Southern Illinois University
CARBONDALE

Department of Computer Science
Southern Illinois University Carbondale

CS 491/531
SECURITY IN CYBER-PHYSICAL SYSTEMS

Lecture 22: Key Management in CPS

DR. ABDULLAH AYDEGER

LOCATION: ENGINEERING A 409F

EMAIL: AYDEGER@CS.SIU.EDU

Outline

Security Goals in CPS

CPS Key Management Principles

Verifying Keys

Recall: Privacy Issues in CPS

Privacy is about personal data

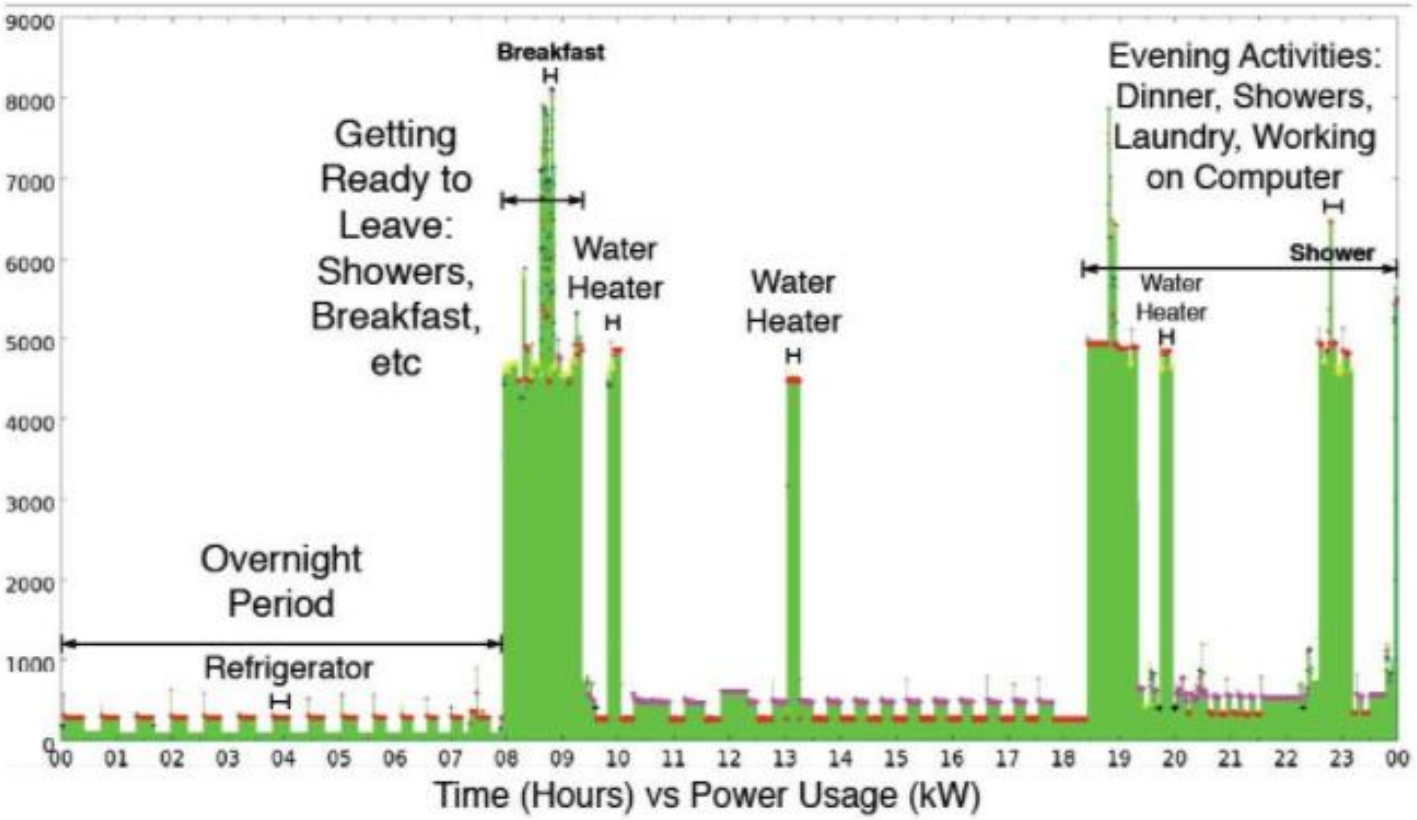
- How to collect, store and share it?
- Privacy vs Security

Some CPSes will collect data that relates to people -> This may raise privacy issues

- Smart Grid CPS;
  - Smart meter data – about people's power usage
- Vehicular CPS;
  - Electric Vehicles charge locations
- Smart Building CPS;
  - People's location for HVAC control purposes

Recall: Smart Meter's Power Data

Without detailed knowledge of appliance signatures, intuitive observation with power consumption variations indicates human activity



KEY MANAGEMENT IN CPS

Security Goals in CPS

Authentication: Includes two specific services, user authentication and data origin authentication

- User identity must be verified before a control command can be issued

Access control: Ensures only the authorized person can gain access to the CPS

Confidentiality: Protects the sensor measurements and control commands from passive attacks

- This is usually ensured by encryption

- Integrity: Prevents the CPS data being modified when data is in transit and in use
  - Integrity can be ensured by message authentication code, hash function, or digital signatures

- Nonrepudiation: Prevents either a sender from denying a transmitted message or a receiver from denying receipt of a message. Nonrepudiation can be ensured by digital signatures and handshakes.

- Availability: Ensures a CPS being accessible whenever users request them.

Security Goals in CPS

Integrity and authentication are critical to protect message security

If privacy and preventing eavesdropping are desired, confidentiality must be enforced

Furthermore, nonrepudiation should be adopted if accountability is required

Confidentiality, authentication, integrity, and nonrepudiation can be used to protect message security.

- These security services all require corresponding **keys established** between the communication entities that must exchange data

Secure Keys

A few keys used for message security include:

- Master keys: keys at the highest level in the hierarchy
- Message encryption key (data keys): Used for encryption to ensure the confidentiality of a message
- Message authentication code key: Used to create message authentication code to ensure the integrity of a message
- Message digital signature key: Used to create digital signature to ensure the nonrepudiation of a message
- Secure group communication key: Used for broadcasting a message system-wide or broadcasting a message in a dynamic group
- Key-encrypting keys: symmetric keys or encryption public keys used for key transport or storage of other keys
- Extra keys might be needed to distribute message encryption key, message authentication code key, and so on

Secure group communication in CPS

Forward secrecy: An embedded processor should not be able to read any future messages after it leaves a group

Backward secrecy: A newly joining member should not be able to read any previously transmitted messages in the group

CPS Key Management Design Principles

Scalability: Key management must be scalable to support all computation and networking components in a CPS

Freshness: Key management must ensure the freshness of keys and no adversary can replay old messages

Accountability: Key management needs to provide the required accountability to trace any anomaly in the system

Very challenging issue, due to:

- CPS’s heterogeneity, realtime availability, resilience to attacks, interoperability, and survivability

Heterogeneity

The integration of wireless sensing and actuating devices also poses more challenges to CPS security

Embedded processors have limited computation capability, memory capacity, transmission range, and energy consumption

- Complex cryptographic algorithms cannot be used
- Flash memory and RAM
    - Not enough space to run complicated algorithms after loading OS and application code
- If wireless sensing and actuating devices are used, the communication range of these components is limited

Heterogeneity

If sensors and actuators run on battery power, energy consumption must be considered

Energy consumption can be categorized into four parts:

- Energy for the sensor transducer
- Energy for moving or controlling a mechanism
- Energy for communication
- Energy for microprocessor computation

Key management protocols must consider these constraints

Resilience to Attacks

Ideally, a CPS should continue to provide real-time availability when under cyber-attacks

Key management is part of the CPS;

- Thus, key management must also be resilient to attacks

Key management often uses a key distribution center (KDC) to distribute keys and requires handshakes between communication entities

- Cybercriminals can launch DoS attacks targeting the KDC and the handshakes to degrade or disable key management services
- Key management protocols may decrease key exchange rate to reduce the exposure of secret credentials when under attack

Interoperability

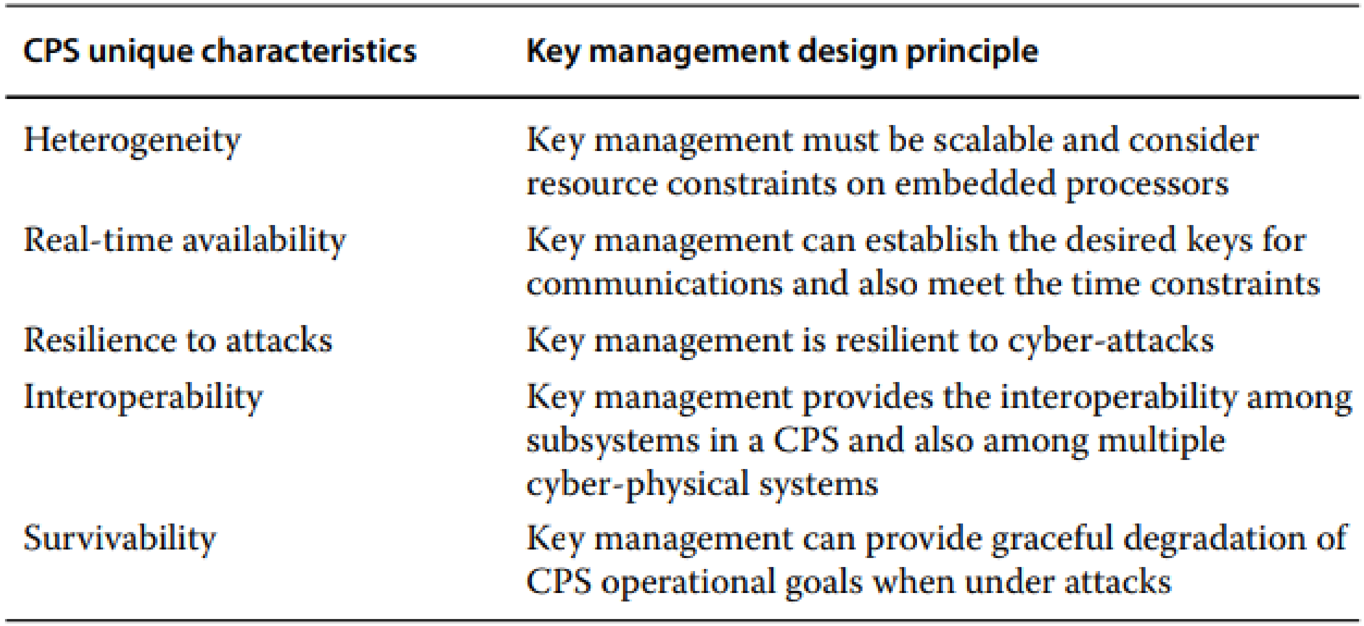
Key management and security mechanisms must be designed from top-down and enforced in each subsystem

- Otherwise, additional components such as gateways will be required to integrate the systems
- Adding additional components also increases the number of points of failure in the system

Standardization can be used to resolve interoperability issues in a CPS

- Public key infrastructure (PKI) has been widely used in the Internet to protect data security

CPS key management design principles



CPS unique characteristics	Key management design principle
Heterogeneity	Key management must be scalable and consider resource constraints on embedded processors
Real-time availability	Key management can establish the desired keys for communications and also meet the time constraints
Resilience to attacks	Key management is resilient to cyber-attacks
Interoperability	Key management provides the interoperability among subsystems in a CPS and also among multiple cyber-physical systems
Survivability	Key management can provide graceful degradation of CPS operational goals when under attacks

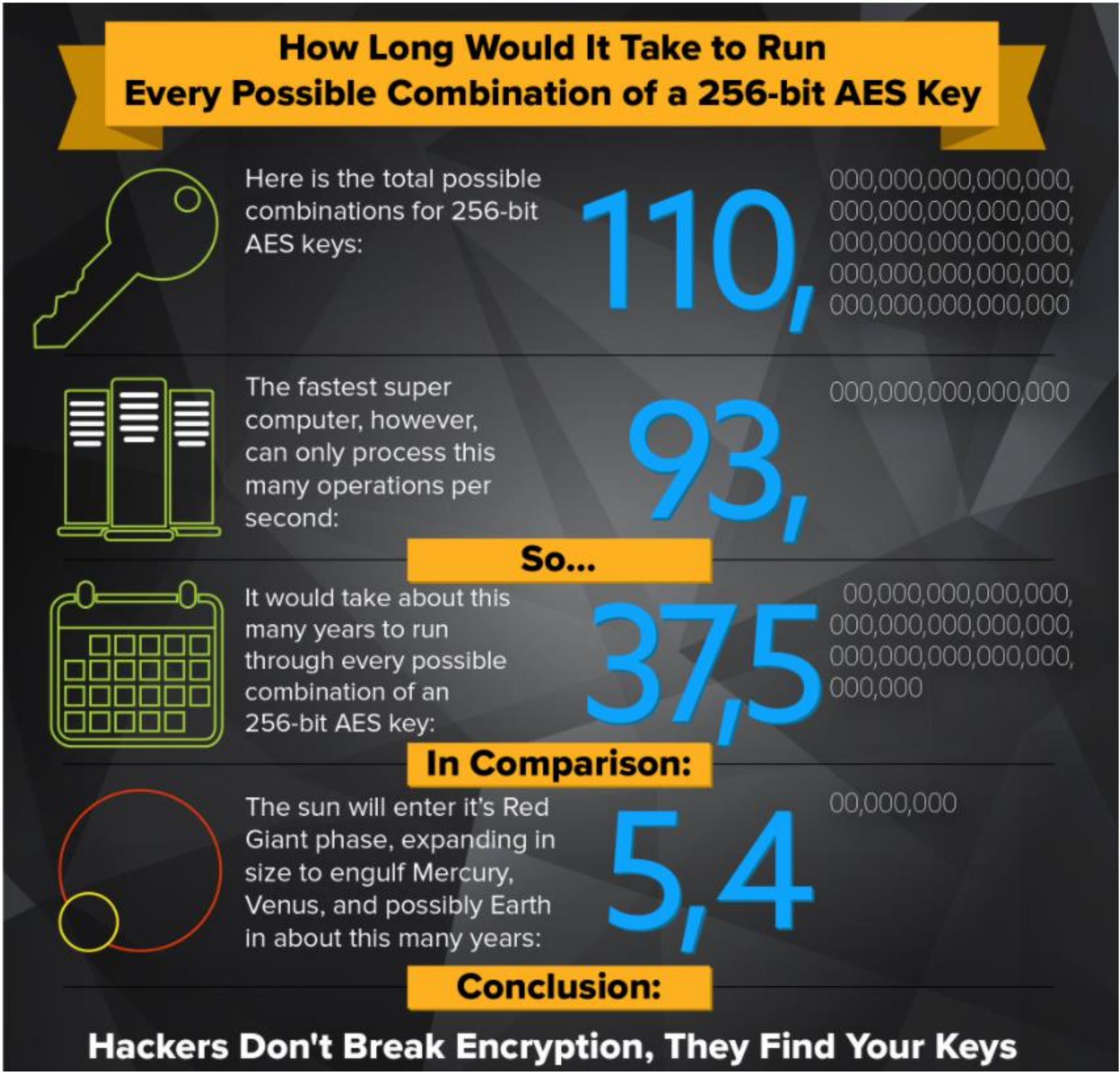
Why This Matters?

It is not simple to brute-force

Most attacks aimed at key management level rather than cryptographic algorithm itself

Deliver a key to two parties that need to communicate securely

- Delivery needs to be <u>secure</u>: only the two parties have access to the key



How Long Would It Take to Run Every Possible Combination of a 256-bit AES Key

Here is the total possible combinations for 256-bit AES keys:

110,
000,000,000,000,000,
000,000,000,000,000,
000,000,000,000,000,
000,000,000,000,000,
000,000,000,000,000

The fastest super computer, however, can only process this many operations per second:

93,
000,000,000,000,000

So...

It would take about this many years to run through every possible combination of an 256-bit AES key:

375
00,000,000,000,000,
000,000,000,000,000,
000,000,000,000,000,
000,000

In Comparison:

The sun will enter it's Red Giant phase, expanding in size to engulf Mercury, Venus, and possibly Earth in about this many years:

5,4
00,000,000

Conclusion:

Hackers Don't Break Encryption, They Find Your Keys

CPS Key Management

According to if handshakes are used in a protocol, key management can be divided into dynamic scheme and static scheme

According to the cryptography algorithms used, the protocols can be divided into public-key-based scheme and symmetric-key-based scheme

According to the network structure, the protocols can be divided into centralized scheme and distributed scheme

According to the probability of key sharing among communication entities, key management can be divided into probabilistic and deterministic schemes

Static key management schemes

Messages are protected by secret keys pre-distributed in each communication node

- No handshake among these communication nodes

Static schemes are easy and straightforward, however challenges:

- A single master key (MK) can be found in each communication node
  - A cybercriminal may crack a sensor node or an actuator to compromise the MK
- If the MK is compromised, there is no easy way to replace the MK
  - Shutting down a CPS and replacing the MK in each communication node might be impossible
- Has scalability issues

Dynamic key management schemes

Handshakes are often used to establish keys among communication entities

- Diffie–Hellman key exchange protocol or other public key cryptography algorithms
- Require fewer keys to be distributed in the system

Challenges:

- More complicated and need to be carefully designed
  - Cybercriminals may use defects in a dynamic approach to breach the security of a system

Public Key Cryptography

Public key algorithms, such as ***RSA*** are computationally intensive and usually execute thousands or even millions of multiplication instructions to perform a single security operation

- Symmetric key cryptography algorithms and hash functions consume much less computational energy than do public key algorithms
- Not feasible for most CPS devices

Symmetric Key Cryptography

Symmetric key cryptography is superior to public key cryptography in terms of speed and low energy cost

- However, the key distribution schemes based on symmetric key cryptography are not perfect

Centralized vs. Distributed key distribution scheme

Centralized key distribution scheme; One entity, which is often called a KDC, controlling the generation, regeneration, and distribution of keys

- Single point of failure
- The network may become too large to be managed by a single entity, thus affecting scalability

In the distributed key distribution approaches; Different controllers are used to manage key generation, regeneration, and distribution, minimizing the risk of failure and allowing for better scalability

- In this approach, more entities are allowed to fail before the whole network is affected

Key Management in SCADA

Initialization of the system and users

Generation, Distribution and Installation of keying material

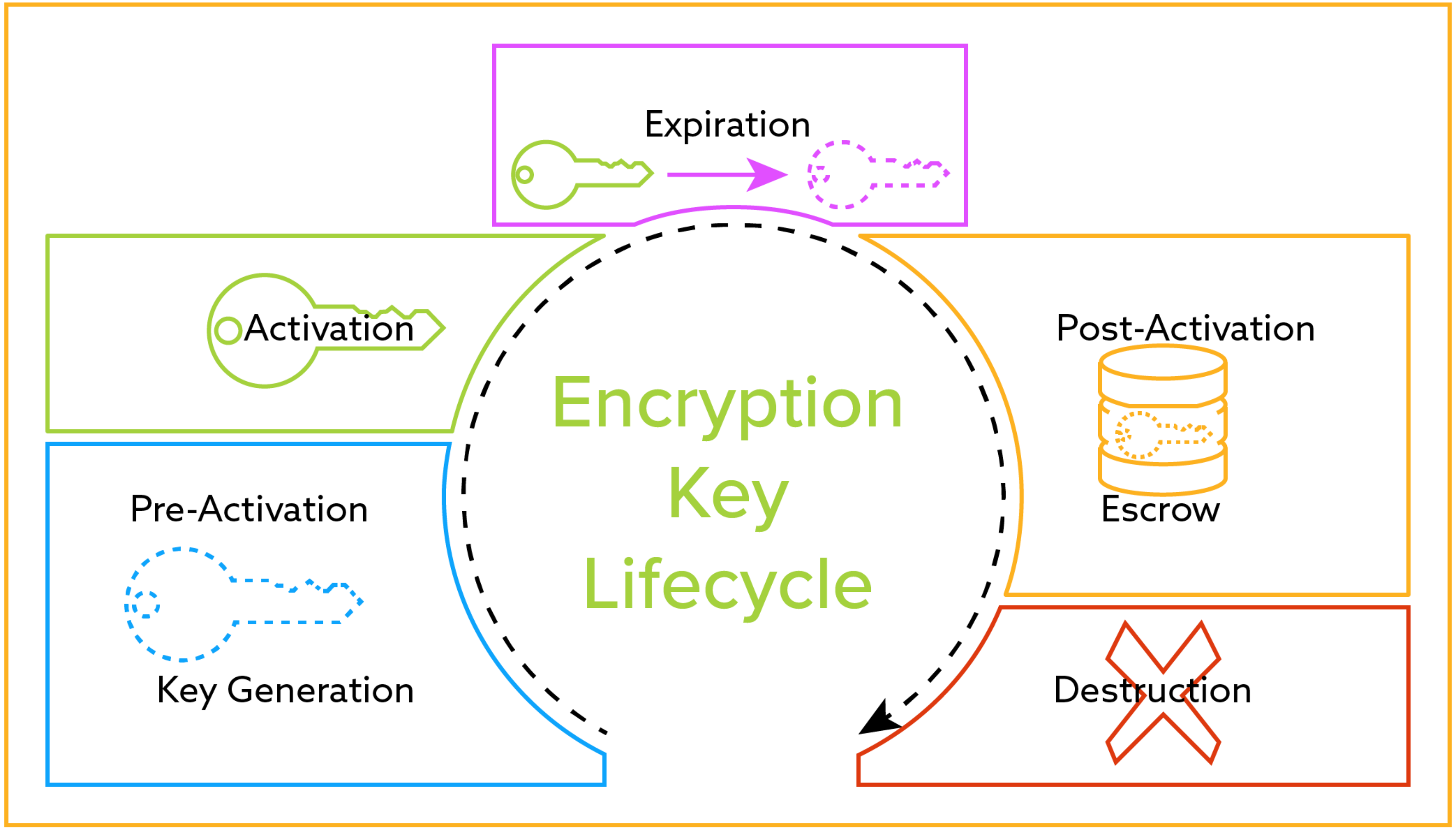
Oversight of operational use of keying material

Update, Revocation, and Destruction of keying material

Storage and Recovery of keying material

Effective key management means protecting the crypto keys from loss, corruption and unauthorised access

Key Management



Challenges to Key Management

Might require to be transmitted across multiple geographical locations

A common practice followed by many organizations is to store these keys separately in FIPS-certified Hardware Security Modules (HSMs) that are in-built with stringent access controls and robust audit trail mechanisms

- Not sufficient, as apart from secure storage, efficient management of the crypto keys at every phase of their lifecycle is very important

The correct methodologies to update system certificates and keys before they expire and dealing with proprietary issues when keeping a track of crypto updates on legacy systems

Key Management Policy (KMP)

Most organizations have comprehensive Information Security and Cybersecurity policies, very few have a documented Key Management Policy

Well-defined KMP firmly establishes a set of rules that cover the goals, responsibilities, and overall requirements for securing and managing crypto keys at an organisational level

It should protect key's

- Confidentiality
- Integrity
- Availability
- Source Authentication

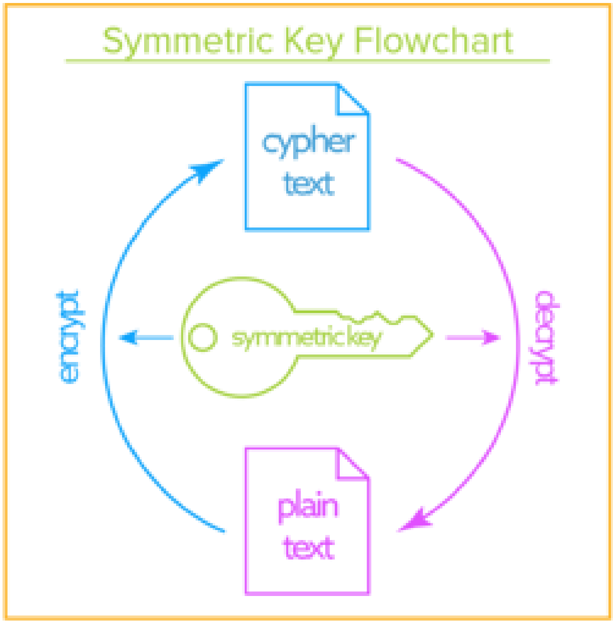
Key Types

Symmetric Keys: Data-at-Rest

The same encryption key is used to both encrypt and decrypt the data

This means of encryption is used primarily to protect data at rest

- An example would be to encrypt sensitive data into ciphertext while it is stored in a database and decrypt it to plaintext when it is accessed by an authorized user, and vice versa



Symmetric Key Distribution Options

1. A can select key and physically deliver to B

2. Third party can select & deliver key to A & B

3. If A & B have communicated previously
   - Use previous key to encrypt a new key

4. If A & B have secure communications with a third party C
   - C can relay key between A & B
   - C is Key Distribution Center (KDC)

Focus: option 4

Key Hierarchy

Two types of keys

- Session keys
  - Symmetric
  - Used for one logical session then discarded
- Master key
  - Used to encrypt session keys
  - Shared by user & key distribution center

Key Types

Asymmetric Keys: Data-in-Motion

Are a pair of keys for the encryption and decryption of the data

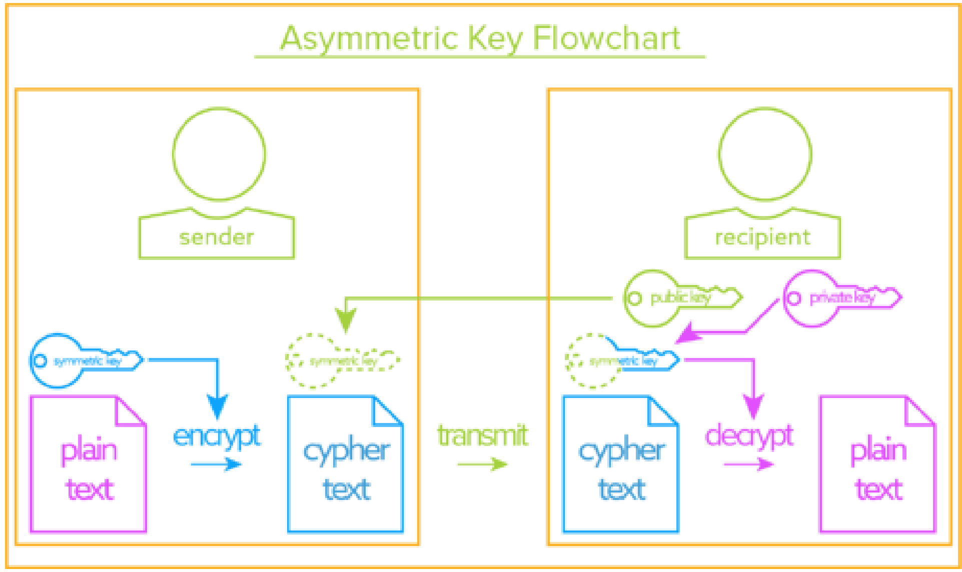
Both keys are related to each other and created at the same time. They are referred to as a public and a private key:

- Public Key: this key is primarily used to encrypt the data and can be freely given as it will be used to encrypt data, not decrypt it
- Private Key: this key is used to decrypt the data that it's counterpart, the public key, has encrypted. This key must be safeguarded as it is the only key that can decrypt the encrypted data

Asymmetric Keys

Asymmetric keys are primarily used to secure data-in-motion. An example might be a virtual private network (VPN) connection. With a VPN:

- An AES symmetric session key is used to encrypt the data
  - A public key is used to encrypt the session key
- Once the encrypted data is received, the private key is used to decrypt the session key
  - So that is can be used to decrypt the data



Verifying Keys

When Bob receives a key, how does he know it came from Alice and not from some-one pretending to be Alice?

Alice can use a digital signature protocol to sign the key

Bob has to trust public-key database to verify Alice's signature

KDC can sign Alice's public key

Bob has to trust KDC's public key he has

In this sense, some people argue that public-key cryptography is useless

Verifying Keys

Error detection during key transmission

- Send key as well as a known constant 2 to 4 bytes encrypted by the key
- At the receiving do the same to verify

Key-error detection during decryption

- For ASCII plaintext, decrypt and see whether you can read
- For random plaintext
  - Attach a verification block header, a known header encrypted by the key
  - Decrypt at the receiver to verify