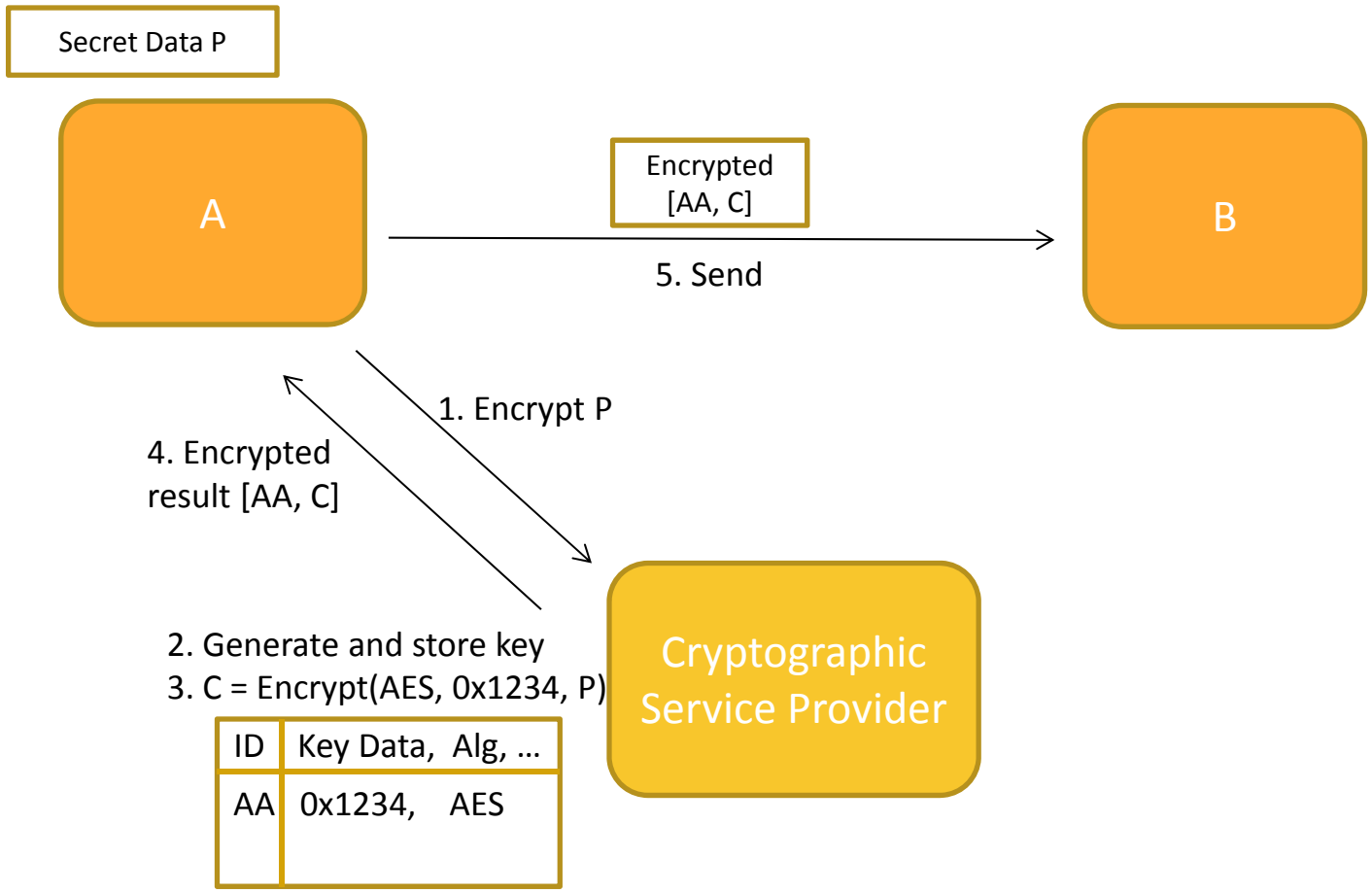
Cryptography as a Service



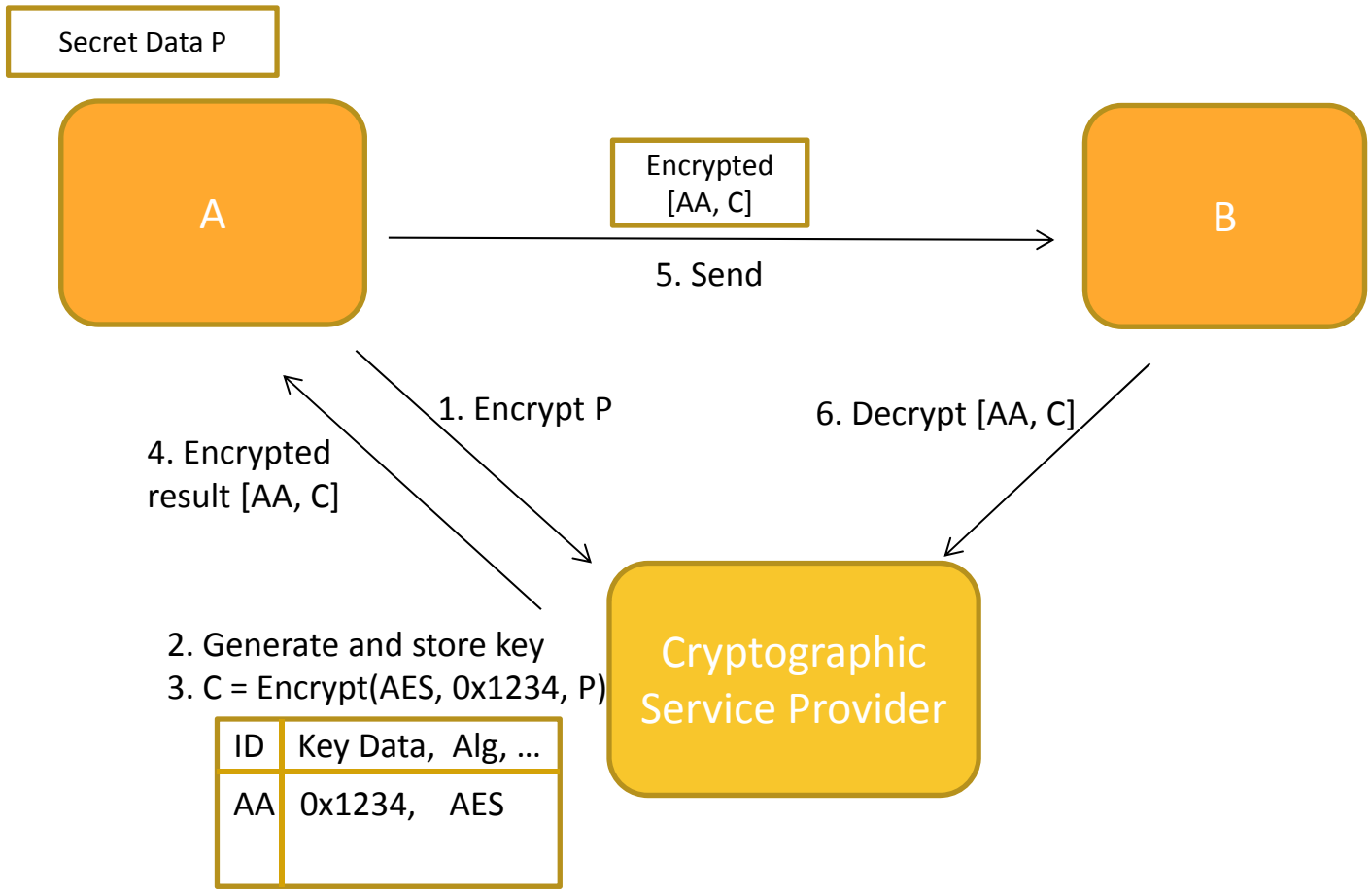
Cryptography as a Service



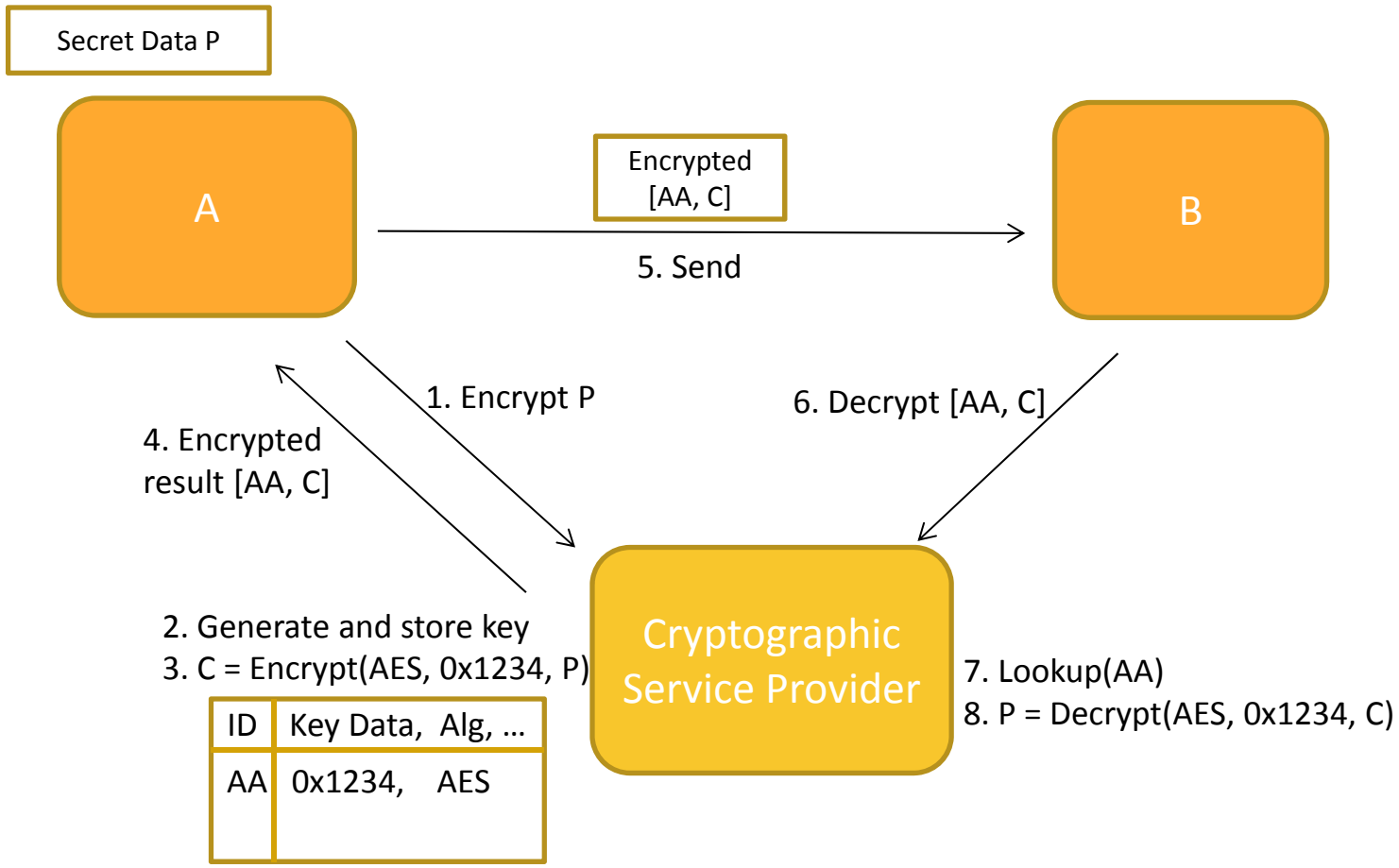
Cryptography as a Service



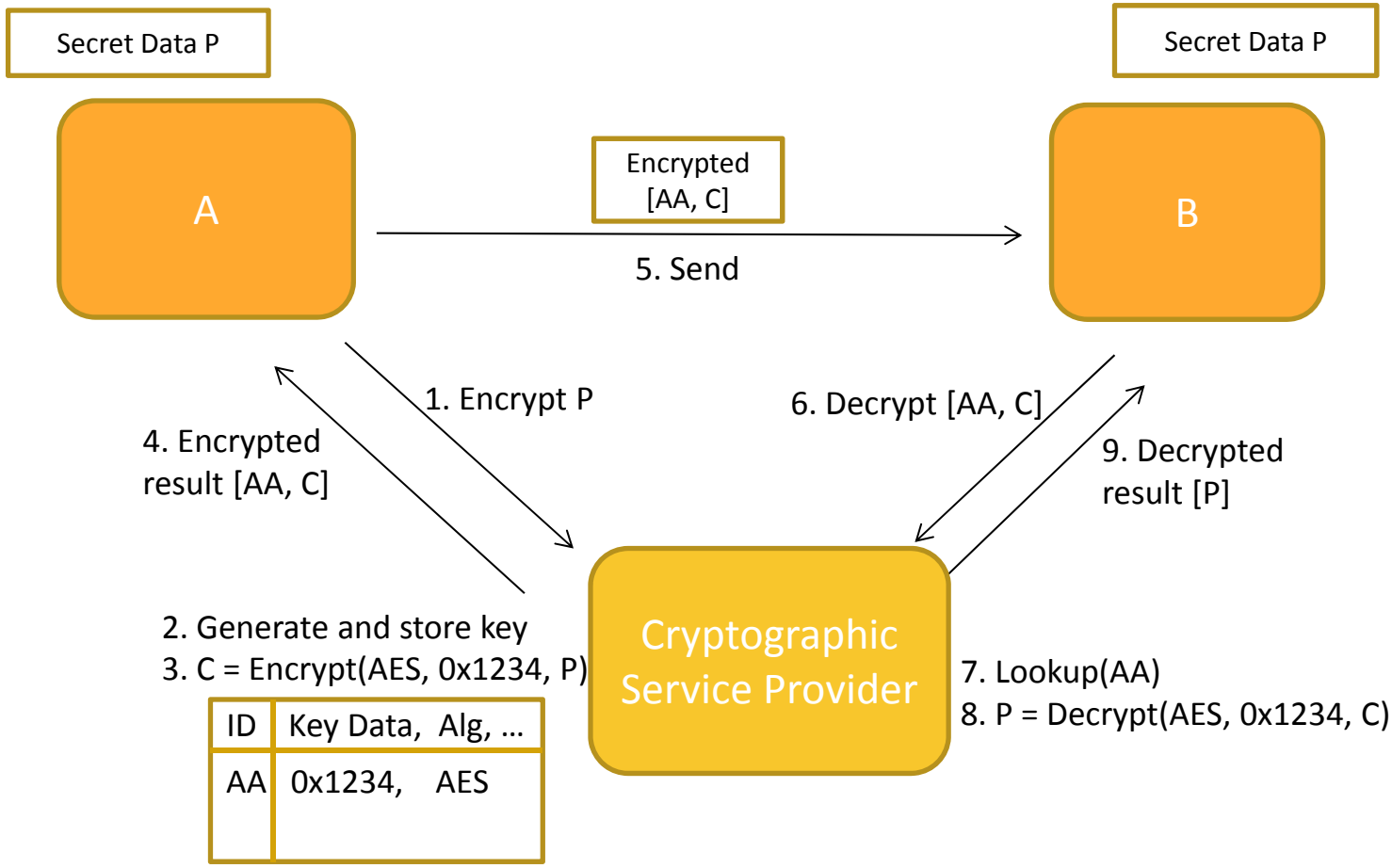
Cryptography as a Service



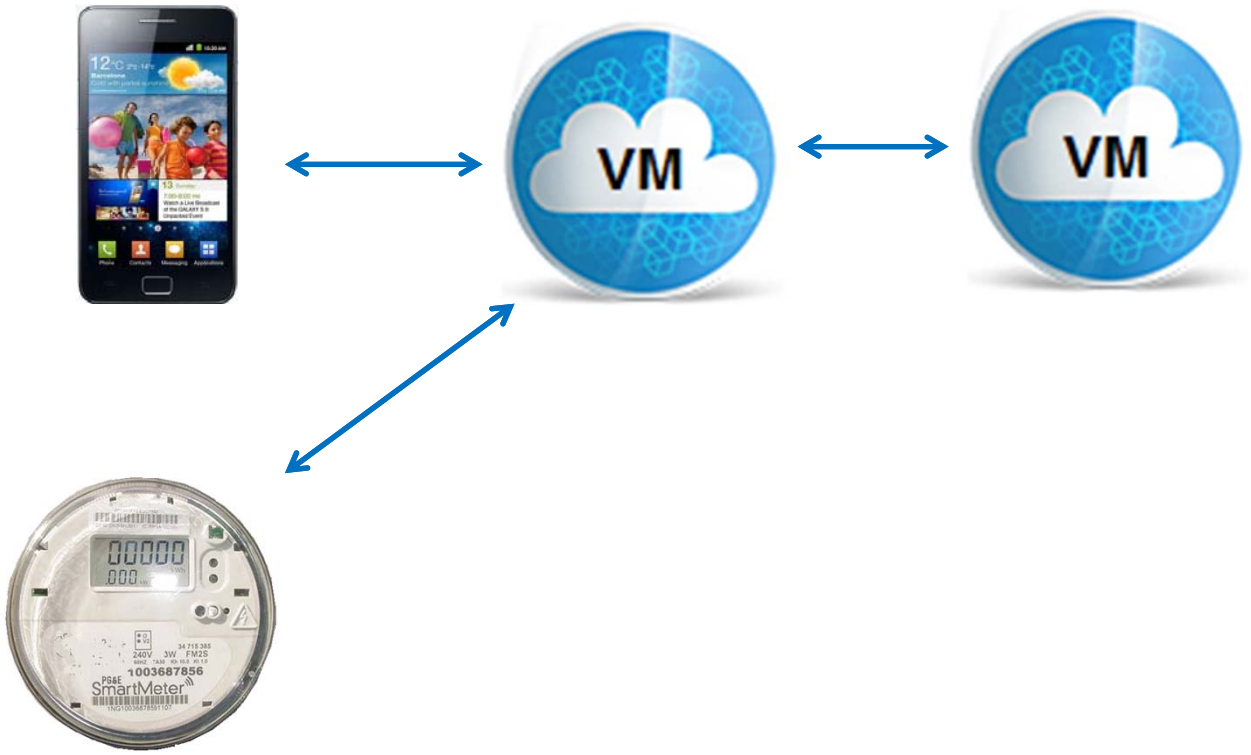
Cryptography as a Service



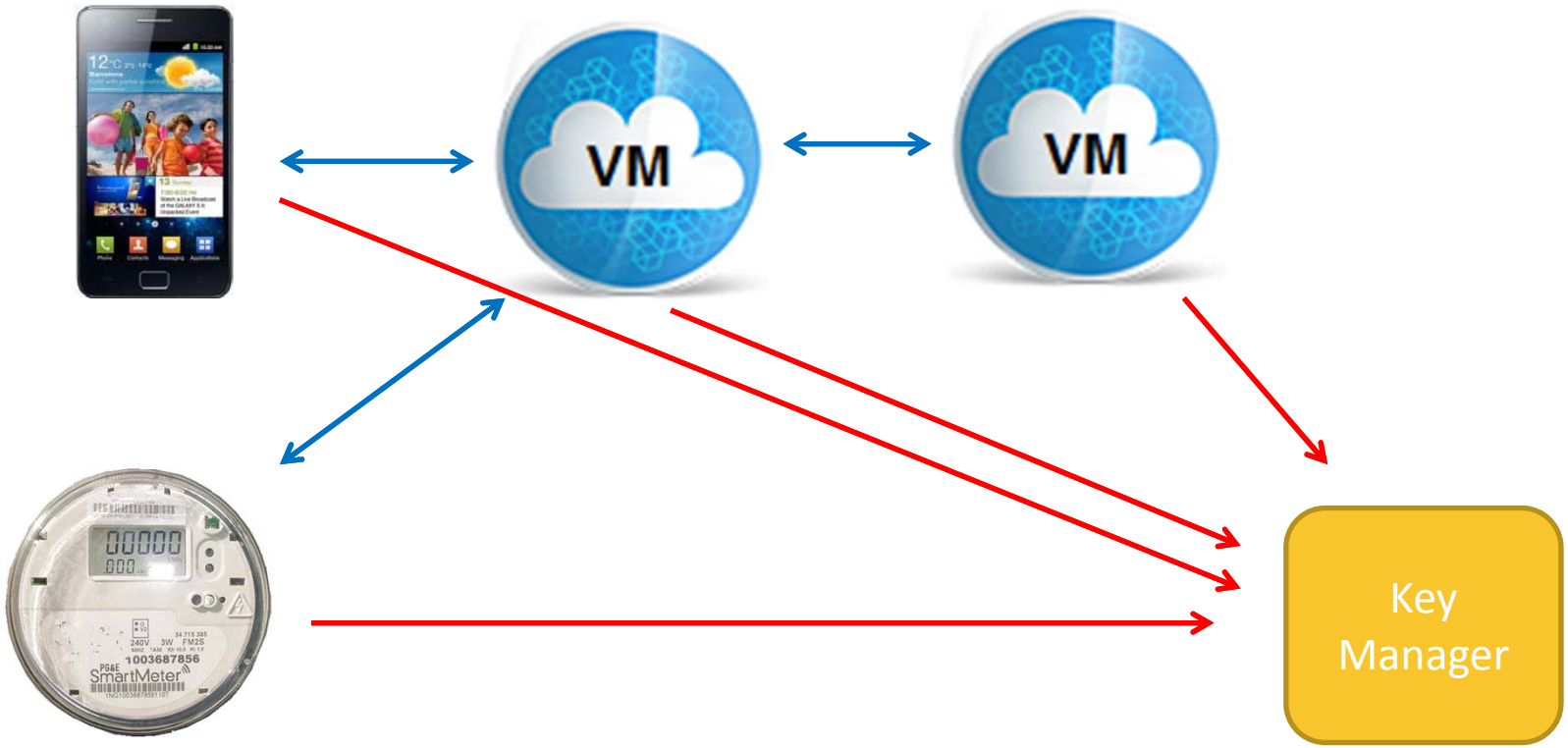
Cryptography as a Service



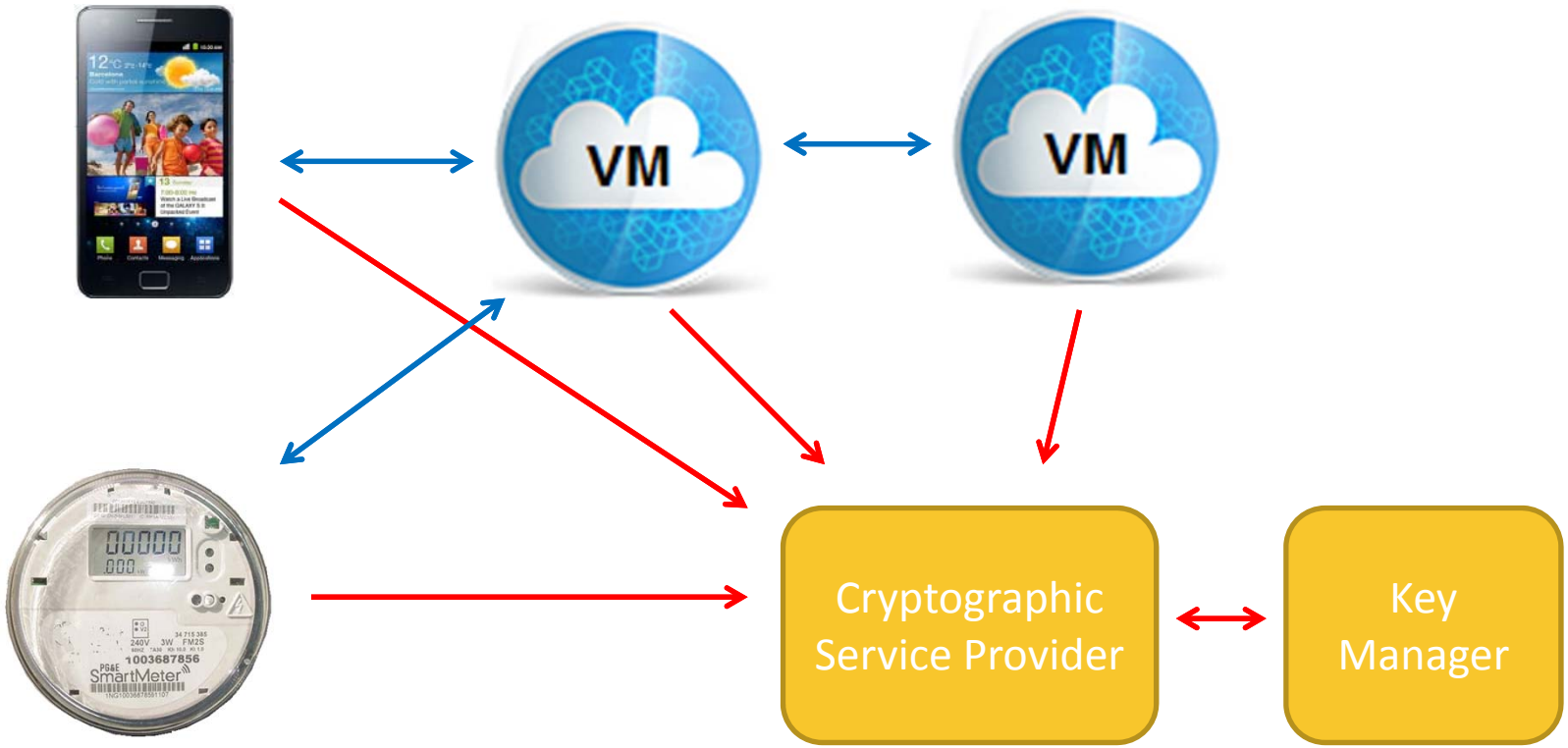
End Point Key Gen and Crypto



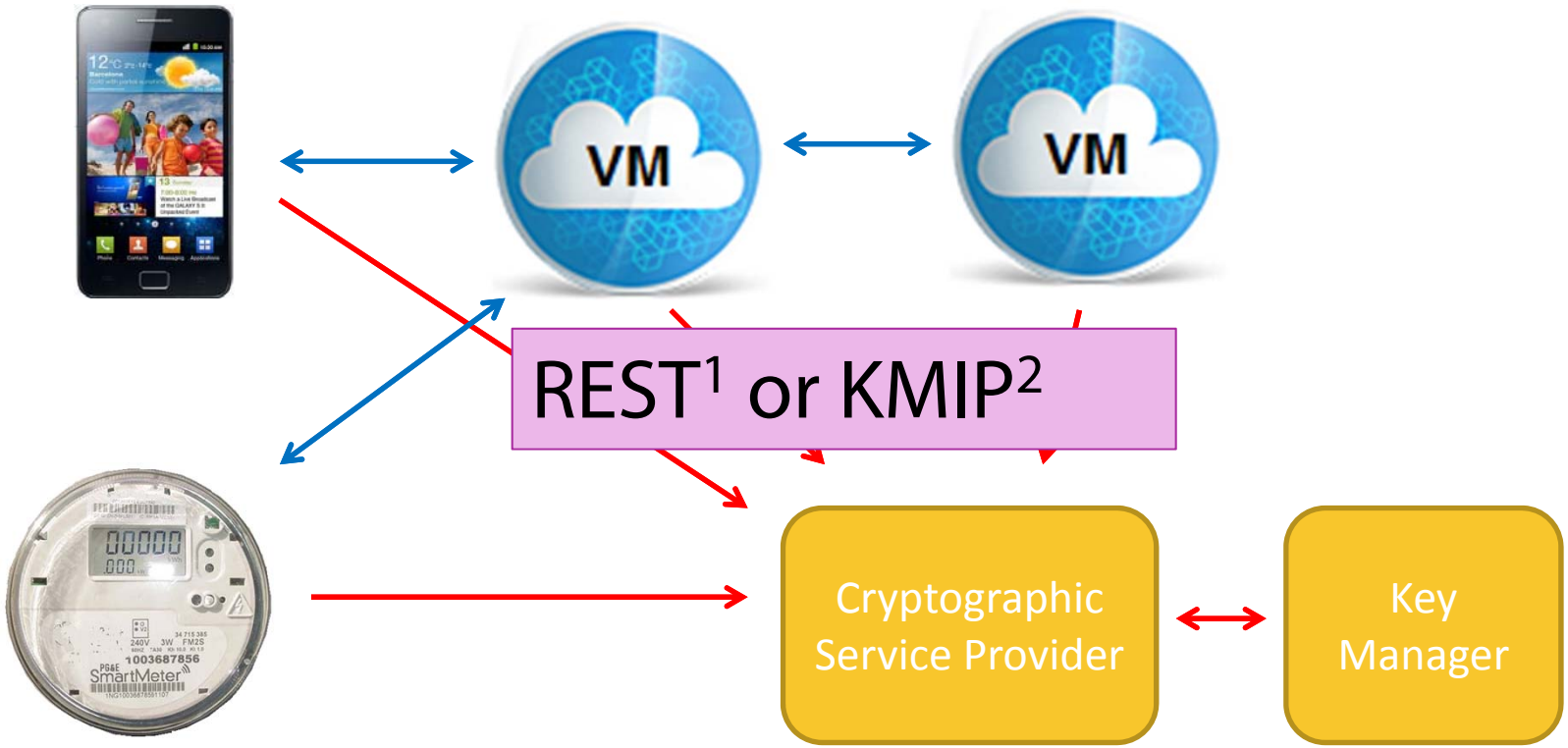
End Point Cryptography



Cryptography as a Service



Cryptography as a Service



1. Representational State Transfer (REST). Often over mutually authenticated TLS.
2. Key Management Interoperability Protocol (KMIP). Uses mutually authenticated TLS.

End Point Authentication

#RSAC

RSA CONFERENCE EUROPE 2013

Authentication

- End Point Authentication ensures only authorized end points can use the CaaS services.

Auth: Smart Phone App

- User:
  - Password.
  - Voice print.
  - Facial recognition.
  - Motion based.
  - SecurId One Time Password.
- Environment:
  - Device: Device ID, SIM number, Phone number.
  - Location.
  - Apps allowed to be on phone.
  - Time of day.

Auth: Smart Phone App

- Risk based authentication could be applicable:
  - Services available depend on the degree to which the identity is authenticated.
  - Alternatively, perhaps the level of authentication could be "stepped-up" if the requested service requires a higher degree of authentication than has been provided.

Auth: Virtual Machine

- Mutually authenticated TLS:
  - TLS Client private key must be stored on VM and protected.
- Server authenticated TLS with a password or PIN[1]:
  - Password must be stored on VM and protected.

1. Personal Identification Number. In this context, a very large pseudo-random number.

Auth: Virtual Machine

- Environment:
  - System values: Complex due to virtualization layer, and technologies such as vMotion.
    - Can be used to encrypt local secrets.
  - Software attestation / environmental comparison.
  - Virtual Trusted Platform Module (vTPM).
  - Local authentication: Use techniques to determine if VM is "local" to a service provider.
  - Operational heuristics.

Auth: Smart Grid

- System values:
  - Device unique identifier.
  - MAC address.
- User:
  - Installation PIN.

Crypto Algorithms and Security Operations

#RSAC

RSA CONFERENCE EUROPE 2013

Cryptographic Algorithms

- This section explores which types of security operations make sense in a CaaS context.
- The impact of compromise of the end point is also reviewed.

Cryptographic Algorithms

- Private key operations:
  - Asymmetric signing.
  - Asymmetric decryption.
  - Key agreement.

- Compromise: Private key not exposed. However, private key could be used.

Cryptographic Algorithms

- Secret key operations:
  - MACing.
  - Symmetric encryption / decryption.

- Compromise: Secret key not exposed. However, secret key could be used.

Cryptographic Algorithms

- Entropy and Pseudo-random Numbers.
- Usage: Entropy from CaaS can be mixed into an end point's entropy to dramatically improve the quality of entropy on the end point. This will greatly improve the quality of end point produced keys.
- Compromise: None.

Cryptographic Algorithms

- Public key operations:
  - Asymmetric signature verification.
  - Asymmetric encryption.
- Usage: Crypto acceleration by off-loading processing?
- Compromise: Public key not exposed. However, public key could be used.

Cryptographic Algorithms

- Message Digesting.

- Usage:
  - To connect securely to a CaaS server, operations such as Message Digesting are required. As such, if the cryptographic library used to connect to the CaaS server can not be trusted to Message Digest values correctly, then the result also can't be trusted.
  - Possible usage if the output of the Message Digesting operation can be directly supplied to another CaaS operation.

Cryptographic Algorithms

- Combining algorithms:
  - Advantageous to be able to combine multiple operations in the one web services call: Sign and encrypt for instance.
- CMS / PKCS #7:
  - Secure messaging which offers a range of standard encapsulation types, for instance signed and enveloped.

System Impact of Moving to CaaS

#RSAC

RSA CONFERENCE EUROPE 2013

System Impact: Advantages

- Improved security:
  - No important cryptographic keys on end points.
  - Important cryptographic keys stored and backed up in one place; in a key manager.
- Other advantages:
  - Possible improved performance due to scalable web services.

System Impact: Disadvantages

- Disadvantages:
  - Increased architectural complexity.
  - Latency due to web calls.
  - Possibly reduced performance due to overhead of making web calls.
  - Possibly more hardware required to host the CaaS provider.
  - Denial of Service: Causing a loss of connection between an end point and the CaaS provider results in the end point not being able to perform cryptographic operations.

System Impact: Challenges

- Authentication of the end point requesting cryptography services is the major challenge of CaaS.
- Network connectivity is required to allow the end point to have the CaaS provider perform the required action when required.
  - For some situations, such as Smart Grid, this may not be possible.

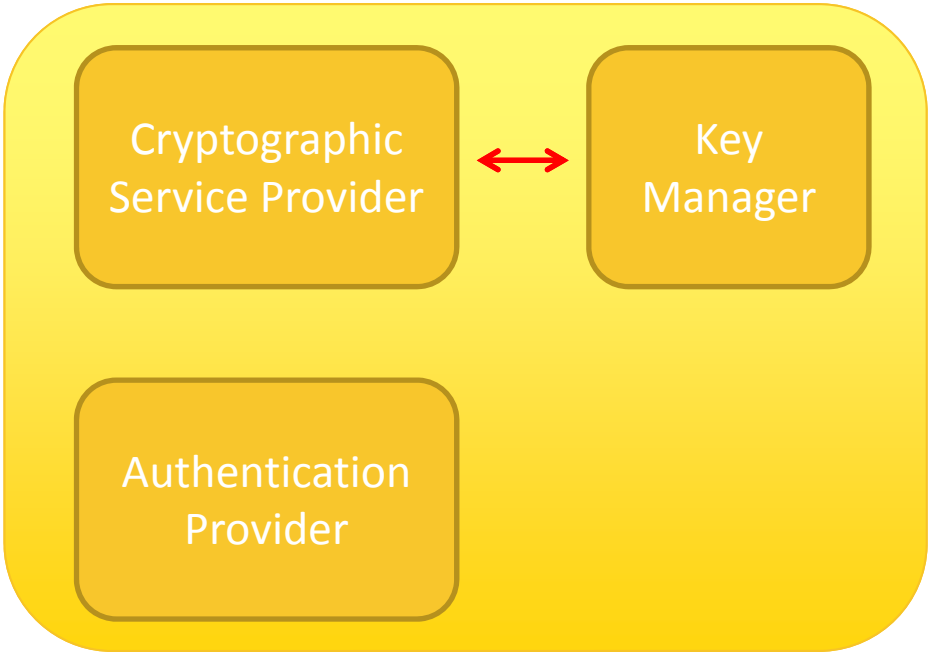
Usage

Usage: Smart Phone App

- Example: Sending a signed email.

Usage: Smart Phone App

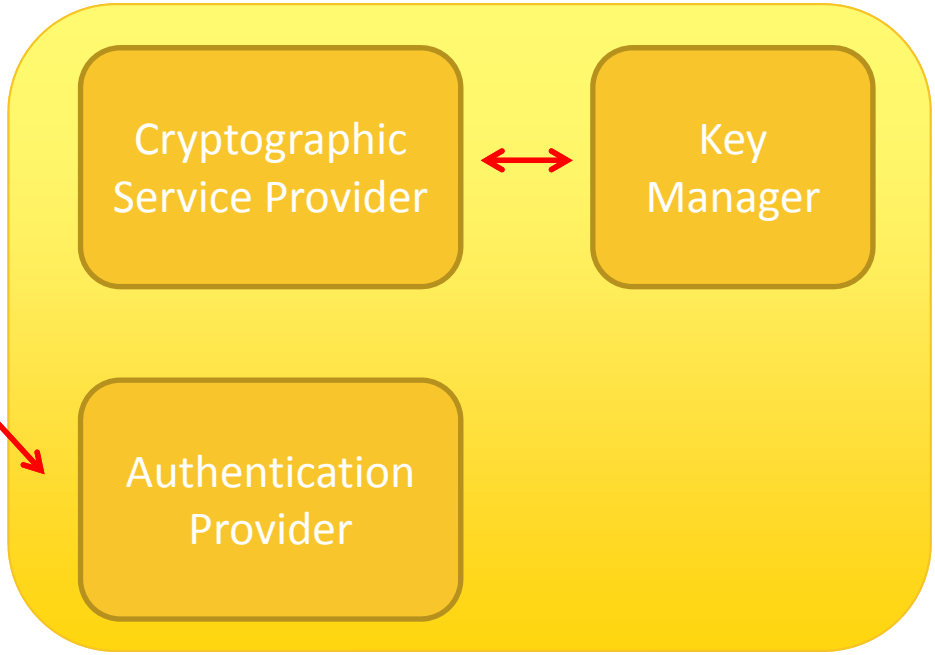
1. Login



Usage: Smart Phone App

1. Login

2. Authenticate

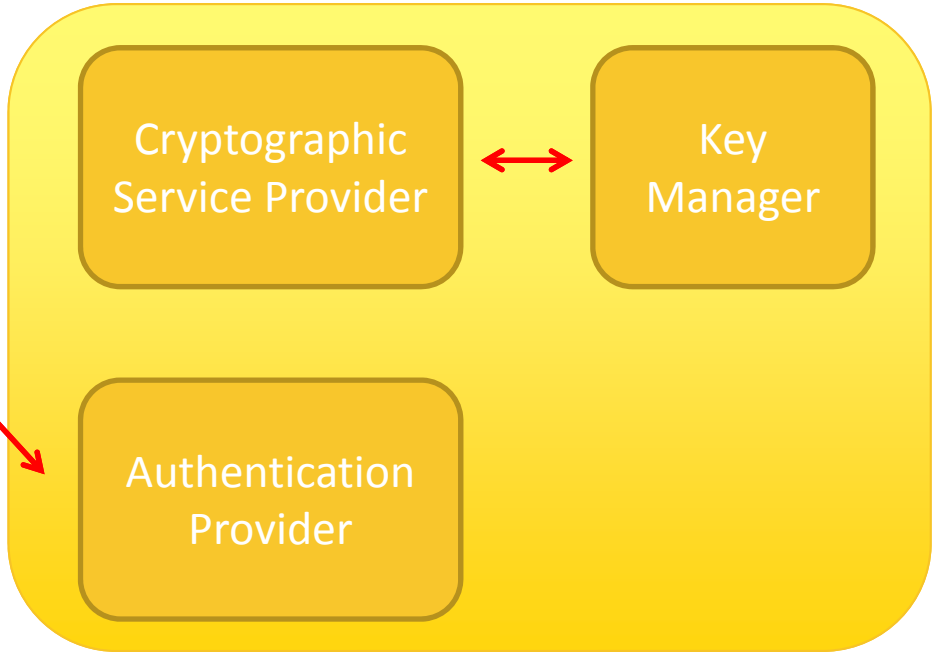


Usage: Smart Phone App

1. Login

2. Authenticate

3. Authentication Token

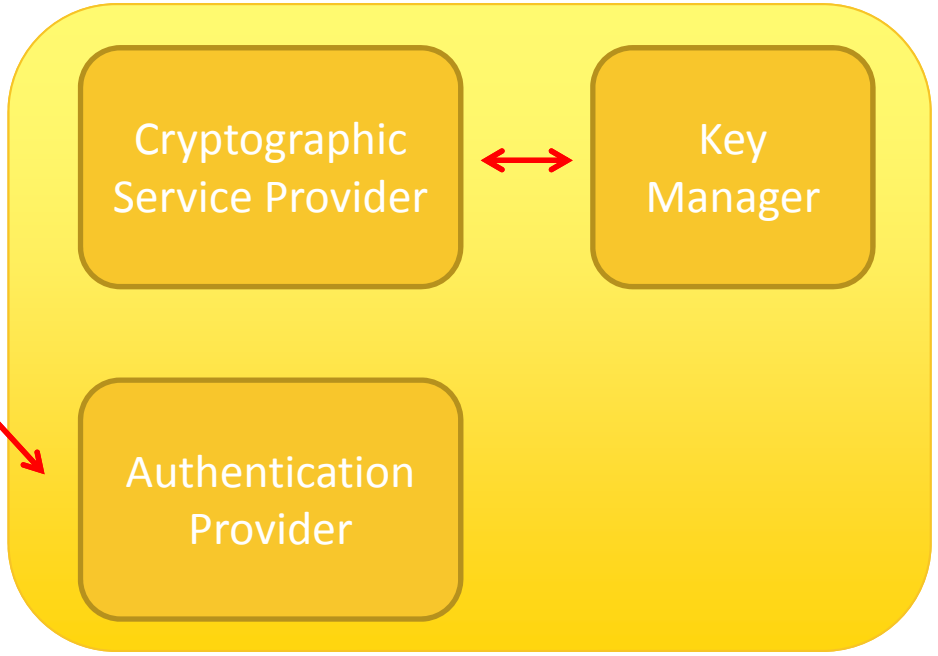


Usage: Smart Phone App

1. Login
4. Write email

2. Authenticate

3. Authentication Token



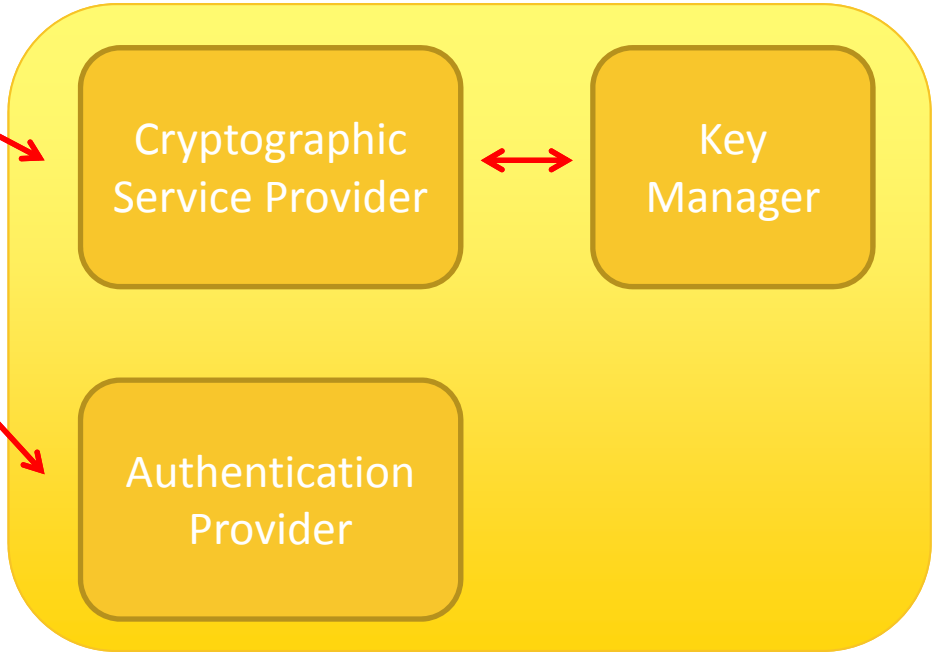
Usage: Smart Phone App

1. Login
4. Write email

5. Plain text email & Authentication Token

2. Authenticate

3. Authentication Token



Usage: Smart Phone App

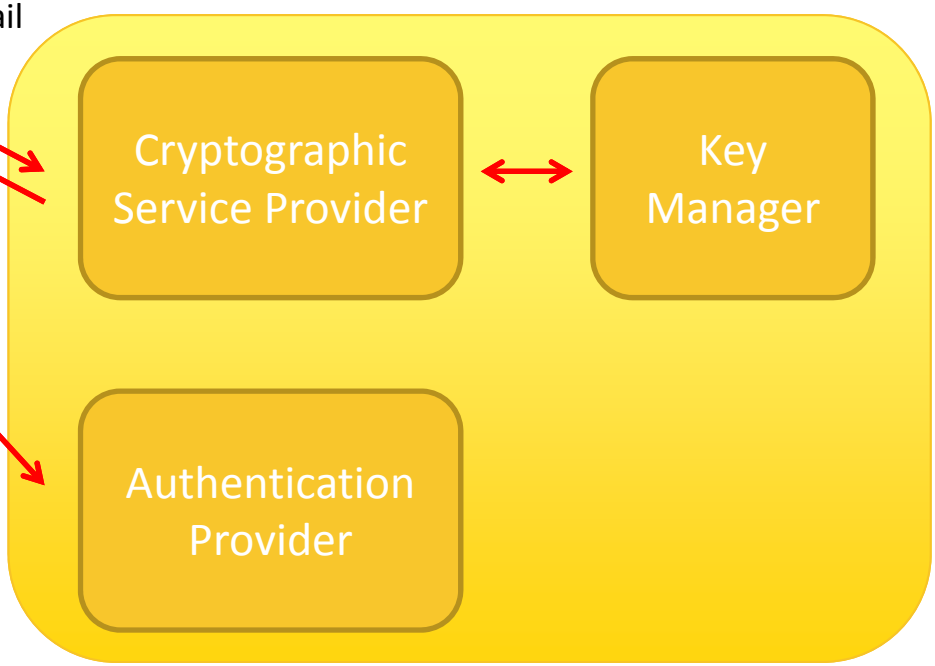
1. Login
4. Write email

5. Plain text email & Authentication Token

6. Signed email

2. Authenticate

3. Authentication Token



Usage: Smart Phone App

1. Login
4. Write email

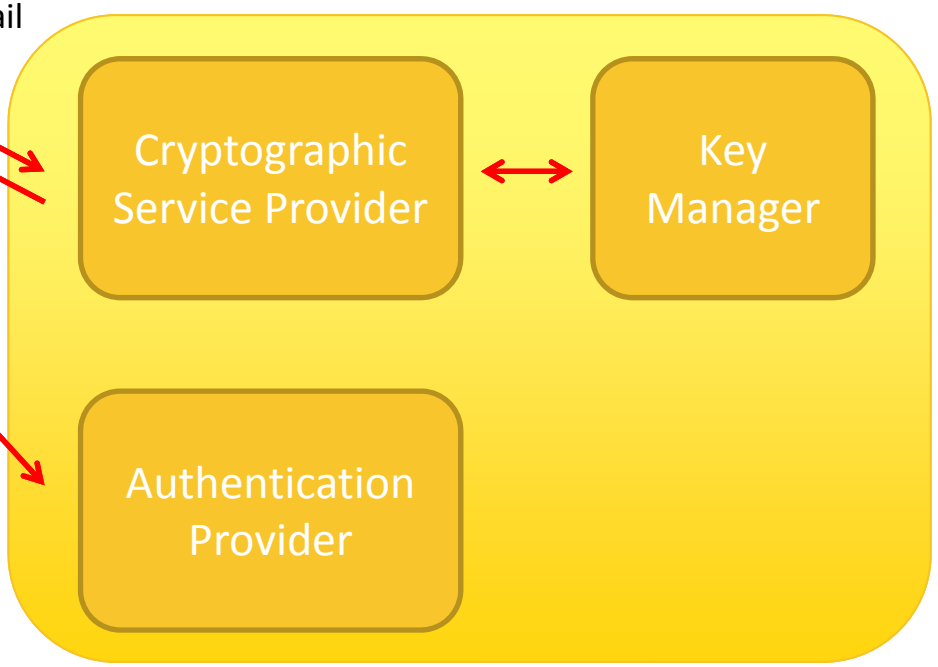
7. Send signed email to email server.

5. Plain text email & Authentication Token

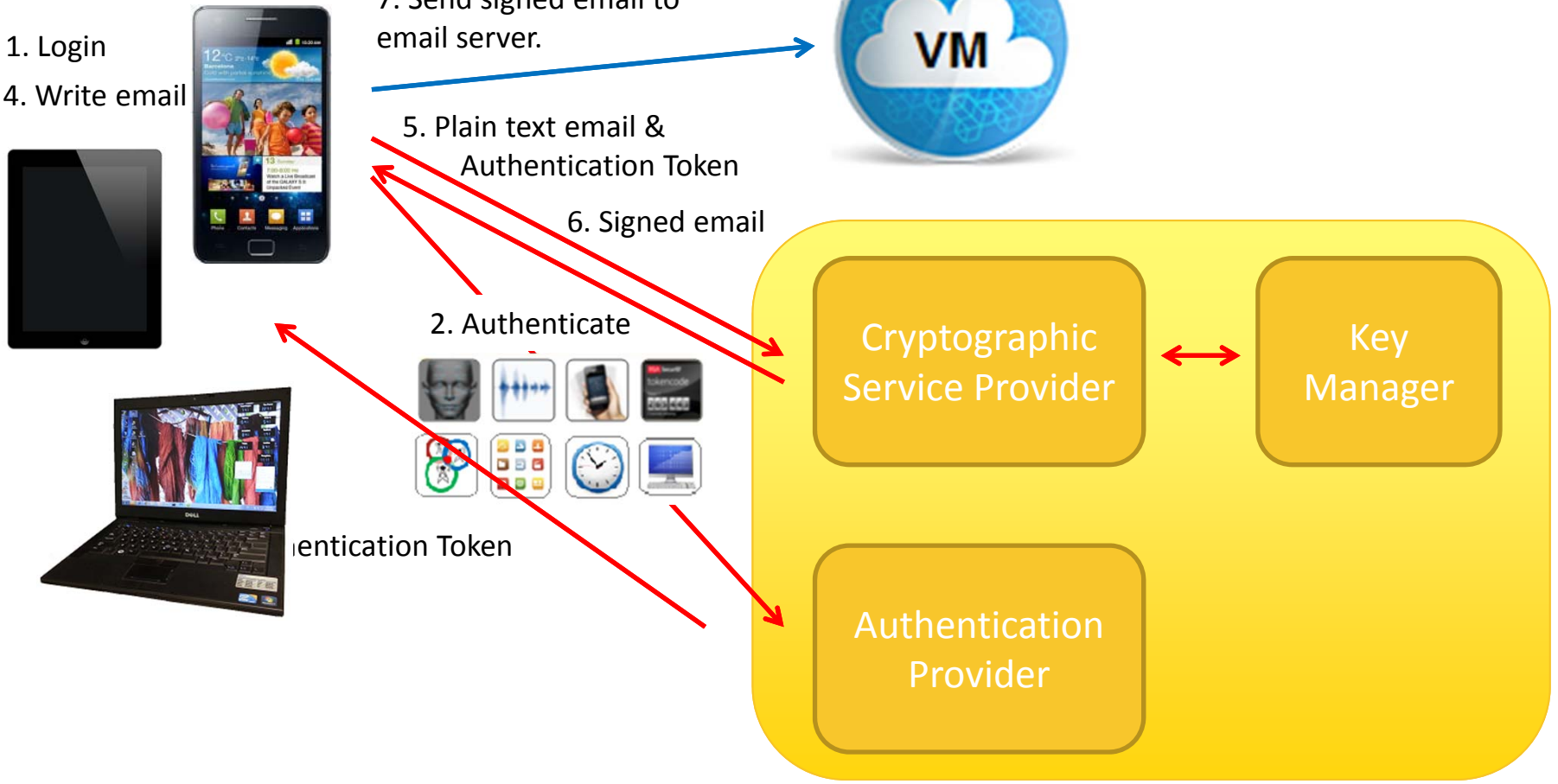
6. Signed email

2. Authenticate

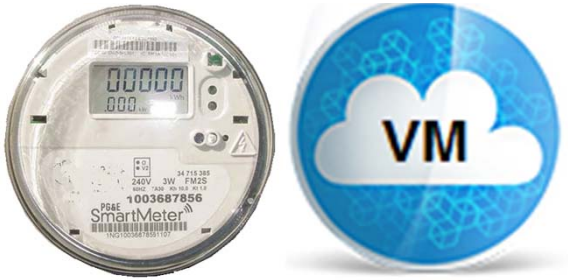
3. Authentication Token



Usage: Smart Phone App



Usage: Smart Grid & VM

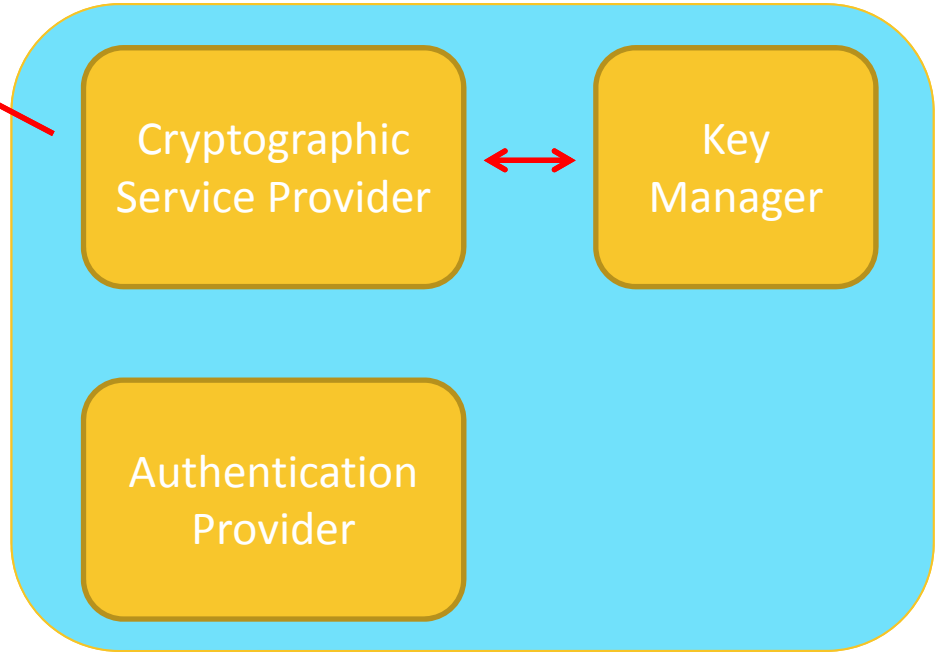


- Example: Smart Grid and VM

Usage: Smart Grid & VM

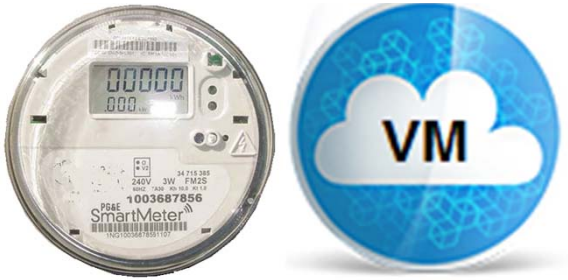
At time of manufacture:

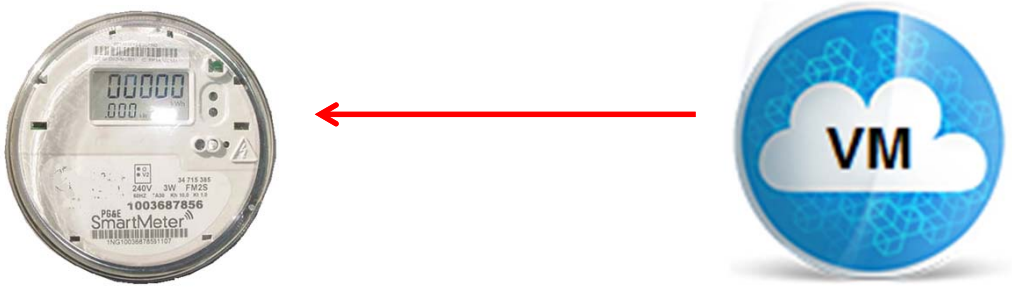
1. Smart Grid device manufacturer puts onto the device:
   a. Serial Number.
   b. Start-up entropy.
   c. Manufacturer's public key used for software update verification.



Manufacturer puts boot-strap information onto device.

Usage: Smart Grid & VM

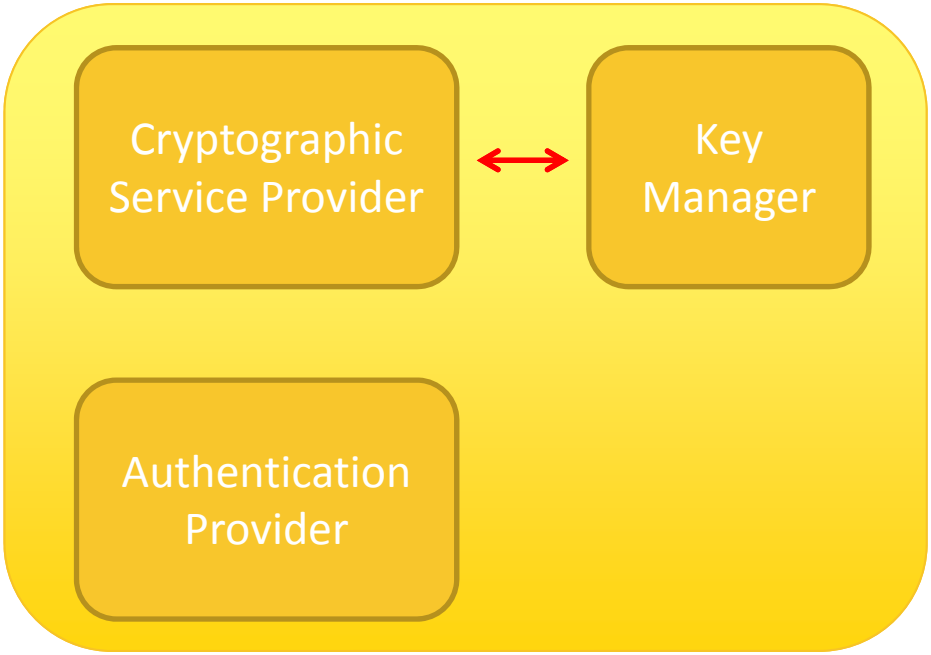




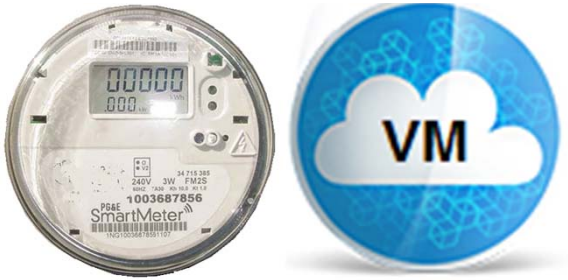
At installation time:

2. Device gets a software update signed by the Manufacturer's private key. The update contains:
   a. Utility's public key used for verifying control messages.
   b. Utility's public key used for decrypting messages from devices.

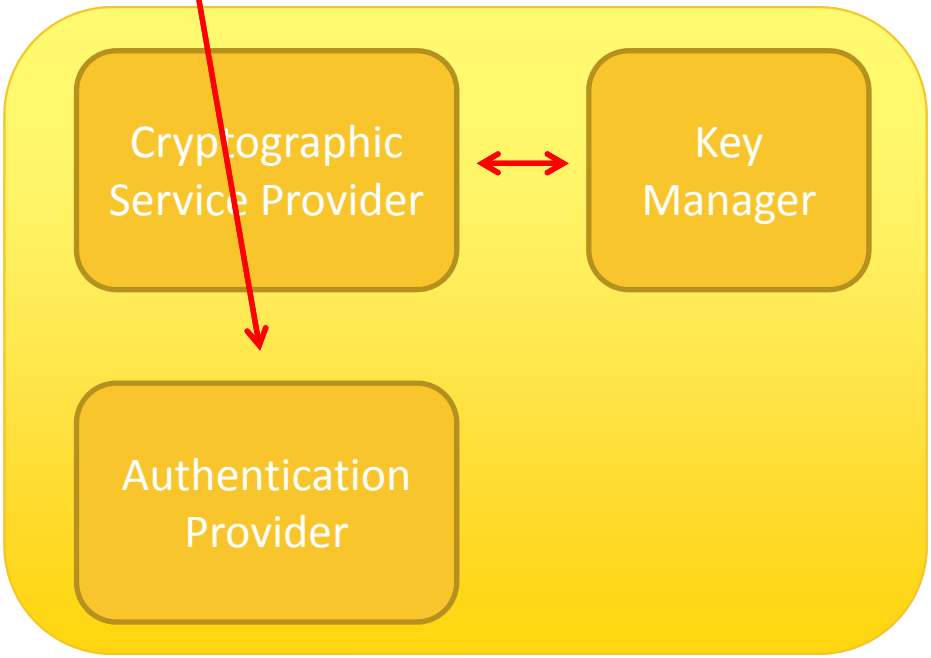
which allows Utility to put more boot-strap information on device.



Usage: Smart Grid & VM

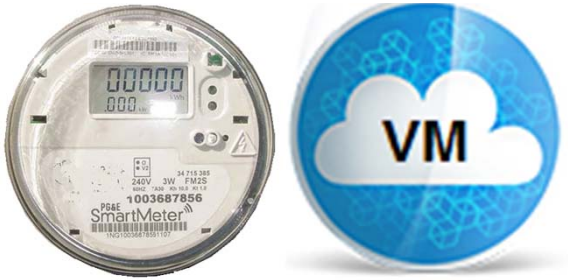


At installation time:
3. Device authenticates to server sending serial number and installation PIN encrypted using Utility's encryption public key.



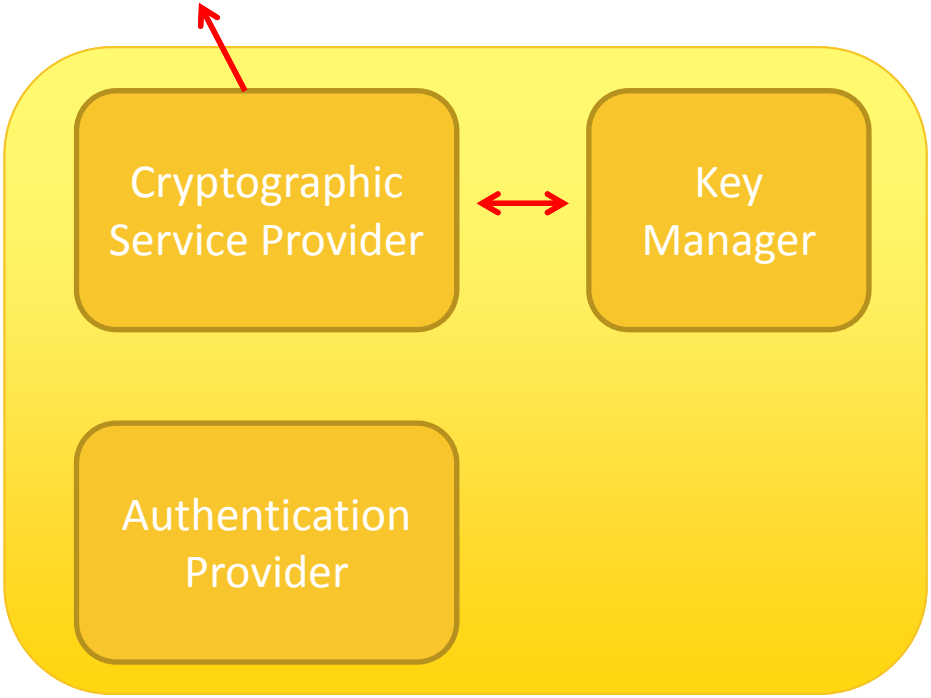
The device authenticates.

Usage: Smart Grid & VM

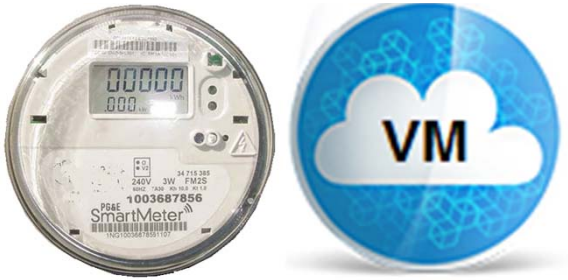


At installation time:
4. Device and server use serial number and installation PIN as an initial symmetric key.
5. Server generates an ephemeral device specific EC key pair for use in ECDH.
6. Server sends to the device, encrypted against the initial symmetric key and signed with the Utility's Control private key:
   a. Server's device specific ECDH public key.
   b. Entropy.

CaaS delivers entropy to device.

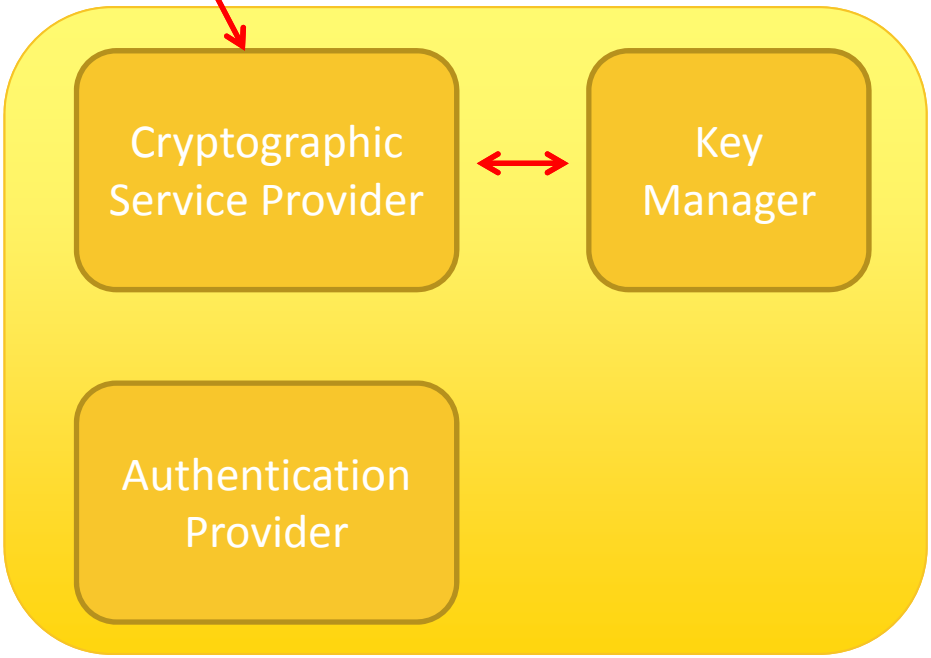


Usage: Smart Grid & VM



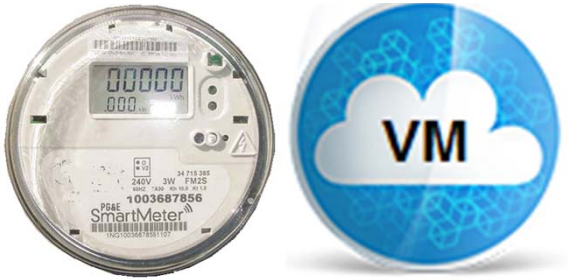
At installation time:

7. Device combines entropy from server and local entropy to generate an ephemeral EC key pair for use in ECDH, and an EC key pair for signing status messages.
8. Device uses ephemeral EC private key and server's device specific EC public key to derive an the device specific AES key.
9. Device sends to the server encrypted using server's encryption public key:
   a. Device's Status public key.
   b. Device's ephemeral EC public key.



CaaS and device complete ECDH key agreement.

Usage: Smart Grid & VM

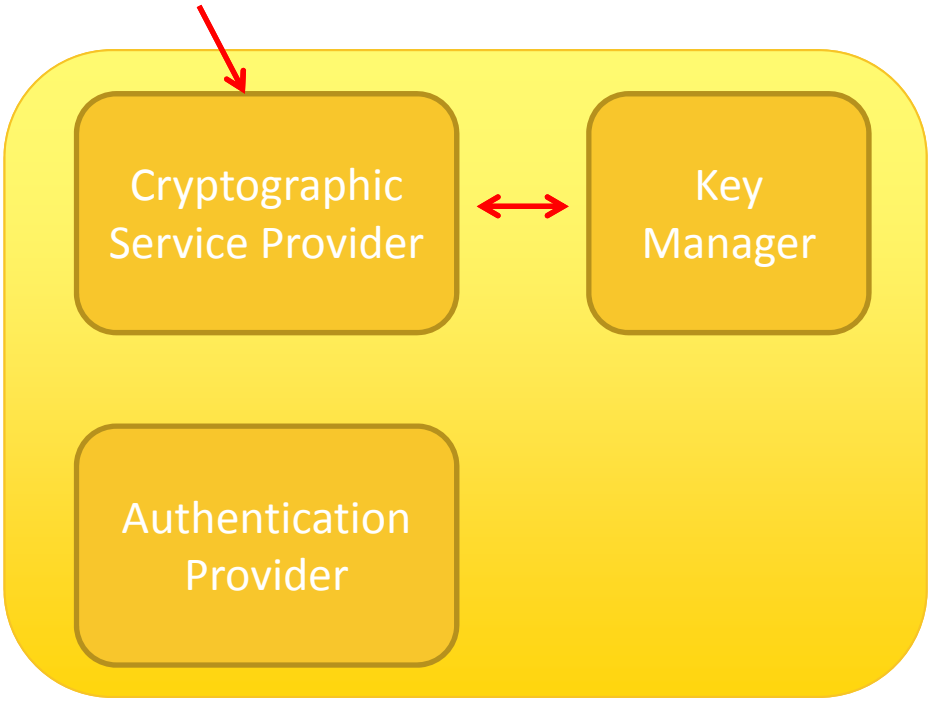


At installation time:

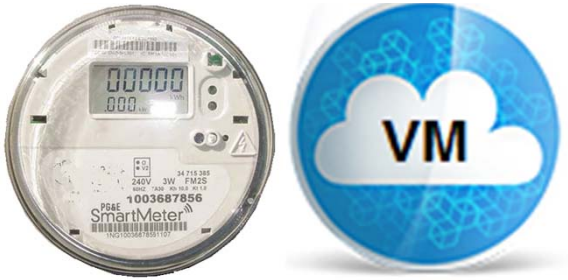
10. Server uses device's ephemeral EC public key and server's ephemeral EC private key to generate the device specific AES key.

11. Device now has the utility's public keys and the utility's servers have the device's public key. The device and the utility's servers share an AES key.

Device and server can communicate securely.



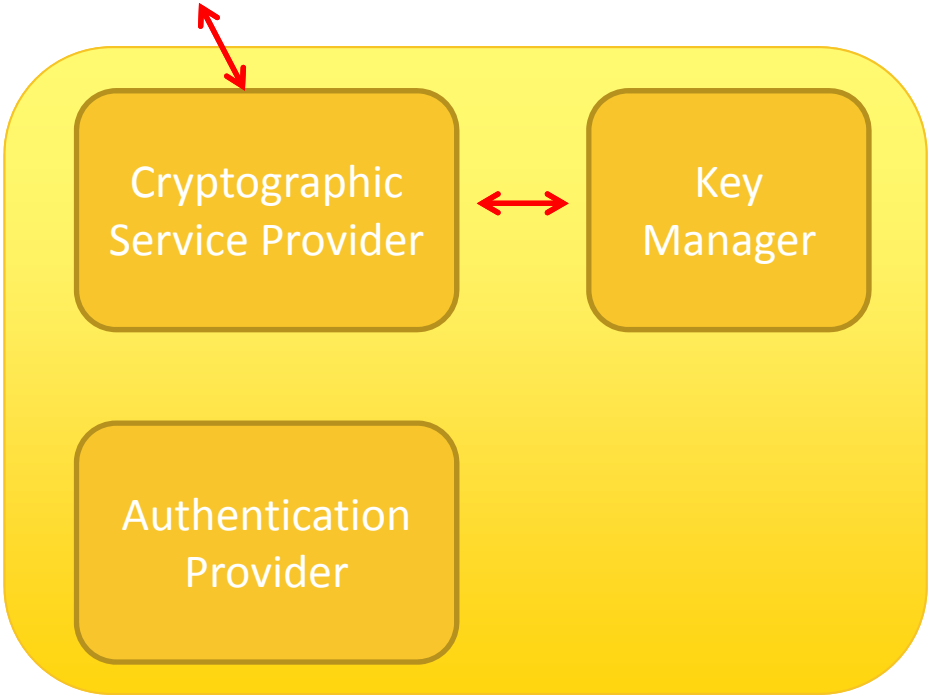
Usage: Smart Grid & VM



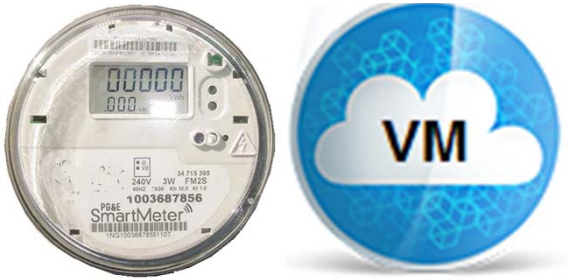
When operational:
1. Server to device: Signed and symmetric encrypted control messages.

2. Device to server: Signed and symmetric encryption status messages.

Device has keys.
CaaS has keys.
VM has no keys.



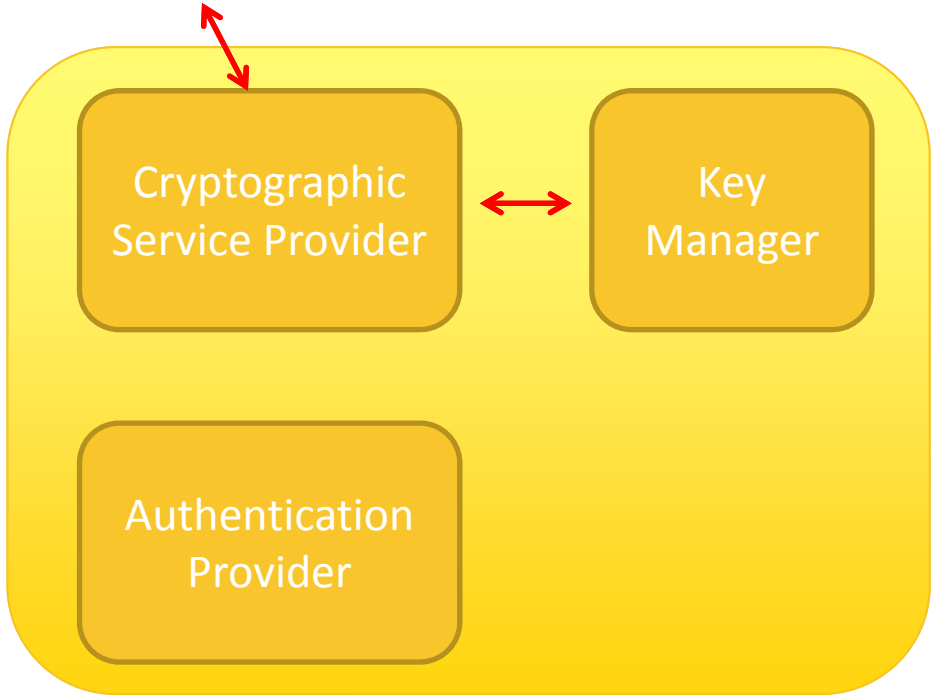
Usage: Smart Grid & VM



When operational:
1. Server to device: Signed and symmetric encrypted control messages.

2. Device to server: Signed and symmetric encryption status messages.

Device keys were generated in conjunction with CaaS.



Closing

#RSAC

RSA CONFERENCE EUROPE 2013

Apply

- Review what end points you have.
- For each end point:
  - Which keys are on the end point?
  - What cryptographic operations are performed and why?
  - What would be the cost of compromise of the keys be?
  - Which cryptographic operations would be suitable for a CaaS model?
  - What authentication mechanisms could be used?

Limitations of Technology

- CaaS does not apply to all scenarios:
  - No authentication mechanism.
  - No network connection.

Summary

- End points are bad places to generate and store keys.
- CaaS allows cryptographic services to be performed on behalf of end points without exposing important cryptographic keys to the end points.
- CaaS when combined with strong authentication can greatly improve the security of a system.
- CaaS does not apply to all cryptographic operations.
- Some environments, such as embedded devices, are challenging for CaaS. However, CaaS can still be used in these situations to improve the security.

Q & A

- Any Questions?

Thank you!

Peter Robinson

RSA, The Security Division of EMC

peter.robinson@rsa.com

www.rsa.com

#RSAC

RSA CONFERENCE EUROPE 2013