CS578: Internet of Things

Security in IoT

Dr. Manas Khatua

Assistant Professor

Dept. of CSE, IIT Guwahati

E-mail: manaskhatua@iitg.ac.in

Introduction

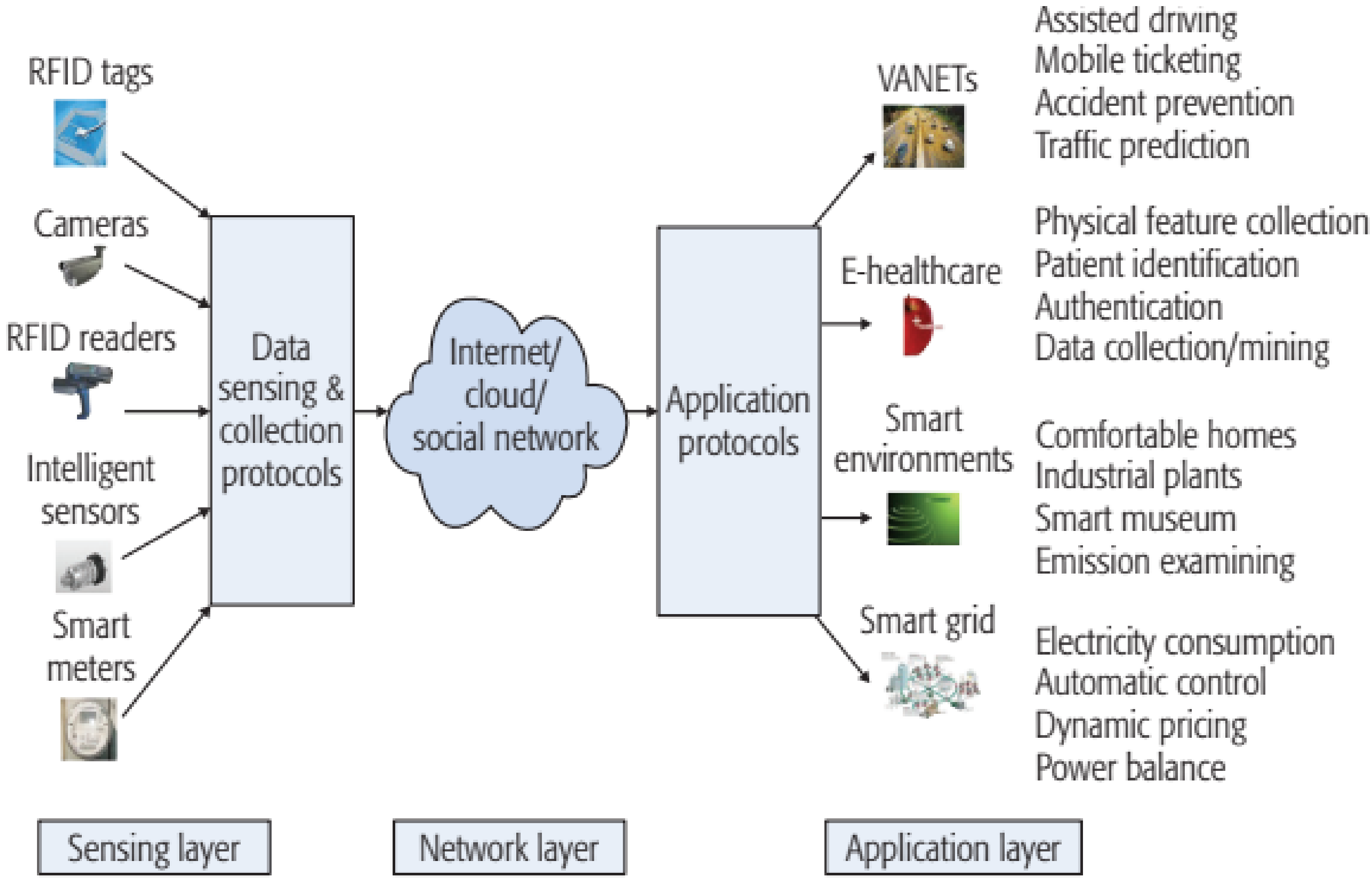
- IoT
  - uses low-power low-rate wireless communication protocol stack
  - millions of sensing and actuating devices are attached

- IoT security is the act of securing IoT devices and the networks they're connected to.

- Security solutions employed in TCP/IP are ill suited for IoT
  - Because of the constrained node and network

Cont…

Traditional Network architecture of cloud-based IoT Solutions.



Source: Zhou et. al., "Security and Privacy for Cloud-Based IoT: Challenges, Countermeasures, and Future Directions" IEEE Comm. Magazine, Jan. 2017, pp. 26-33.

Security Requirements

Fundamental Requirements:

- Confidentiality
  - refers to protecting information from being accessed by unauthorized parties
- Integrity
  - Data must not be changed in transit.
- Availability
  - is a guarantee of reliable access to the information by authorized people.
- Authentication
  - merely ensures that the individual is who he or she claims to be
- Non-repudiation
  - is the assurance that someone cannot deny the validity of something
- Resilience
  - is the ability to prepare for, respond to and recover from an attack.

Additional Requirements - Privacy ; Anonymity; Accountability; Trust

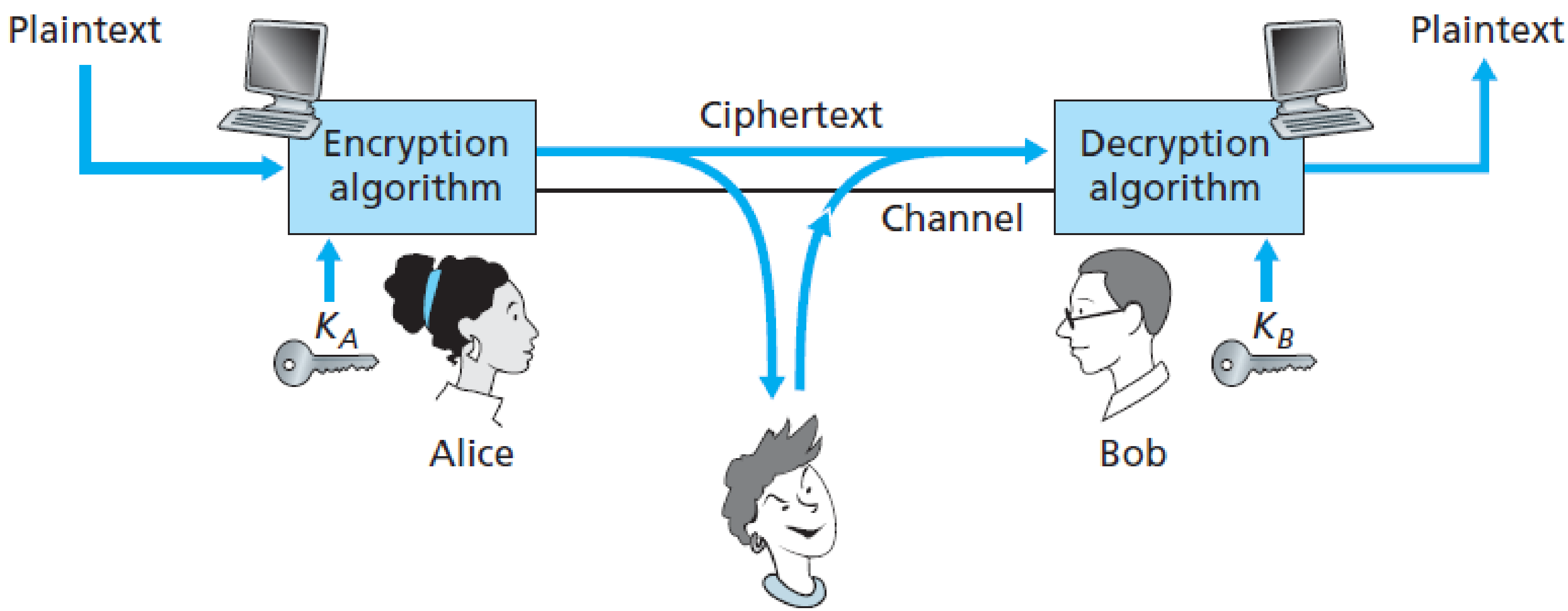
Common Security Attacks

- *Snooping :* unauthorized access to or interception of data.
- *Traffic Analysis :* obtain some other types of information by monitoring online traffic.
- *Modification :* modifies the information to make it beneficial to the attacker
- *Masquerading* or *spoofing* : the attacker impersonates somebody else.

- *Replaying :* the attacker obtains a copy of a message sent by a user and later tries to replay it.
- *Repudiation :* The sender of the message might later deny that she has sent the message; the receiver of the message might later deny that he has received the message.
- *Denial of Service :* It may slow down or totally interrupt the service of a system.

Principles of Cryptography

- It has a long history dating back at least as far as Julius Caesar.
- Cryptographic techniques allow a sender to disguise data so that an intruder can gain no information from the intercepted data.



Cont…

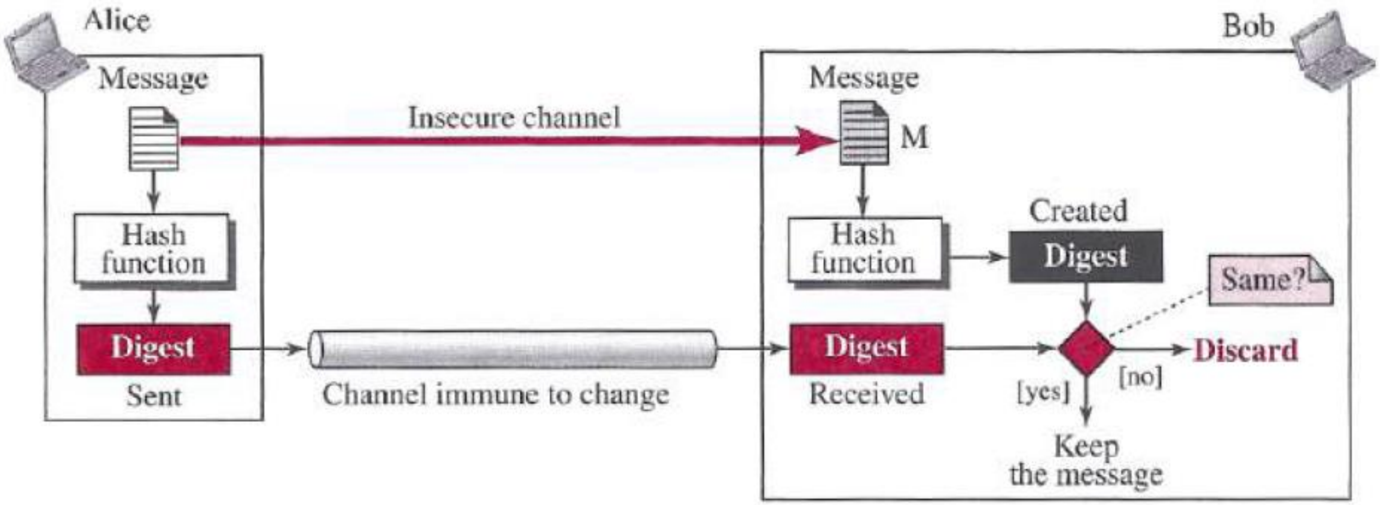
- To create the ciphertext from the plaintext, Alice uses an encryption algorithm and a key

- To create the plaintext from ciphertext, Bob uses a decryption algorithm and a key

- Based on type of key
  - symmetric key systems : Alice's and Bob's keys are identical and are secret
  - public key systems : a pair of keys is used. One of the keys is known to both Bob and Alice (indeed, it is known to the whole world). The other key is known only by either Bob or Alice (but not both).

Message Integrity

- In many instances, we must have integrity: the message should remain unchanged.
- One way to preserve the integrity of a document is through the use of a *fingerprint.*
- The electronic equivalent of the document and fingerprint pair is the *message* and *digest* pair.
- Message Digest is generated by a hash function



Hash Function

- Cryptographic Hash Function is required to have the following properties:
  - takes an input, *m*, and computes a fixed size string *H(m)* known as a hash
  - It is computationally infeasible to find any two different messages *x* and *y* such that *H(x) = H(y)*

- Popularly used Hash Functions:
  - MD5
  - SHA-1

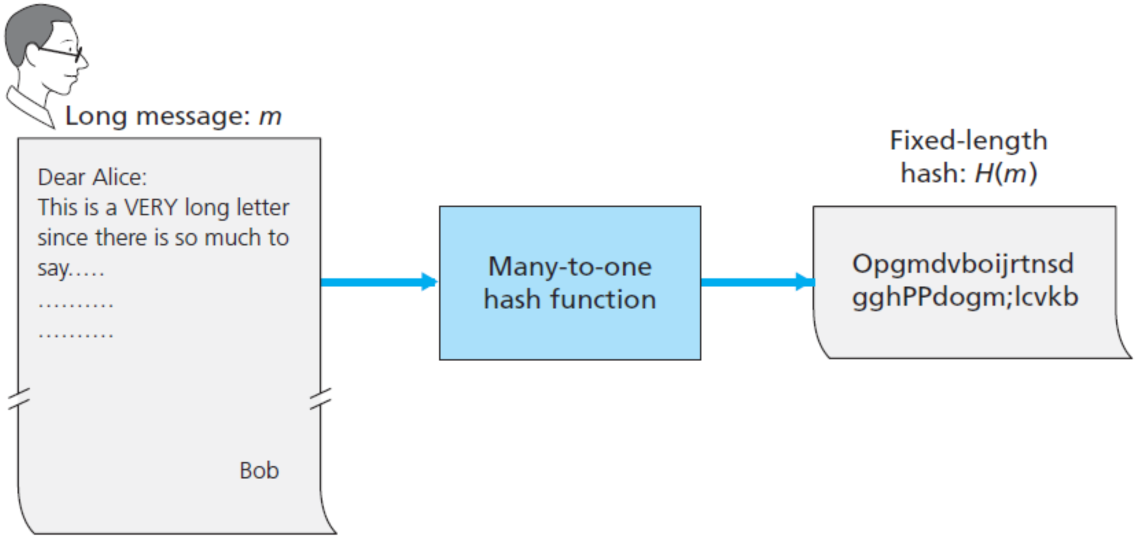


Figure 8.7 ♦ Hash functions

Message Authentication Code

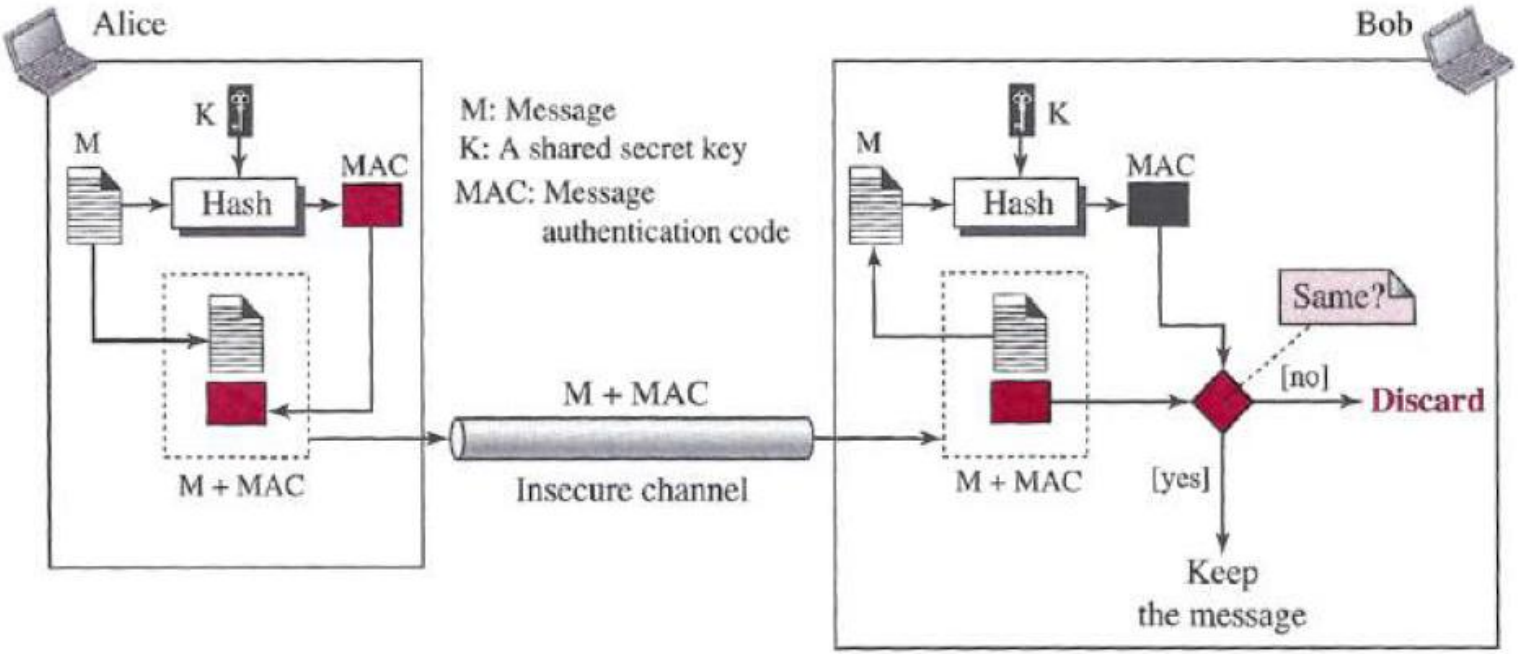
- To authenticate a message, Bob needs to verify:
  - The message indeed originated from Alice
  - The message was not tampered with on its way to Bob

- A digest can be used to check the integrity of a message.
-
- But, to ensure the integrity and authentication of the message, we need to include a secret shared between Alice and Bob in the process.

- This shared secret, which is nothing more than a string of bits, is called the **authentication key**.

- Message digest of message and authentication key is called Message authentication code (**MAC**).

Cont…



- One nice feature of a MAC is that it does not require an encryption algorithm
- Application:
  – In the link-state routing algorithm, communicating entities are only concerned with message integrity and are not concerned with message confidentiality

IoT PHY Layer (1/2)

- IEEE 802.15.4 **PHY** manages
  - Physical RF transceiver of the sensing device, Channel selection,
  - Energy management, and Signal management

- Standard supports multiple channels (e.g. 16 channels in the 2.4 GHz ISM radio band)
- Reliability is ensured at PHY by modulation techniques:
  - Direct Sequence Spread Spectrum (DSSS), Direct Sequence Ultra-Wideband (UWB), and Chirp Spread Spectrum (CSS).
  - They achieve reliability by occupying
    - more bandwidth at a lower power spectral density (in W/Hz) in order to achieve less interference along the frequency bands,
    - together with an improved Signal to Noise (SNR) ratio at the receiver.

- PHY data frames occupy at most 128 bytes
  - packets are *small* in order to minimize the probability of errors take place in low-energy wireless communication environments
    - <SYNC Header 5 byte> <PHY Header 1 byte><PHY Payload 127 byte>

IoT MAC Layer (2/2)

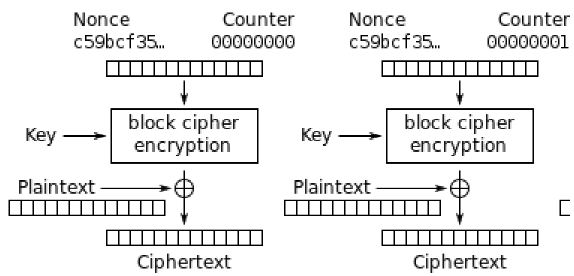
- IEEE 802.15.4 **MAC manages**
  - accesses to the physical channel
  - network beaconing
  - validation of frames
  - guaranteed time slots (GTS)
  - node association/joining
  - link level security
  - data services
- Types of devices based on the capabilities and roles in the network
  - FFD
  - RFD
- Types of frames
  - Data
  - ACK
  - Beacon
  - Command
    - Association request, Association response, Disassociation notification, Data request, Beacon request, GTS request, etc.
- IEEE 802.15.4 can support network topologies
  - peer-to-peer,
  - star,
  - cluster networks
- Devices are identified using
  - 16-bit short identifier
    - usually employed in restricted environments
  - 64-bit EUI-64
- Collisions during data communications are managed by CSMA/CA
- IEEE **802.15.4e** devices are
  - synchronize to a slot frame structure
  - Slotframe is a group of slots repeating over time
- Time synchronization is done using
  - acknowledgment-based, or
  - frame-based method

Security in IoT PHY and MAC

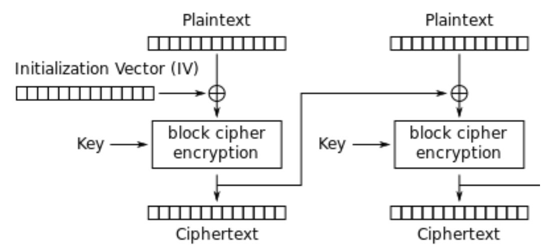
- *Security Suite in MAC:*
  - IEEE 802.15.4 standard support various *security suite* at the MAC layer,
  - *Security suites* are distinguished by
    - security guarantees provided, and
    - size of integrity data employed
  - Security suite includes security mechanisms defined for MAC frames

Name	Description
Null	No security
AES-CTR	Encryption only, CTR Mode
AES-CBC-**MAC**-128	128 bit **MAC**
AES-CBC-**MAC**-64	64 bit **MAC**
AES-CBC-**MAC**-32	32 bit **MAC**
AES-CCM-128	Encryption & 128 bit **MAC**
AES-CCM-64	Encryption & 64 bit **MAC**
AES-CCM-32	Encryption & 32 bit **MAC**

CTR Mode



CBC Mode



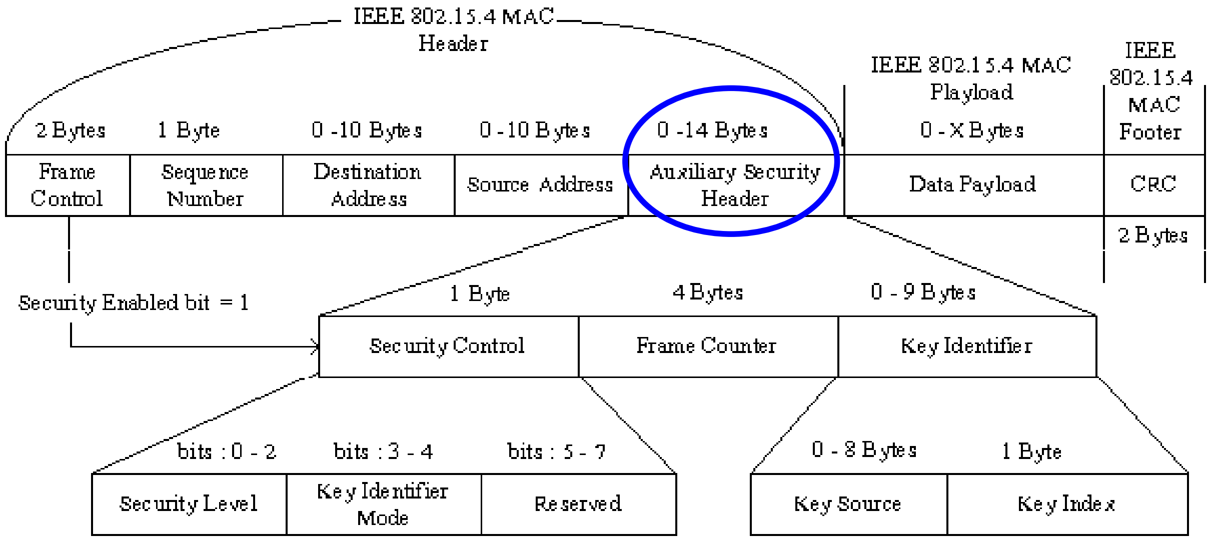
MAC: Message Authentication Code
CBC: Cypher Block Chaining
CTR: Counter Mode
CCM: Counter with CBC-MAC

Security suites

Security Suites	Achieved Security Requirements	Remarks
AES-CTR	Confidentiality	
AES-CBC-MAC-32, AES-CBC-MAC-64, AES-CBC-MAC-128	*Data Authenticity,* *Integrity,*	Message Authentication Code (MAC) or Message Integrity Code (MIC) is created from 802.15.4 MAC layer header plus the payload data
AES-CCM-32, AES-CCM-64, AES-CCM-128	*Confidentiality,* *Data Authenticity,* *Integrity*	

MAC: Message Authentication Code
CBC: Cypher Block Chaining
CTR: Counter Mode
CCM: Counter with CBC-MAC

Data Frame with Security Fields



- *Security Enabled Bit* field of the *Frame Control* field being set
- *Auxiliary Security Header* is employed only when security is used
  - ***Security Control*** field identifies the *Security Level* mode
  - ***Frame Counter*** used to ensure sequential freshness of frames sent by this device.

- Standard employs 128-bit keys that is known implicitly by the two communication parties,
- **OR** determined from information transported in the ***Key Identifier*** field.
  - *Key Source* subfield specifies the group key originator
  - *Key Index* subfield identifies a key from a specific source

Security-related information

- The various *security suites* require
  - transportation of security-related information in different configurations

	4B	1B	Variable
AES-CTR	Frame Counter	Key Control	Encrypted Payload

	Variable	32/64/128b
AES-CBC-MAC-X	Payload	MAC

	4B	1B	Variable	32/64/128b
AES-CCM-X	Frame Counter	Key Control	Encrypted Payload	Encrypted MAC

- Protection against message replay attack:
  - ✓ The *Frame Counter* and *Key Control* fields are used to achieve the protection in all security modes
  - ✓ *Frame Counter* sets the unique message ID by the sender
  - ✓ The *key counter* (Key Control field) is incremented if the maximum value for the *Frame Counter* is reached.
- To support semantic security and replay protection,
  - ✓ each block is encrypted using a different *nonce* or *Initialization Vector* (IV).

Cont…

- Initialization Vector (IV) can be
  - ✓ a pseudorandom number, OR
  - ✓ {4byte Frame Counter,
  - ✓ 1byte Key Counter,
  - ✓ a static 1byte Flags field,
  - ✓ 8byte sender's address,
  - ✓ 2byte Block Counter field}

  - ✓ *Block counter*: The sender breaks the original packet into 16-byte blocks, with each block identified by its own block counter.

Access Control Mechanisms:

✓ IEEE 802.15.4 MAC also provides access control functionalities

✓ The 802.15.4 radio chips of the device stores an access control lists (ACL) with a maximum of 255 entries (generally for 255 destination address)

✓ Each ACL entry stores
- an IEEE 802.15.4 ***address***,
- a ***Security Suite*** *identifier* field
- the security material required to process security
  - cryptographic ***Key*** for suites supporting encryption,
  - the *Nonce* (**IV**),
  - the most recently received packet's identifier in the ***Replay Counter*** field

Address	Security Suite	Key	Last IV	Replay Counter

Security with TSCH

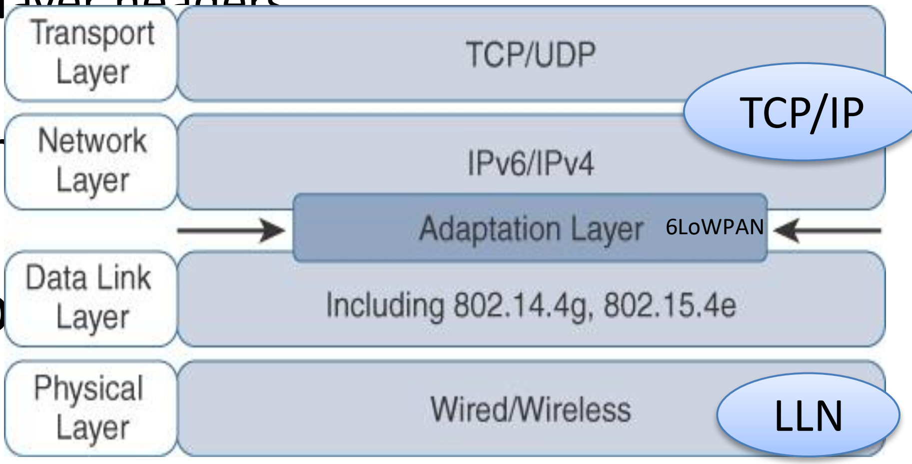
- IEEE 802.15.4 - 2011 provides security services at the MAC layer
- IEEE 802.14.4e - 2012 introduces few modifications in security services corresponding to different modes

- Addendum defines the possibility of using **null** and **4 or 5-byte** *Frame Counter* values
  - shall be set to the global Absolute Slot Number (ASN) of the network
  - usage of the ASN as a global frame counter value enables
    - time-dependent security,
    - replay protection and
    - semantic security.
- Employs two bits from the reserved space of *security control* filed:
  - bit 5 to enable suppression of the *Frame Counter* field i.e. null
  - bit 6 to distinguish between a *Frame Counter* field occupying 4 or 5 bytes

- In consequence, the *Auxiliary Security Header* in MAC frame may now transport a null, a 4-byte or a 5-byte *Frame Counter* field.
- **Open Issues:**
  - Key management;
  - ACL entry for group keying or network shared keying;
  - Protection of ACK message;
  - etc.

Adaptation Layer Protocol

- 6LoWPAN is currently a key technology to support Internet communications in the IoT
- It defines:
  - Header compression
    - Compress the NET and TRN layer headers
  - Neighbor discovery
  - Address auto-configuratior



- It supports four header typ
  - No 6LoWPAN
  - Dispatch
  - Mesh Addressing
  - Fragmentation

Security in 6LoWPAN

- No security mechanisms are currently defined in the context of the 6LoWPAN adaptation layer

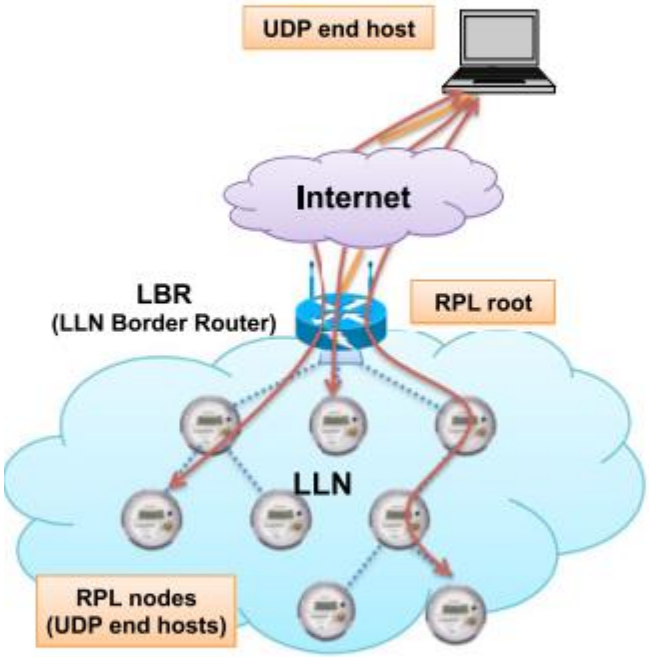
- but many relevant documents include discussions on the security vulnerabilities, requirements and approaches to consider for the usage of network layer security

  - ✓ RFC 6606: identifies the importance of addressing security and the usefulness of AES/CCM available at the hardware of IEEE 802.15.4 sensing platforms.

  - ✓ RFC 3971: SEcure Neighbor Discovery (SEND)

  - ✓ RFC 3972: Cryptographically Generated Addresses

RPL



RPL protocol supports many **control messages**

- DIO (DODAG Information Object)
- DIS (DODAG Information Solicitation)
- DAO (Destination Advertisement Object)
- DAO-ACK (DAO acknowledgment)
- CC (Consistency Check)
- Etc.

- RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) **topology**
  - ✓ Each DODAG starting from root has DODAGID
  - ✓ Parameters used: link costs, node attributes (e.g. rank), node status information
  - ✓ Function: objective function.

Security in RPL

- RPL specification defines
  - secure versions of the various routing control message exchange
  - three **security modes**
    - Unsecured
    - Preinstalled
      - using a pre-configured symmetric key
      - Achieve: Confidentiality, Integrity, Data authentication
    - Authenticated
      - A device may initially join the network using a pre-configured key, and then obtain a different key from the key authority with which it may start functioning as a router.

1b	7b	1B	2b	3b	3b	1B
T	Reserved	Algorithm	KIM	Resvd	LVL	Flags
Counter						
Key Identifier						

Fig: **Security fields** after the 4-byte ICMP header for a secure RPL control message

T: Timestamp
KIM: Key Identifier Mode
-- indicates Cryptographic key required
LVL: Security Level
Flags: Reserved Flag
Counter:
Key Identifier:

RFC 6550 defines the various values to identify the presence of Confidentiality, Integrity, and Data Authentication

Security in RPL

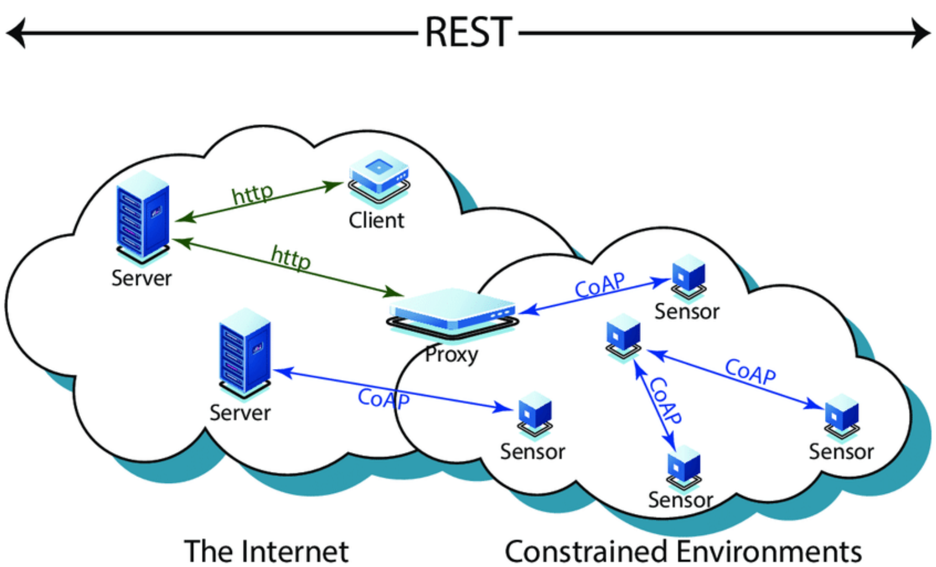
- Support of Confidentiality & delay protection
  - Using AES-CCM(Counter with CBC-MAC)
  - MAC is applied on entire IPV6 packet

- Support of Integrity and Data Authenticity:
  - Using AES-CCM-128
  - OR, using RSA with SHA-256

- Support of Semantic Security and Protection Against Replay Attacks:
  - It is provided with the help of the *Counter* field

- Support for Key Management:
  - The KIM (Key Identifier Mode) field of the *Security* section illustrates different key management approaches

CoAP



CoAP: Constrained Application Protocol

- For constrained and low-power networks.
- Follows the request-response messaging pattern
- The request is sent using Confirmable (CON) or Non-Confirmable (NON) message.
  - Uses stop-and-wait mechanism with exponential backoff to achieve reliability
- Response is sent using several scenarios (e.g. piggy-backed, separate response, etc)

Other Features:

- Very efficient RESTful protocol (i.e. uses REST: Representational State Transfer architecture)
- Low header overhead and parsing complexity
- Uses both asynchronous & synchronous messaging
- Mainly uses for UDP communications

CoAP Security

- CoAP Protocol defines bindings to DTLS (Datagram Transport-Layer Security) to transparently apply security to all CoAP messages

- Security Modes in CoAP:
  - *NoSec*
    - CoAP messages are transmitted without security applied.
  - *PreSharedKey*
    - sensing devices that are pre-programmed with the symmetric cryptographic keys required to support secure communications
  - *RawPublicKey*
    - A given device must be pre-programmed with an asymmetric key pair
    - supports authentication based on public-keys
    - The device has an identity calculated from its public key and a list of identities and public keys of the nodes it can communicate with.
    - This is mandatory for implementing CoAP
  - *Certificates*
    - The device has an asymmetric key pair with an X.509 certificate
    - supports authentication based on public-keys

CoAP Security

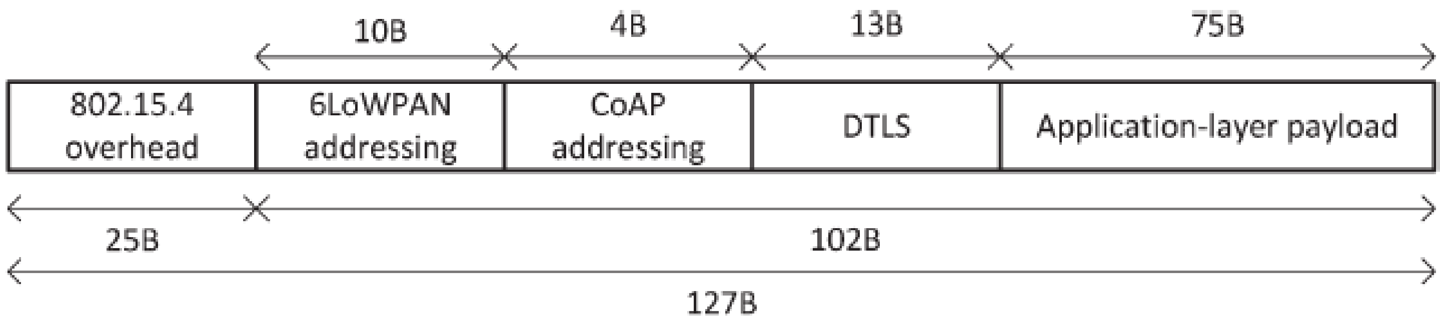


Fig: Payload space with DTLS on 6LoWPAN environments

- Elliptic Curve Cryptography (ECC) is adopted to support the ***RawPublicKey*** and ***Certificates*** security modes

- ECC supports device authentication using the Elliptic Curve Digital Signature Algorithm (ECDSA), and also key agreement using the Elliptic Curve Diffie-Hellman Algorithm with Ephemeral keys (ECDHE).

- ***PreSharedKey*** security mode requires the TLS_PSK_WITH_AES_128_CCM_8 suite, which supports authentication using pre-shared symmetric keys and 8-byte nonce values, and encrypts and produces 8-byte integrity codes.

View from Operational Level

- Security Requirements from three Operation Levels:
  - Information Level
    - Integrity: received data should not been altered during the transmission
    - Anonymity: identity of the data source should remain hidden
    - Confidentiality: Data cannot be read by third parties
    - Privacy: The client's private information should not be disclosed
  - Access Level
    - Access Control: only legitimate users can access to the devices and the network for administrative tasks
    - Authentication: checks whether a device has the right to access a network and vice-versa
    - Authorization: ensures that only the authorized devices /users get access to the network services or resources.
  - Functional Level
    - Resilience: refers to network capacity in case of attacks and failures
    - Self Organization: denotes the capability of an IoT system to adjust itself in order to remain operational even in case of partial failure or attack

Security Mechanisms for IoT

- **Standard security mechanisms** that have been designed to satisfy the security requirements for IoT Services.

  - ## Encryption:
    - Standard Mechanisms:
      - Symmetric mechanism: AES-128, AES-192, AES-256 with CTR or CBC or CCM mode
      - Asymmetric mechanism: RSA , Elgamal
    - Above mechanisms mainly used for confidentiality
    - Authentication and Integrity protection are provided by Message Authentication Code (Symmetric Mechanism), Digital Signature (Asymmetric Mechanism) and Hash Functions.
      - Tag is computed and concatenated with message
      - Tag is obtained by AES-CBE-MAC (Symmetric scheme), Digital Signature Algorithm (Asymmetric scheme) , Elliptic Curve DSA (ECDSA), RSA with Hash Function e.g. MD5, SHA-160, SHA-256, SHA-512

  - ## Lightweight Cryptography:
    - Block cipher : PRESENT, CLEIFA, PRINCE
    - Hash function: PHOTON, SPONGENT

Security Mechanisms for IoT

- **Standard security mechanisms** that have been designed to satisfy the security requirements for IoT Services.

  - Secure Hardware
    - illegitimate user can perform side channel attacks as devices can be deployed in remote areas with less protection
    - It is possible to exploit both hardware and software solutions to eliminate or to mitigate
    - e.g. PUF – Physically Unclonable Functions
    - The basic concept of PUF is to exploit little differences introduced by the fabrication process of the chip to generate a unique signature of each device.
    - Shortcomings: increase of *power consumption* of the device
  - Intrusion Detection Systems
    - Complex anti-virus software and traffic analyzers cannot be used in IoT devices
    - Lightweight IDS: anomalies in system parameters, like CPU usage, memory consumption, and network throughput, may be indicative of an ongoing attack

Implementation in Commercial Device

- In general, IoT devices include at least two microcontrollers:
  - one responsible for the management and processing of the data
  - other for connectivity

  - Most of the IoT devices use ARM microcontrollers of Cortex-M series
    - Require minimal costs, power, and size: M0, M0+, and M23
    - Offers balance between performance and energy efficiency: M3 and M4
    - high performance embedded applications: M7

    - In general, Cortex-M family do not integrate
      - any hardware pseudorandom number generator,
      - any module supporting cryptographic algorithms such as AES

    - Support for cryptographic algorithms is implemented **via software** or **by dedicated co-processors**

Example:

- TI CC2540 MCU provides an AES security co-processor for encryption and decryption
- TI SoC CC2538 implements AES in hardware

Thanks!

Important reference:

- Granjal *et al.*, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues" *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 2015.