

**Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)****Submission Title:** MedWiN MAC and Security Proposal – Part 2 of 2**Date Submitted:** May 4, 2009**Source:** David Davenport<sup>(1)</sup>, Neal Seidl<sup>(2)</sup>, Jeremy Moss<sup>(3)</sup>, Maulin Patel<sup>(4)</sup>, Anuj Batra<sup>(5)</sup>, Jin-Meng Ho<sup>(5)</sup>, Srinath Hosur<sup>(5)</sup>, June Chul Roh<sup>(5)</sup>, Tim Schmidl<sup>(5)</sup>, Okundu Omeni<sup>(6)</sup>, Alan Wong<sup>(6)</sup>

- (1) GE Global Research, davenport@research.ge.com, 518-387-5041, 1 Research Circle, Niskayuna, NY, USA
- (2) GE Healthcare, neal.seidl@med.ge.com, 414-362-3413, 8200 West Tower Avenue, Milwaukee, WI, USA
- (3) Philips, j.moss@philips.com, +44 1223 427530, 101 Cambridge Science Park, Milton Road, Cambridge UK
- (4) Philips, maulin.patel@philips.com, 914-945-6156, 345 Scarborough Road, Briarcliff Manor, NY, USA
- (5) Texas Instruments, {batra@ti.com, 214-480-4220}, {jinmengho@ti.com, 214-480-1994}, {hosur@ti.com, 214-480-4432}, {jroh@ti.com, 214-567-4145}, {schmidl@ti.com, 214-480-4460}, 12500 TI Blvd, Dallas, TX, USA
- (6) Toumaz Technology, {okundu.omeni@tomuaz.com, +44 1235 438950}, {alan.wong@toumaz.com, +44 1235 438961}, Building 3, 115 Milton Park, Abingdon, Oxfordshire, UK

**Re:** Response to IEEE 802.15.6 call for proposals**Abstract:** This presentation illustrates the major MAC aspects of a joint MAC and security proposal detailed in an accompanying normative text document doc. IEEE 802.15-09-0327-00-0006.**Purpose:** To submit a joint proposal on MAC and security to the IEEE 802.15.6 task group**Notice:** This document has been prepared to assist the IEEE P802.15. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.**Release:** The contributors acknowledge and accept that this contribution becomes the property of IEEE and may be made publicly available by P802.15

# MedWiN MAC and Security Proposal

## *Part 2 of 2 – Security*

GE Global Research: David Davenport

GE Healthcare: Neal Seidl

Philips: Jeremy Moss, Maulin Patel

Texas Instruments: Anuj Batra, Jin-Meng Ho  
Srinath Hosur, June Chul Roh  
Tim Schmidl

Toumaz Technology: Okundu Omeni, Alan Wong

# Outline

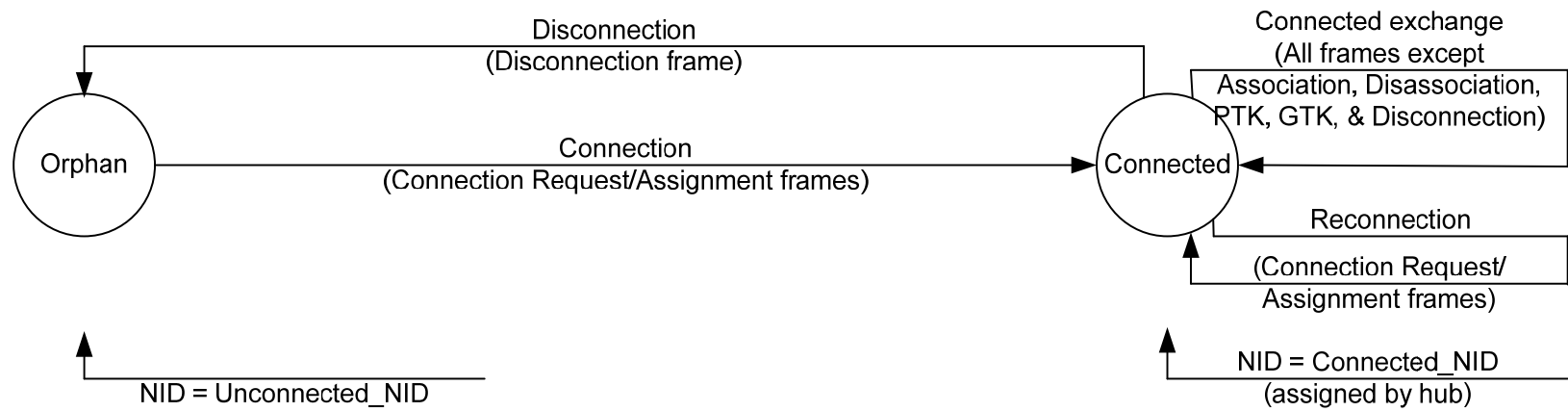
- Motivation
- Access state diagrams & Security hierarchy
- Security services
  - Security fields
  - AES-128 CCM input blocks
  - AES-128 CBC authentication
  - AES-128 CTR mode encryption/decryption
  - Replay filtering
- Security keys
  - Temporal key (TK) creation/distribution
    - Pairwise temporal key (PTK) creation for unicast protection
    - Group temporal key (GTK) distribution for multicast/broadcast protection
  - Master key (MK) generation protocols provided
    - ✓ MK preinstalled association
    - ✓ Unauthenticated association
    - ✓ Public key hidden association
    - ✓ Password authenticated association
    - ✓ Display authenticated association
- Implementation Estimates

# Motivation

- Security needs to be built into the system from the start, not as an after thought
- Patient data usually requires protection from casual eavesdropper
  - Data confidentiality → Encryption
- Integrity of data also needs to be provided → Authentication
- Message source authentication
- Applications which do not require security don't turn it on

# Access State Diagram 1 – No Security

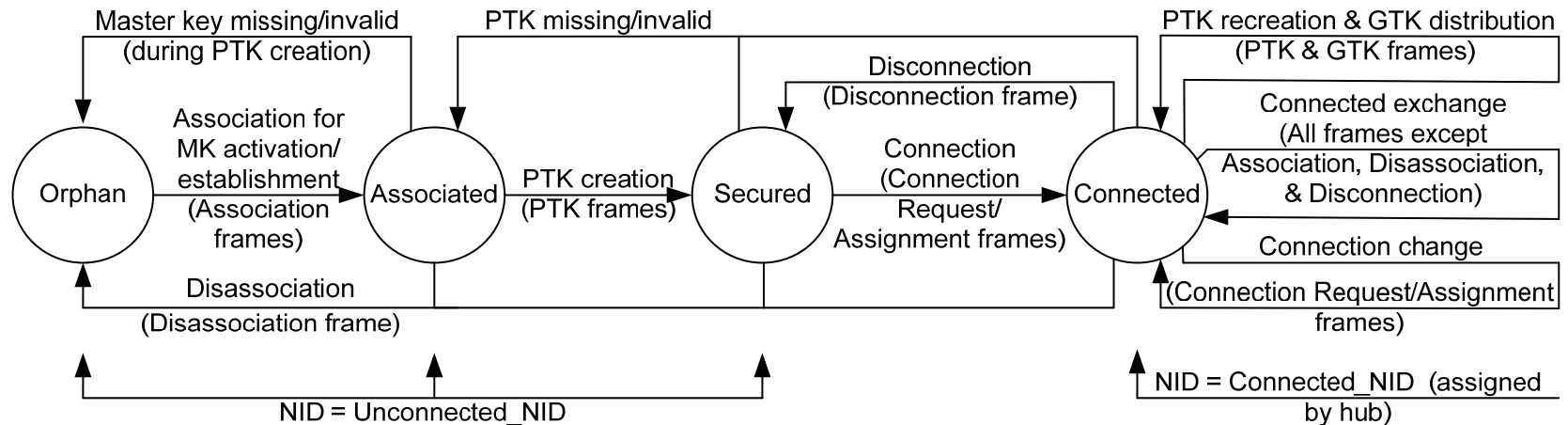
- Orphan state – node not connected to hub. Can only send connection request message
- Connected state – node connected to hub. Can send all unsecured messages



(a) Unsecured communication

## Access State Diagram 2 – Security Enabled

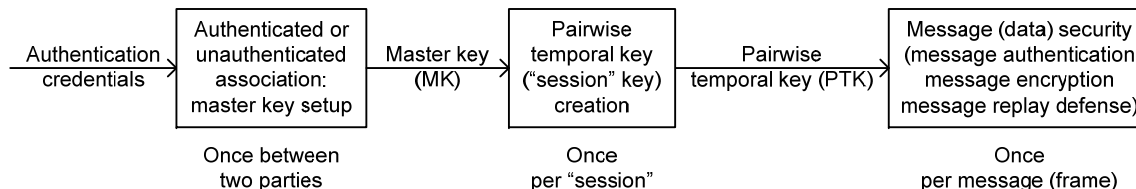
- Associated state – Master keys activated or generated
- Secured state – PTK created
- Connected state – connected to hub. Can send only secured messages



(b) Secured communication

# Security Hierarchy

- A PTK (session key) is needed for data security
  - ❑ PTK (data) → security check sum → data authenticity & integrity
  - ❑ PTK (data) → encryption → data confidentiality & privacy
  - ❑ PTK (data) + security sequence # → security checksum → replay defense
- A master key (MK) is needed for PTK creation
  - ❑ The master key is not used as a session key for security reasons
  - ❑ Compromise of a PTK does not break the master key
- Association is needed for MK setup
  - ❑ How can two devices establish a secret MK—even with attackers around?
  - ❑ How can a device reject a MK setup with an unauthorized device?
  - ❑ How can a device allow a MK setup with a “rescuer” in an emergency?



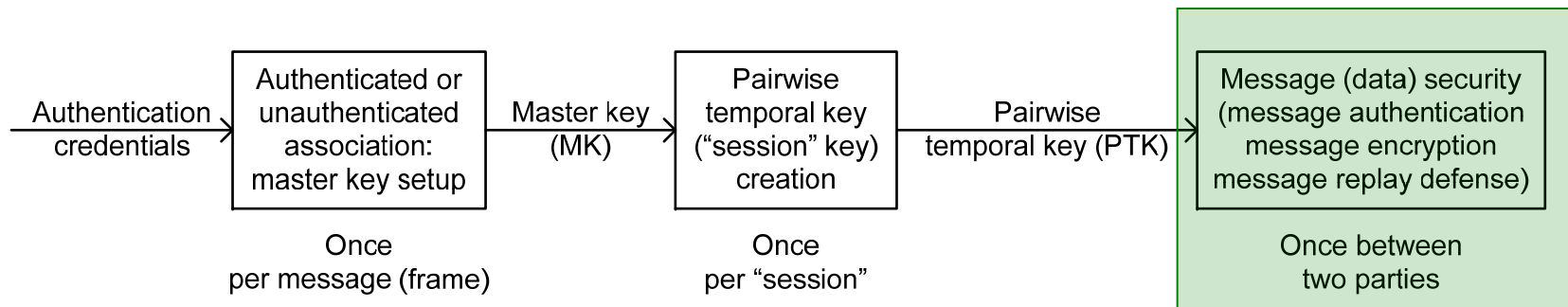
# Connection Time

- Made up of the following components
  - Time to find Hub's channel
  - Time to get contention interval or poll from Hub
  - Time to exchange association & PTK creation frames (in secured connection)
    - Only required when joining for the 1<sup>st</sup> time
  - Time to exchange connection frames
- Also depends on the beacon period and frequency of channel hopping of the hub



# Security Services

- Security indication
  - Security fields in MAC header and frame body
- Authentication, encryption, and decryption
  - Format of AES-128 CCM blocks
  - Authentication, encryption, and decryption operation
- Replay filtering
  - Security sequence number

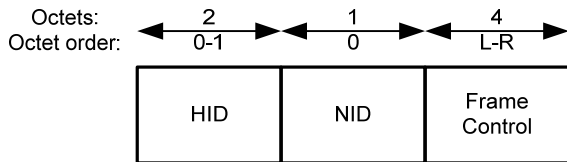


# Security Fields in MAC Header

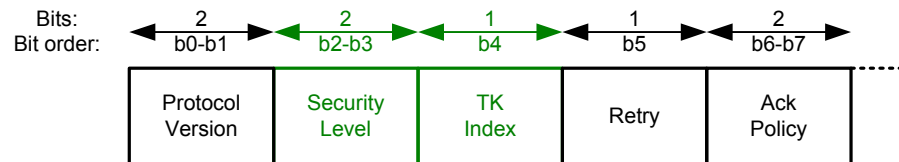
- Security Level
  - ❑ Indicates the security level of the current frame
- TK Index
  - ❑ Indicates the pairwise temporal key (PTK) or group temporal key (GTK) being used to secure the current frame
  - ❑ Provided for PTK or GTK change

Table 1 — Security Level field encoding

Field value b2 b3	Security level of current frame
00	Level 0 – frame not secured
10	Level 1 – frame authenticated but not encrypted
01	Level 2 – frame authenticated and encrypted
11	Reserved



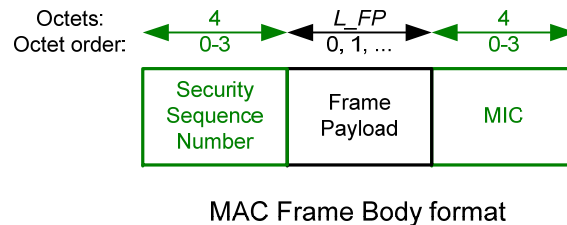
MAC Header format



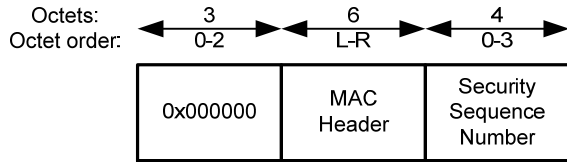
Frame Control format

# Security Fields in MAC Frame Body

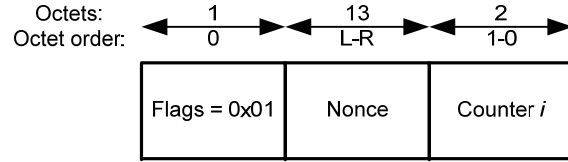
- Security Sequence Number
  - ❑ Increments by every frame transmission or retransmission secured with the same PTK or GTK
  - ❑ Provided for nonce construction and replay detection
- MIC
  - ❑ Set to a keyed message authentication check computed based on AES-128 CCM
  - ❑ Provided for preserving the authenticity and integrity of the current frame



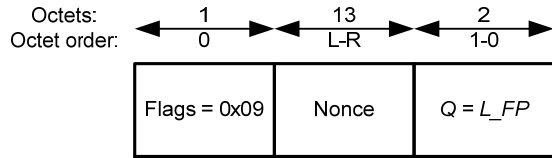
# AES-128 CCM Input Blocks



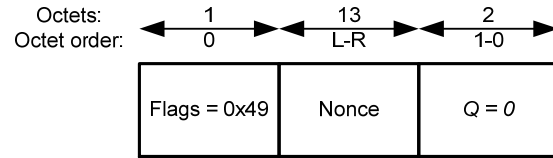
Nonce format



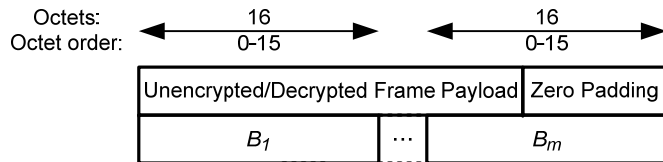
$Ctr_i$  format



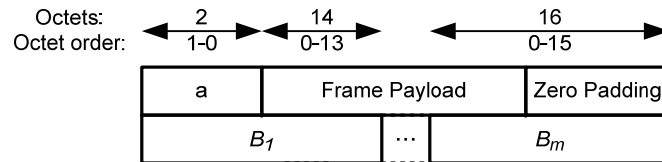
$B_0$  format (frame payload encrypted)



$B_0$  format (frame payload not encrypted)



$B_1, \dots, B_m$  format (frame payload encrypted)

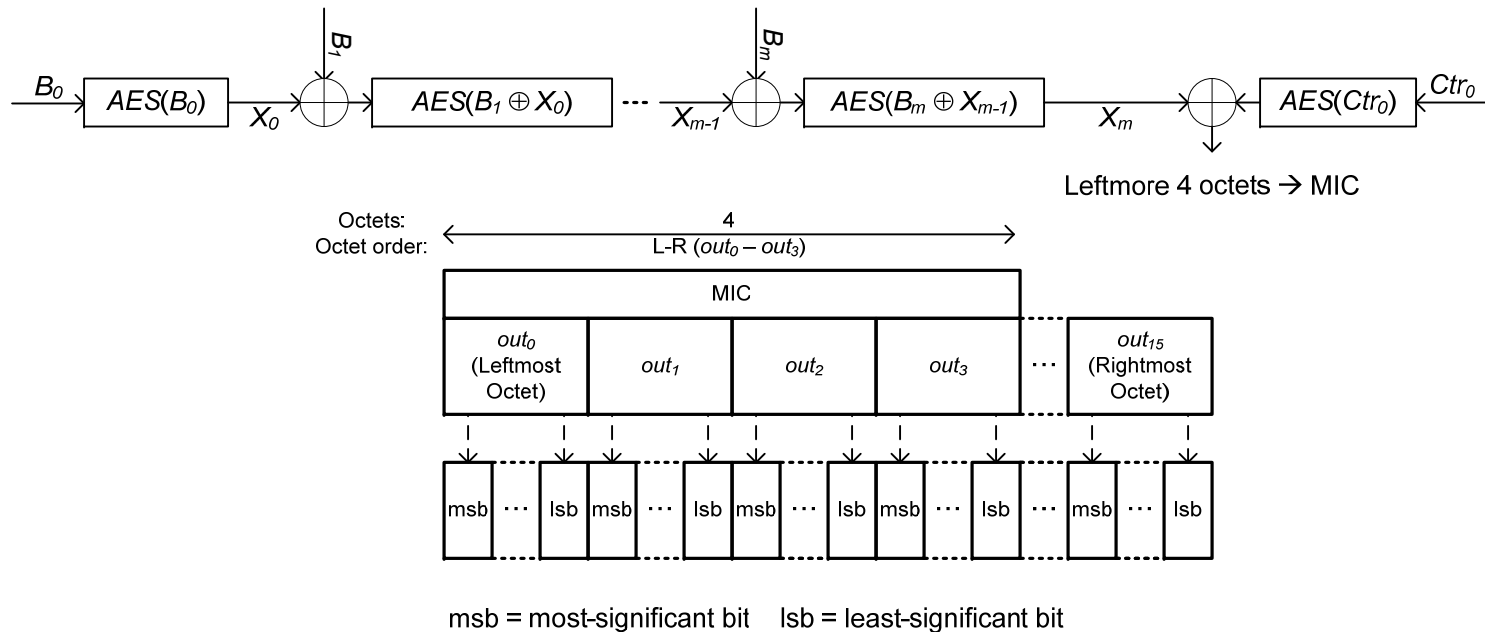


$B_1, \dots, B_m$  format (frame payload not encrypted)

# AES-128 CBC Authentication

$$X_0 = AES(B_0), \quad X_i = AES(B_i \oplus X_{i-1}), \quad i = 1, \dots, m$$

$$MIC = LMB\_n(M), \quad M = AES(Ctr_0) \oplus X_m$$

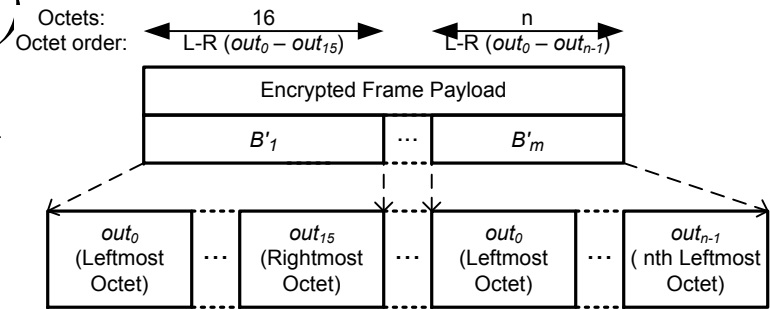
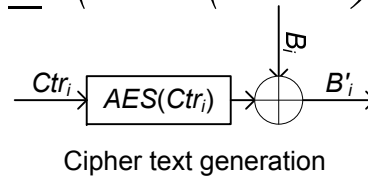


MIC calculation and transmit order

# AES-128 CTR Mode Encryption/Decryption

$$B'_i = B_i \oplus AES(CTR_i), i = 1, \dots, m-1$$

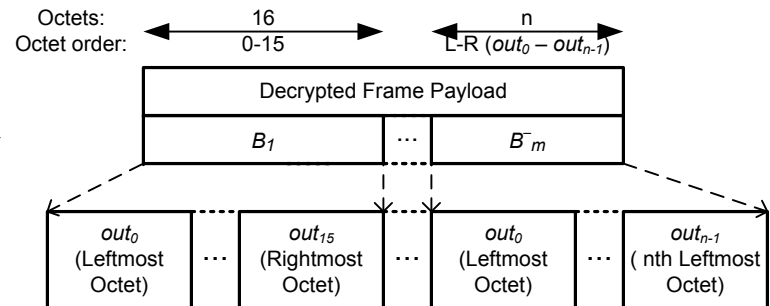
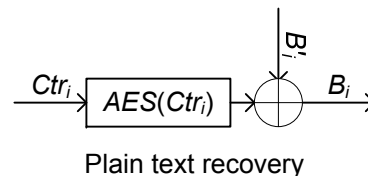
$$B'_m = L_n(B_m) \oplus L_n(AES(CTR_m))$$



Encrypted Frame Payload format

$$B_i = B'_i \oplus AES(CTR_i), i = 1, \dots, m-1$$

$$B_m = B'_m \oplus L_n(AES(CTR_m))$$



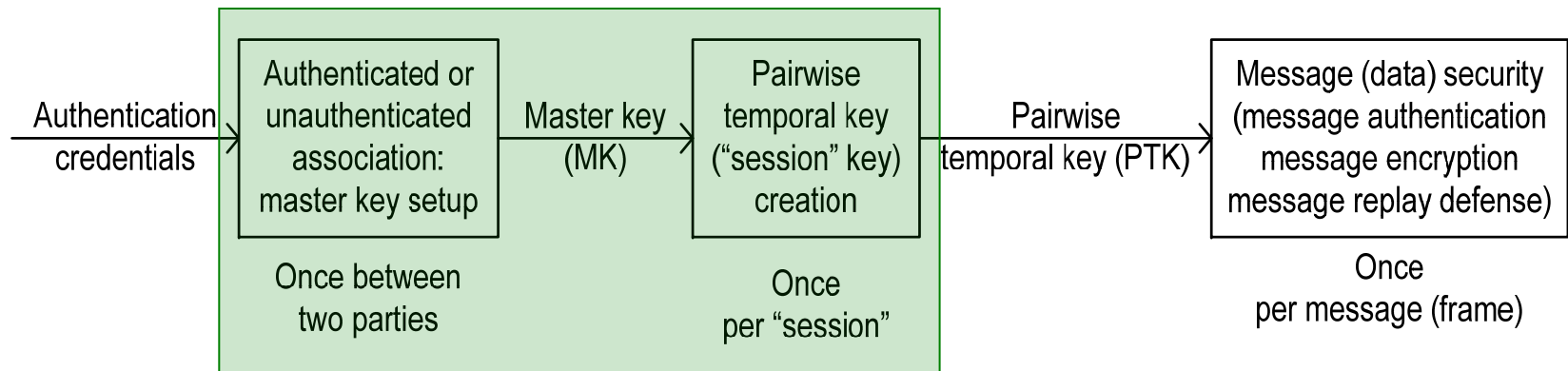
Decrypted Frame Payload format

# Replay Filtering

- Security sequence number setting on transmission
  - Set to 1 in a frame secured with a new PTK or GTK
  - Incremented by one for each successive frame transmission or retransmission secured with the same PTK or GTK
- Replay filtering on reception
  - Accept a MIC-valid frame secured with a new PTK
  - Accept a MIC-valid frame with a security sequence number  $>$  the security sequence number contained in the last accepted frame secured with the same PTK
  - Discard a frame with a security sequence number  $\leq$  the security sequence number contained in the last accepted frame secured with the same PTK

# Security Keys

- PTK creation & GTK distribution
  - ❑ PTK creation based on a master key
  - ❑ GTK distribution based on a PTK
  - ❑ A “Session” as indicated below can be for as many as  $2^{32}$  data frame transactions which would usually be the lifetime of a node
- MK setup
  - ❑ Done only once between a node and a hub
    - By running an association protocol
    - Five association protocols provided





# PTK Creation

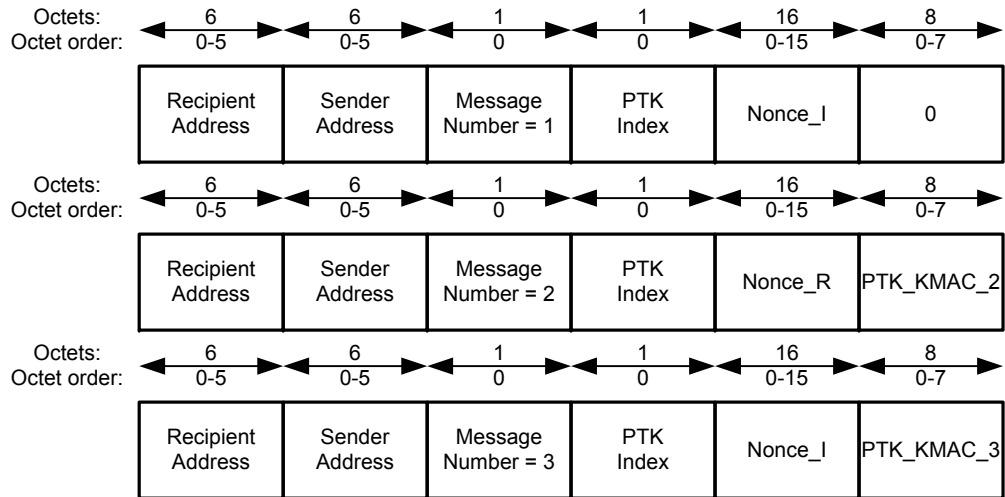
- MK → PTK
  - PTK compromised → MK **NOT** compromised → One-way hash
  - Freshness → both parties have a say in PTK creation → Nonces combined

$$PTK = CMAC(MK, Address\_I || Address\_R || Nonce\_I || Nonce\_R || PTK\_Index)$$

$$KCK = CMAC(MK, Address\_R || Address\_I || Nonce\_R || Nonce\_I || PTK\_Index)$$

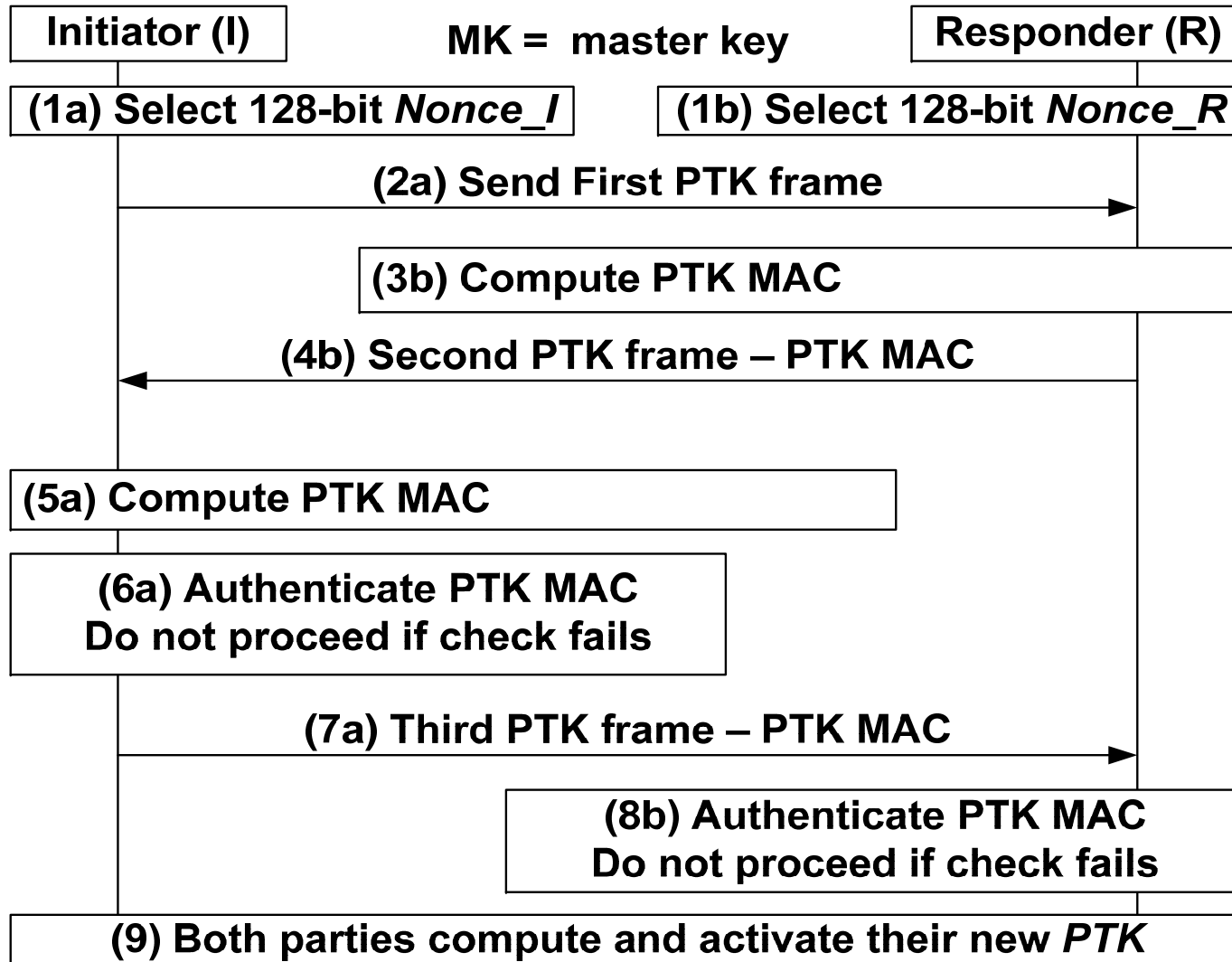
$$PTK\_KMAC\_2 = LMB\_64(P), PTK\_KMAC\_3 = RMB\_64(P)$$

$$P = CMAC(KCK, Address\_I || Address\_R || Nonce\_R || Nonce\_I || PTK\_Index)$$



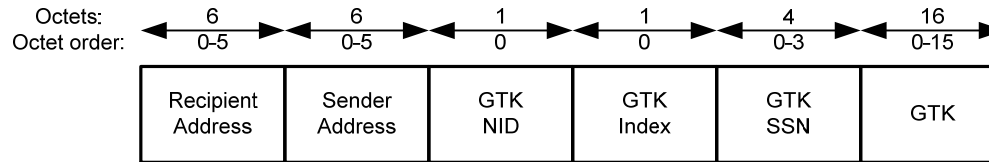
Frame Payload format for PTK frames

# PTK Creation – Flowchart



# GTK Distribution

- PTK → GTK
  - For securing broadcast & multicast
  - GTK contained in a frame encrypted & authenticated by the PTK



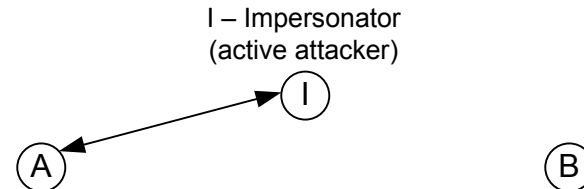
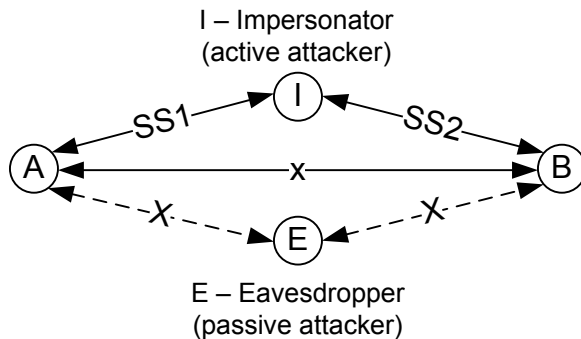
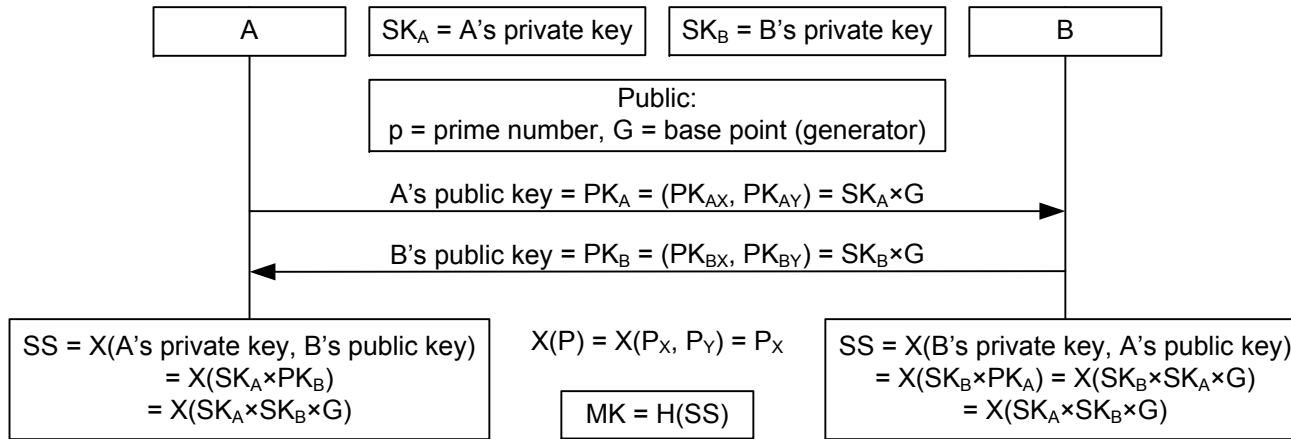
Frame Payload format for GTK frames

# Diffie-Hellman Key Exchange

- 2 common methods:
  - Modular Exponentiation (traditional Diffie Hellman)
  - Elliptic Curve
- Both based on discrete logarithm – mathematically hard → resistant to eavesdropping
- Subject to impersonation & man-in-the-middle attacks → authentication needed
- Elliptic curve based implementation requires significantly fewer bits for same level of security, and so lower complexity implementation e.g. 192-bits versus 1536-bits for modular exponentiation

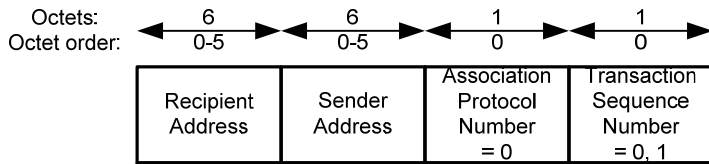
# Diffie-Hellman Key Exchange (2)

$$y^2 = x^3 + ax + b \pmod{p}, \quad a, b \in GF(p), \quad 4a^3 + 27b^2 \neq 0$$

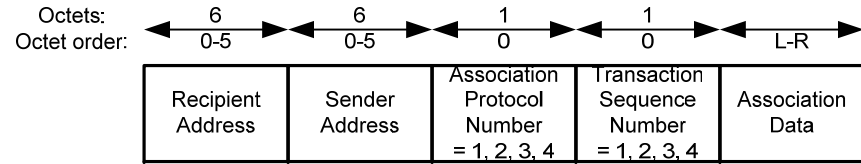


# Association frames

- Exchanged to set out a pre-shared master key (MK) or set up a new MK
- Same general framework for different association protocols



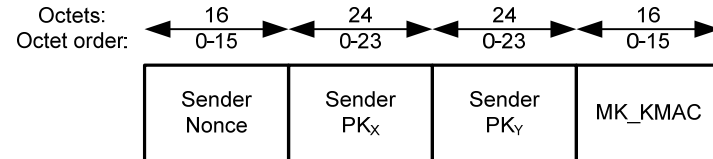
Frame Payload format for Association frames – master key pre-shared association protocol



Frame Payload format for Association frames – unauthenticated association, public key hidden association, password authenticated association, and display authenticated association

Table — Association Protocol Number field encoding

Field value decimal	Association protocol
0	Master key pre-shared association
1	Unauthenticated association
2	Public key hidden association
3	Password authenticated association
4	Display authenticated association
5-255	Reserved



Association Data format for association protocols 1-4

$$DHKey = X(SK_A \times PK_B) = X(SK_B \times PK_A)$$

$$MK\_KMAC\_2 = CMAC(DHKey, Address\_A \parallel Address\_B \parallel Nonce\_A \parallel Nonce\_B)$$

$$MK\_KMAC\_3 = CMAC(DHKey, Address\_B \parallel Address\_A \parallel Nonce\_B \parallel Nonce\_A)$$

$$H = CMAC(DHKey, Address\_A \parallel Address\_B \parallel Nonce\_A \parallel Nonce\_B)$$

$$D = RMB\_16(H), D\_A = BS2DI(D), D\_B = BS2DI(D)$$

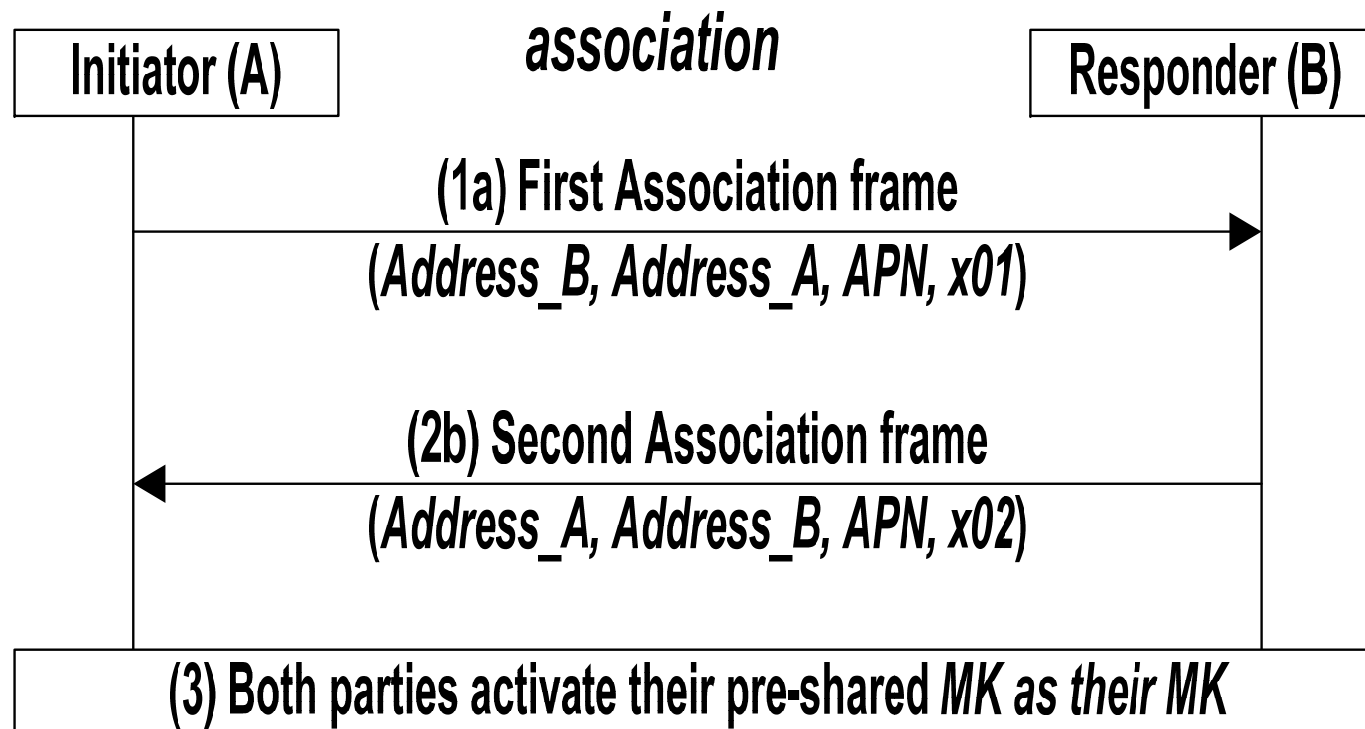
$$Commitment = CMAC(Nonce\_A, Address\_A \parallel Address\_B \parallel PK_{AX} \parallel PK_{AY})$$

$$MK = CMAC(DHKey, Nonce\_A \parallel Nonce\_B)$$

# Master Key Pre-shared Association

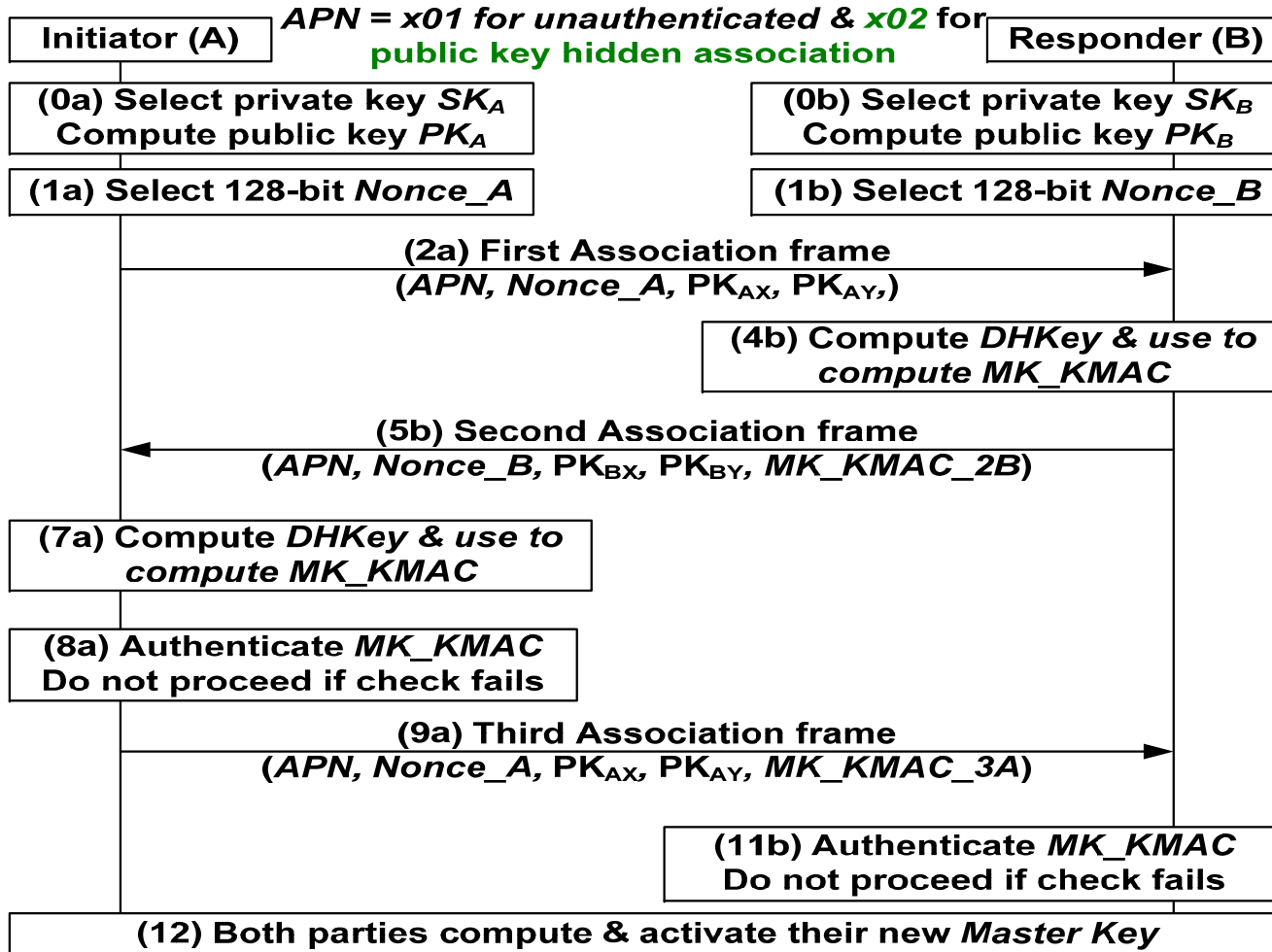
- A simple two-way handshake to set out, i.e., activate, the pre-shared MK
- Loss of the MK list at the hub could expose all the data sent earlier or later
- Cannot “automatically” generate a new MK if the existing MK is compromised

***APN = x00 for master key pre-shared***



# Unauthenticated Association

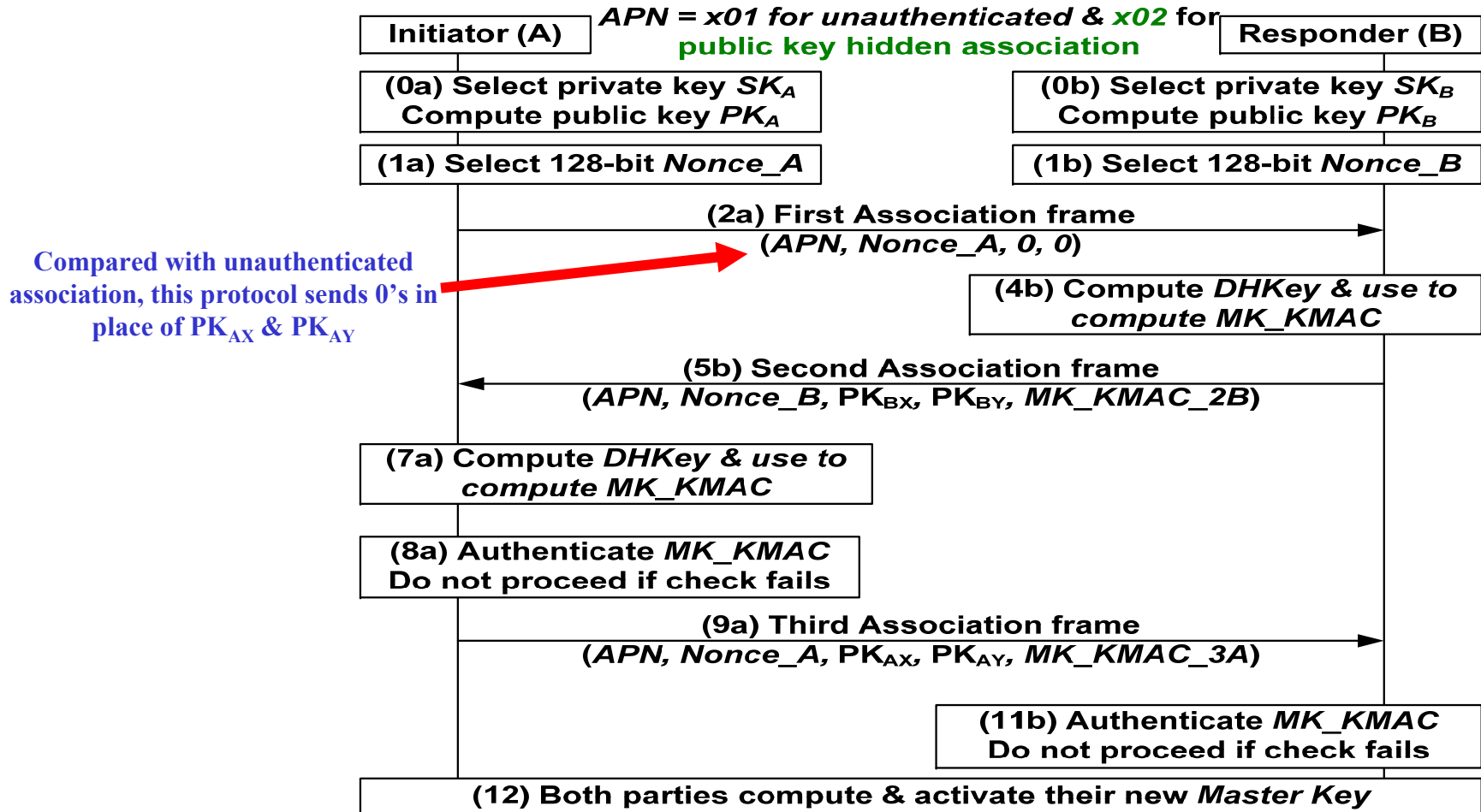
- May be used in a controlled environment with no active attacks expected
- Requires no authentication credentials or special user interfaces for verification





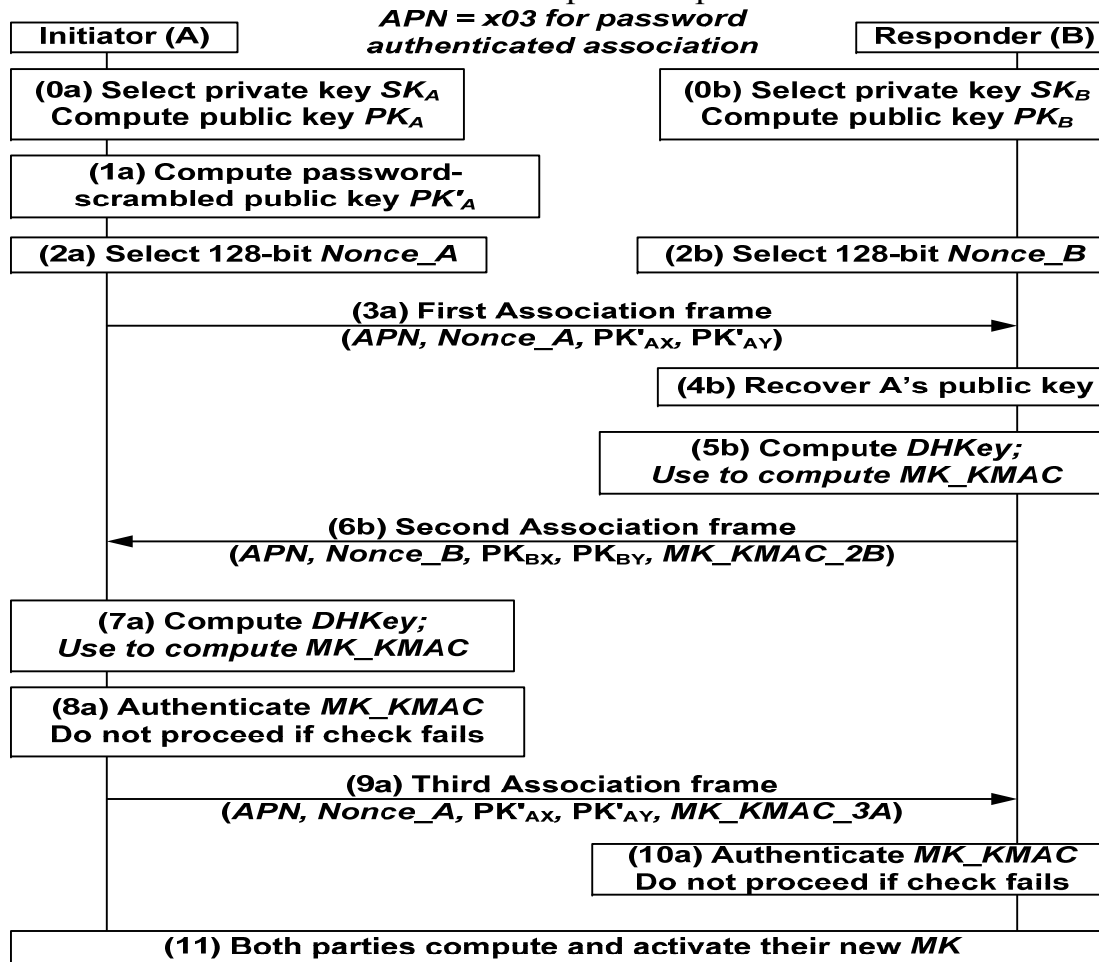
# Public Key Hidden Association

- Node's public key transferred to hub in a secure out-of-band channel in advance
- Improbable over-the-air active (impersonation & MITM) attacks
- Lost of the public key list at the hub would NOT compromise previous data



# Password Authenticated Association

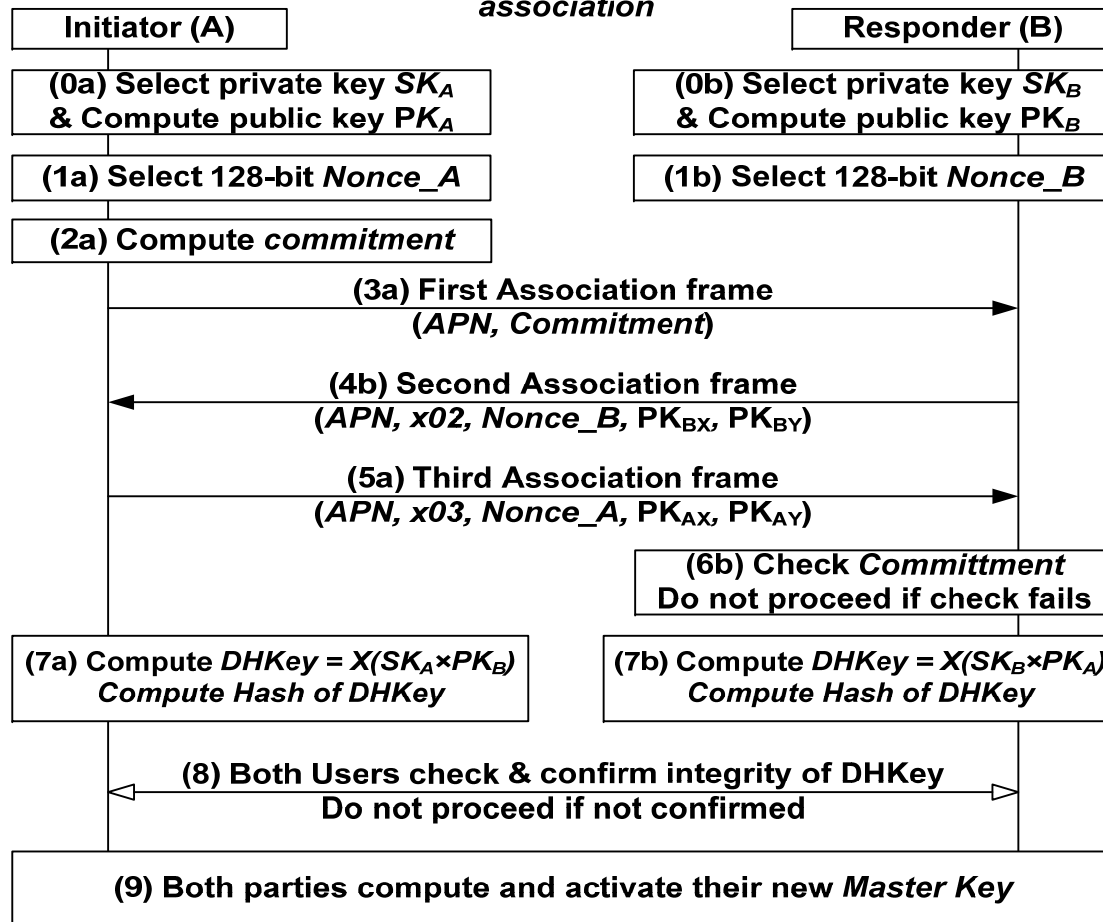
- Node & hub have a shared password in advance
- Improbable over-the-air active (impersonation & MITM) attacks possible
- Lost of the password list at the hub would NOT compromise previous data



# Display Authenticated Association

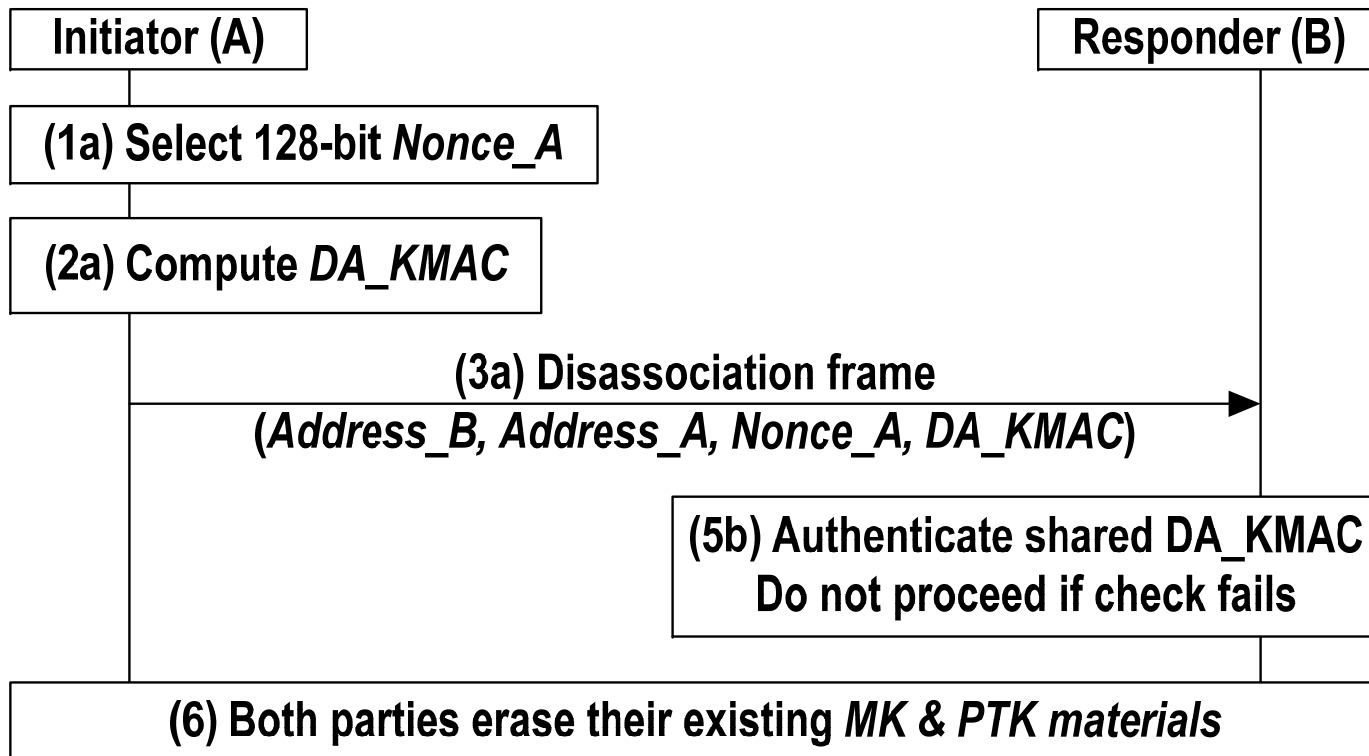
- Node & hub have a visible display of 5-digit decimal numbers
- Improbable over-the-air active (impersonation & MITM) attacks possible

*APN = x04 for display authenticated association*



# Disassociation

- To void an existing association, i.e., to repeal a shared master key



# Implementation Estimates

- AES-128 CCM requires around 10kgates for a hardware implementation
- Master Key pre-shared association protocol doesn't have additional implementation requirements
- Other association protocols require the ECDH key exchange algorithm
  - ❑ Software implementation takes around 20 KB of memory (program and data) on a 16-bit processor running at 8MHz. Run time is a few seconds [1]
  - ❑ Hardware implementation requires between 100-200k gates. At the same 8 MHz clock, it would take around under 100 ms [2,3]
    - This time can be halved if the public key computation is done in advance.
  - ❑ The references were designed with a goal of high throughput, so for this implementation, we expect we can optimize for area and get this down to between 10-20k gates. At 8 MHz a key exchange would then take around 1 second (or 500 ms if public key computation is done in advance)
- Total implementation area for the entire security functionality should be between 20-30kgates
  - ❑ 10kgates if keys are pre-shared

# MedWiN Security Proposal Summary

- Access state diagrams & security hierarchy – defined and described
- Security services – (specified in normative text)
  - Relevant mechanisms defined and illustrated
- Security keys – (specified in an annex)
  - Temporal key (TK) creation/distribution
  - Master key (MK) generation
    - Provided protocols illustrated & described
- Implementation Estimates outlined

*Details including references are provided in the accompanying normative text doc. IEEE 802.15-09-0327-00-00006.*

# Comparison Criteria

Criteria	Proposed Capability
1. Regulatory	Compliant with TG6 regulatory document in multiple frequency bands
2. Raw PHY data rate	100 kbps to 1 Mbps supported between node and hub
3. Transmission distance	PER and link budget shown to support 10% PER for 255 octet PSDU at 3 meters within all operating frequency bands proposed.
4. Packet error rate	
5. Link budget	
6. Power emission level	-10 dBm / -16 dBm maximum EIRP
7. Interference and coexistence	MAC: Channel hopping, Beacon shifting, Acknowledgements, Poll/Post for additional retransmission if necessary. PHY: Channelization $\geq 10$ channels, same channel bandwidth for all modulations at each frequency band, low sidelobes of selected modulation
<b>8. Security</b>	<b>MAC provides 3 levels of security (none, authentication, authentication + encryption) based on AES-128. Association protocols provided for master key setup.</b>
9. Reliability	Acknowledged traffic, guard time and node synchronization to beacon provided. Unique identifications used to distinguish between collocated BANs. Link margin sufficient given TG6 channel models variations.
10. Quality of Service	MAC: Time to join a network $\sim 63$ msec for message exchange. Fast ( $< 1$ sec) channel access available via prioritized CSMA/CA random access as well as scheduled or improvised access mechanisms.
11. Scalability	PHY: Scalable data rate from common symbol rates. MAC: Multiple nodes supported via m-periodic scheduled, improvised and random access methods. Prioritized QoS and beacon configuration.
12. MAC transparency	MAC transparent across multiple frequency bands proposed
13. Power Efficiency	MAC: Sleep and Hibernate modes. PHY: $\leq 3.1$ mW (active), $50 \mu\text{W}$ (standby), $250/125$ nW (deep sleep)
14. Topology	Star topology, broadcast beacon supported. Maximum number of nodes supported via multiple access mechanisms.
15. Bonus Point	<b>Merged proposal focused on satisfying needs of medical BAN applications as defined by TG6 PAR.</b>

# Additional References

- [1] <http://discovery.csc.ncsu.edu/pubs/ipsn08-TinyECC-IEEE.pdf>
- [2] S. B. Ors, L. Batina, B. Preneel, J. Vandewalle, “Hardware Implementation of an Elliptic Curve Processor over  $GF(p)$ ”, Proceedings. IEEE International Conference on Application-Specific Systems, Architectures, and Processors, 2003
- [3] “Certicom® Suite B Public Key IP Core™”, [http://www.certicom.com/images/pdfs/ds-suiteb-pk-  
ipcore.pdf](http://www.certicom.com/images/pdfs/ds-suiteb-pk-ipcore.pdf)