State of the Art of *Secure Internet of Things (S-IoT):*
The Development of New Cryptographic Key Updating Schemes to Improve the Security of Long-Range Wide Area Network (LoRaWAN) Protocol

Kalamullah Ramli and Nur Hayati

Co-Founder, Indonesia Cyber Awareness and Resilience (id-CARE) Institute

Professor, Electrical Engineering Department

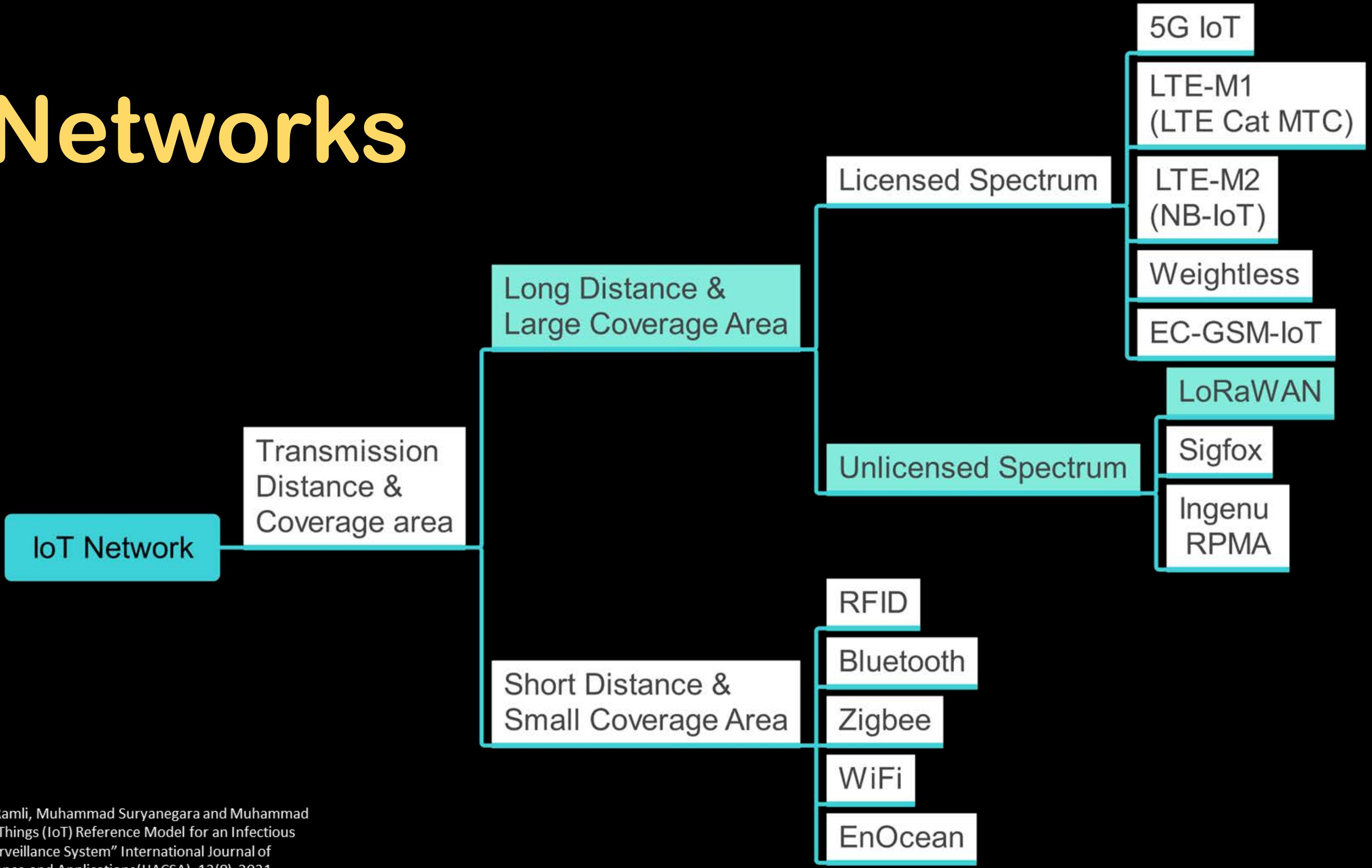
Universitas Indonesia

50th Year of ASEAN-Japan Friendship and Cooperation

OUTLINE

- **Introduction:** IoT Use Case and IoT Security Threats
- **LoRaWAN Security**
- **LoRaWan Security Issues**
- **Proposed Solutions:**
  - **Root Key Update Scheme**
  - **Session Key Update Scheme**
- **Conclusions**

IoT Networks



Nur Hayati, Kalamullah Ramli, Muhammad Suryanegara and Muhammad Salman, "An Internet of Things (IoT) Reference Model for an Infectious Disease Active Digital Surveillance System" International Journal of Advanced Computer Science and Applications(IJACSA), 12(9), 2021. http://dx.doi.org/10.14569/IJACSA.2021.0120956

IoT Security Threats

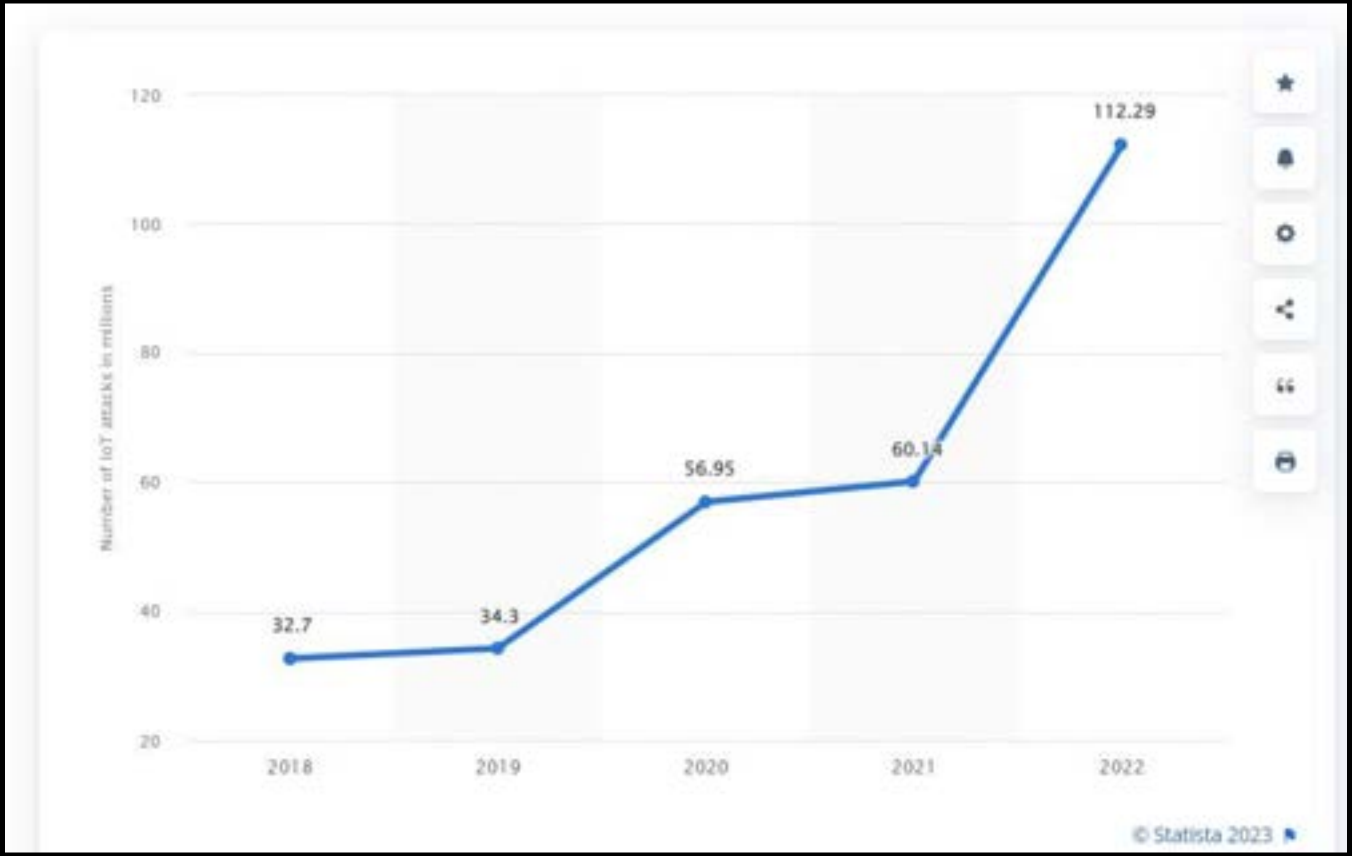
- An IoT attack is a malicious attempt to exploit vulnerabilities in Internet-connected devices such as smart office devices, industrial control system, and critical infrastructure key components
- Attackers may seize control of the device, steal sensitive data, or use the device as a part of a botnet for other malicious purposes
- With limited resources and processing power, IoT devices may lack security features to protect against attacks, making them more vulnerable to attacks than other IT equipment

IoT Security Threat
(in numbers)

The number of Internet of Things (IoT) cyber attacks worldwide amounted to over 112 million in 2022. Over the recent years, this figure has increased significantly from around 32 million detected cases in 2018. In the latest measured year, the year-over-year increase in the number of Internet of Things (IoT) malware incidents was 87 percent

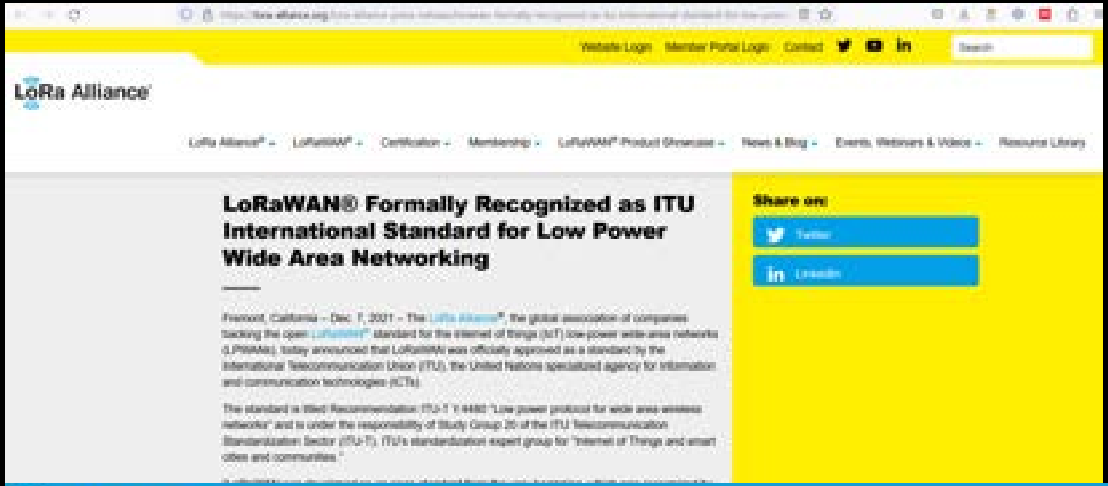
Source: Statista, Feb 2023

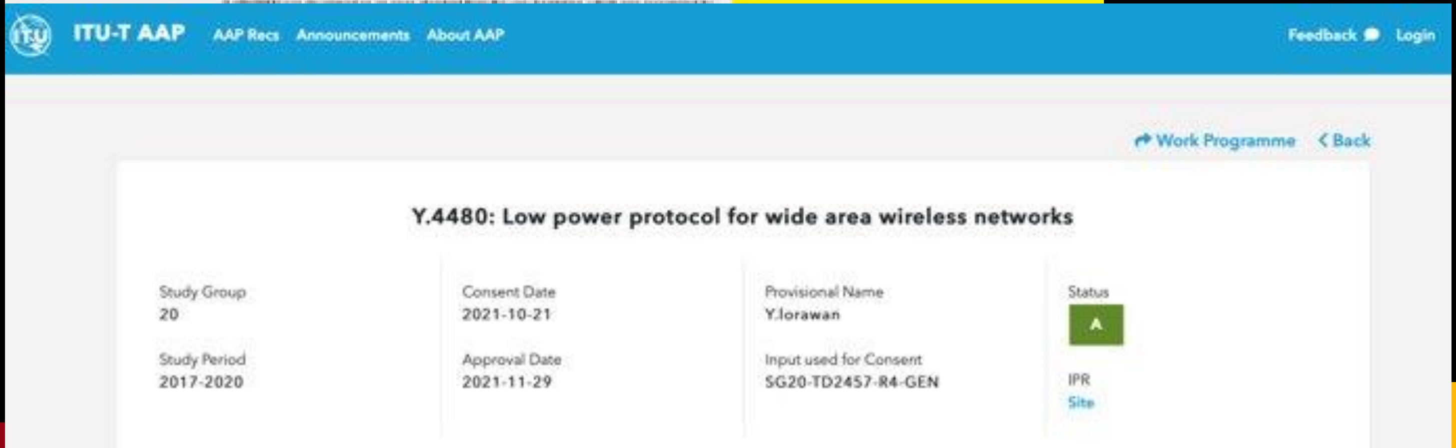
https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/

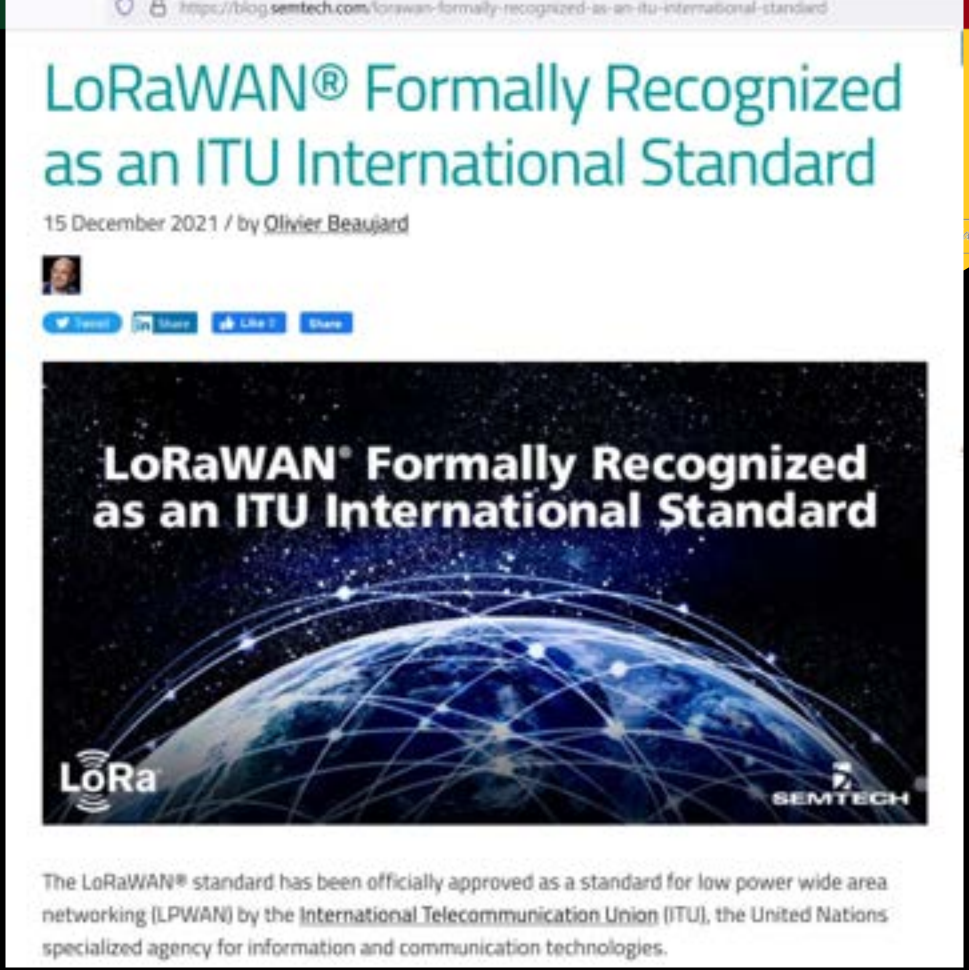


Why LoRaWAN ?

- LoRaWAN become the standard of Internet of Things (IoT) Low Power Wide Area Network (LPWAN) through **ITU-TY.4480 recommendation** (Des, 2021)



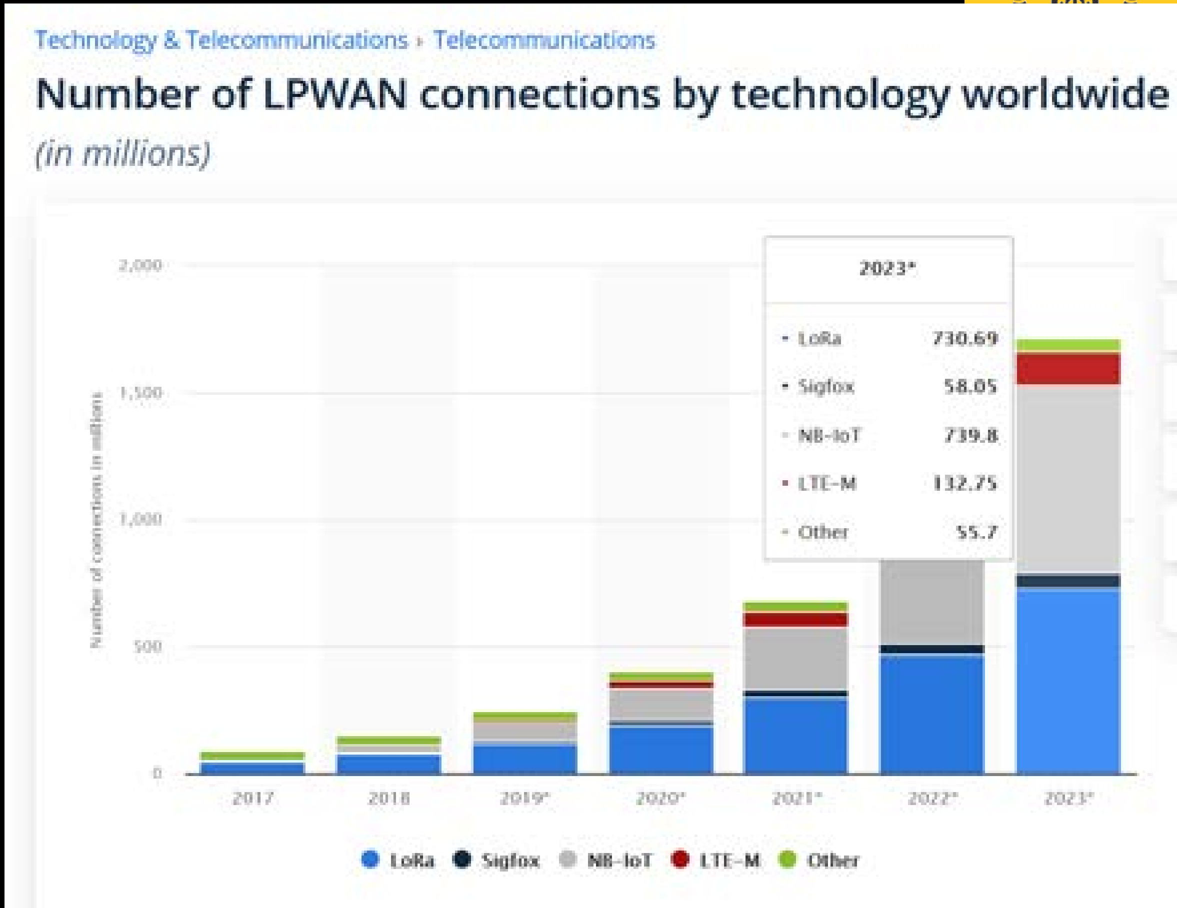




- https://www.itu.int/rec/T-REC-Y.4480/en
- https://lora-alliance.org/lora-alliance-press-release/lorawan-formally-recognized-as-itu-international-standard-for-low-power-wide-area-networking/
- https://blog.semtech.com/lorawan-formally-recognized-as-an-itu-international-standard

Why LoRaWAN ?

- The Number of IoT Connections in 2023 is 1,716.99 Million. LoRa connections reach ± 42.55% of the total or as many as 730.69 million (Source: Statista, July 2023)
- There are 5.9 million LoRa gateways, 300 million end devices/nodes, and 181 public network operators. LoRa technology has been applied to various sectors (Source: Semtech, August 2023)



LoRa By the Numbers

5.9 million	300 million	181	>50%
gateways with LoRa devices deployed worldwide (March 2023)	end nodes with LoRa devices deployed worldwide (March 2023)	public network operators and growing (March 2023)	of all non-cellular LPWA connections will feature LoRa by 2026 (ABI Research)

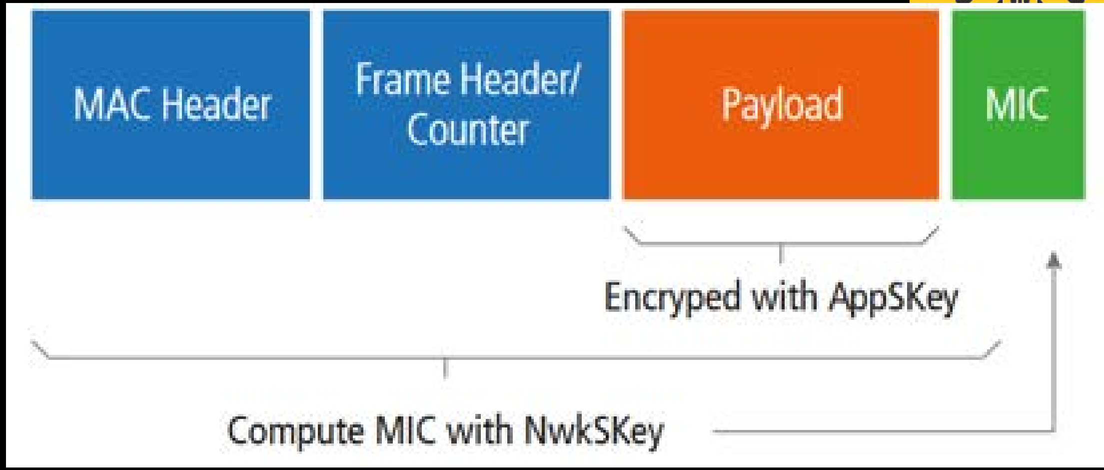
https://www.statista.com/statistics/880822/lpwan-ic-market-share-by-technology/
https://www.semtech.com/lora

LoRaWAN Security

1. Mutual authentication
2. Data Integrity
3. Data Confidentiality

AES 128 bit

LoRaWAN security mechanisms rely on the AES cryptographic algorithms



MIC: Message Integrity Code

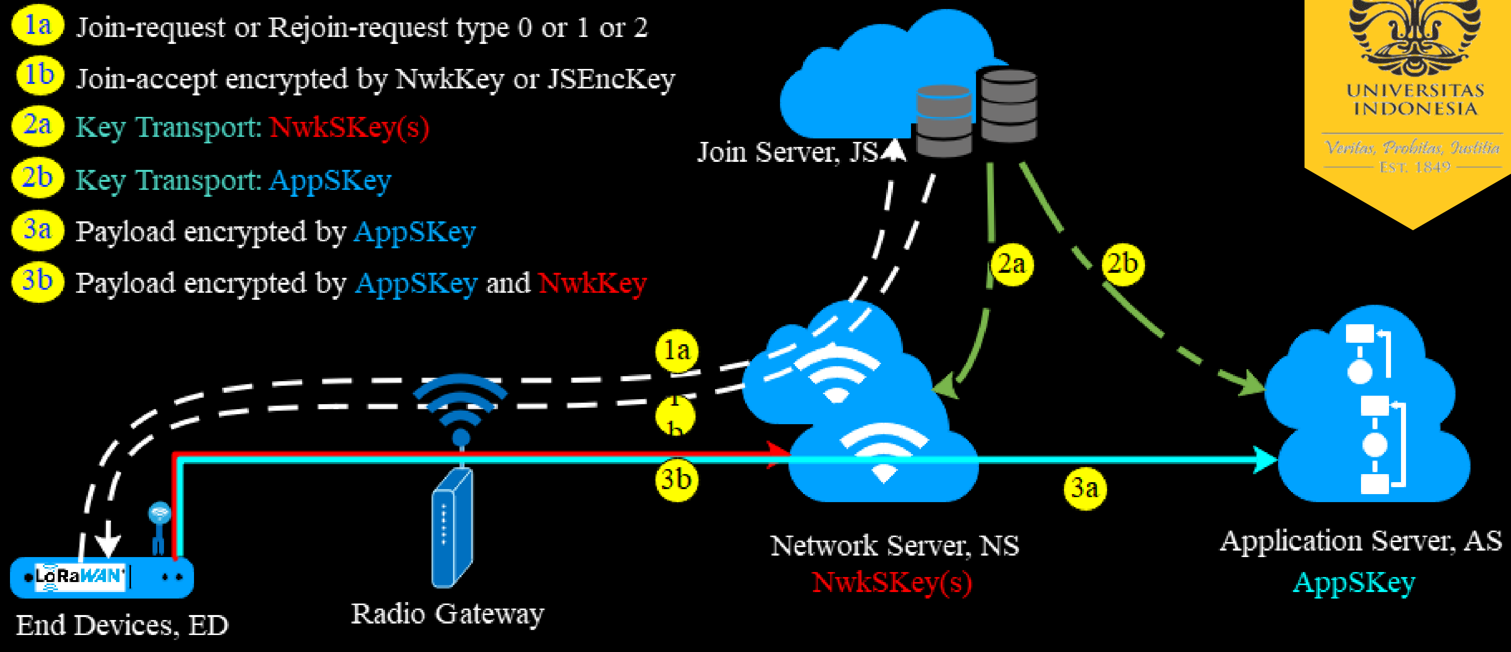
- **Mutual authentication** is established between a LoRaWAN end-device and the LoRaWAN network as part of the network join procedure through Over-the-Air-Activation (OTAA). The OTAA Join Procedure proves that both the end device and the network have the knowledge of the **root key**, specifically AppKey.

- **Data Integrity and Confidentiality Protection:** All LoRaWAN traffic is protected using the **two session keys**. Each payload is encrypted by AES-CTR and carries a frame counter (to avoid packet replay) and a Message Integrity Code (MIC) computed with AES-CMAC (to avoid packet tampering).

Gemalto, Actility, and Semtech, ""LoRaWAN™ Security A White Paper." Accessed: Dec. 03, 2021. [Online]. Available: https://lora-alliance.org /sites/lorawan_security_whitepaper.pdf

LoRaWAN Security

- LoRaWAN security uses the AES cryptographic algorithm for integrity protection and encryption.
- Each LoRaWAN device is personalized with a unique 128 bit AES key (**called root key**)
  - Root key LoRaWAN consist of NwkKey & AppKey



- LoRaWAN **session keys** are then derived, one for providing integrity protection and encryption of the LoRaWAN MAC commands and application payload (the NwkSKey), and one for end-to-end encryption of application payload (the AppSKey).
  - The NwkSKey is distributed to the LoRaWAN network in order to prove/verify the packets authenticity & integrity.
  - The AppSKey is distributed to the application server in order to encrypt/decrypt the application payload.

N. Hayati, K. Ramli, S. Windarta and M. Suryanegara, "A Novel Secure Root Key Updating Scheme for LoRaWANs Based on CTR_AES DRBG 128," in *IEEE Access*, vol. 10, pp. 18807-18819, 2022, doi: 10.1109/ACCESS.2022.3150281.

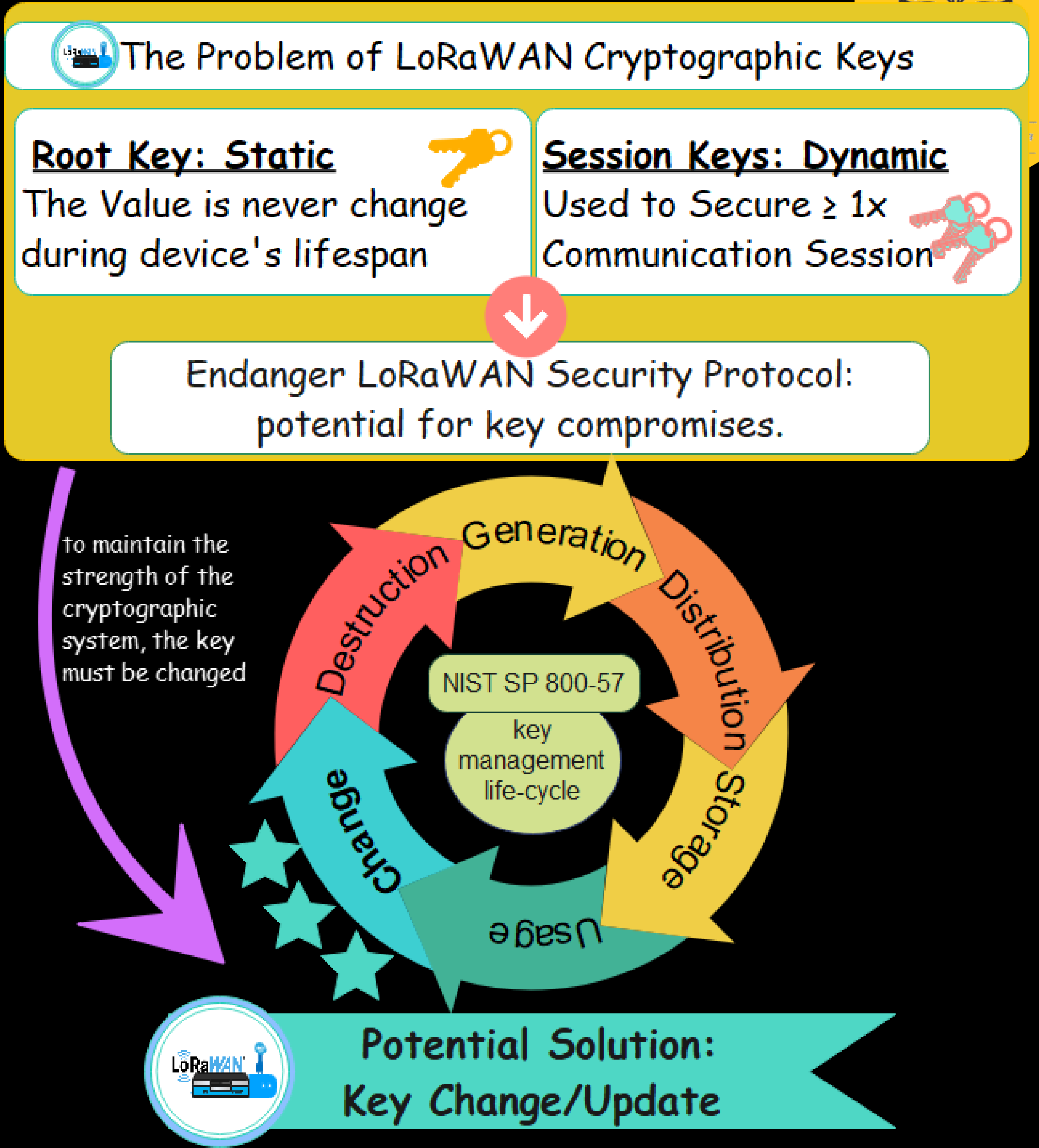
LoRaWAN Security Issues

Root Key

- Root Key is LoRaWAN Master key
- Root Key is the LoRaWAN principal key used to derive all other cryptographic keys
- **Root Key issues :** The root key value **remains the same** throughout the device's lifespan, implying that its crypto period exceeds the recommended value

Session Key

- Session Key is a derivation key used to secure communication and payload transmission.
- **Session Key issue**: LoRaWAN apply the **same session** key to secure **multiple communication sessions** – Key repetition leads to data leakage when it is compromised.



LoRaWAN Security Issues

- Cryptoperiod of **Root Key** →It must be changed **at least once** a year (NIST Special Publication 800-57 Part 1 Rev. 5)
- Root Key = LoRaWAN's Master key

NIST SP 800-57

Key Type	Cryptoperiod	
	Originator-Usage Period (OUP)	Recipient-Usage Period
2. Public Signature-Verification Key	Several years (depends on key size)	
3. Symmetric Authentication Key	≤ 2 years	≤ OUP + 3 years
4. Private Authentication Key	1 to 2 years	
5. Public Authentication Key	1 to 2 years	
6. Symmetric Data Encryption Keys	≤ 2 years	≤ OUP + 3 years
7. Symmetric Key-Wrapping Key	≤ 2 years	≤ OUP + 3 years
8. Symmetric RBG Keys	See SP 800-90	–
9. Symmetric Master Key/Key Derivation Key	About 1 year	–

9. *Symmetric master key/key-derivation key:*

a. Type Considerations: A symmetric master key (also called a key-derivation key) may be used multiple times to derive other keys using a (one-way) key-derivation function or method (see Section 8.2.4). Therefore, the cryptoperiod consists of only an originator-usage period for this key type. A suitable cryptoperiod depends on the nature and use of the key(s) derived from the master key and on considerations provided earlier in Section 5.3. The cryptoperiod of a key derived from a master key could be relatively short (e.g., a single use, communication session, or transaction). Alternatively, the master key could be used over a longer period of time to derive (or re-derive) multiple keys for the same or different purposes. The cryptoperiod of the derived keys depends on their use (e.g., as a symmetric data-encryption or integrity authentication key).

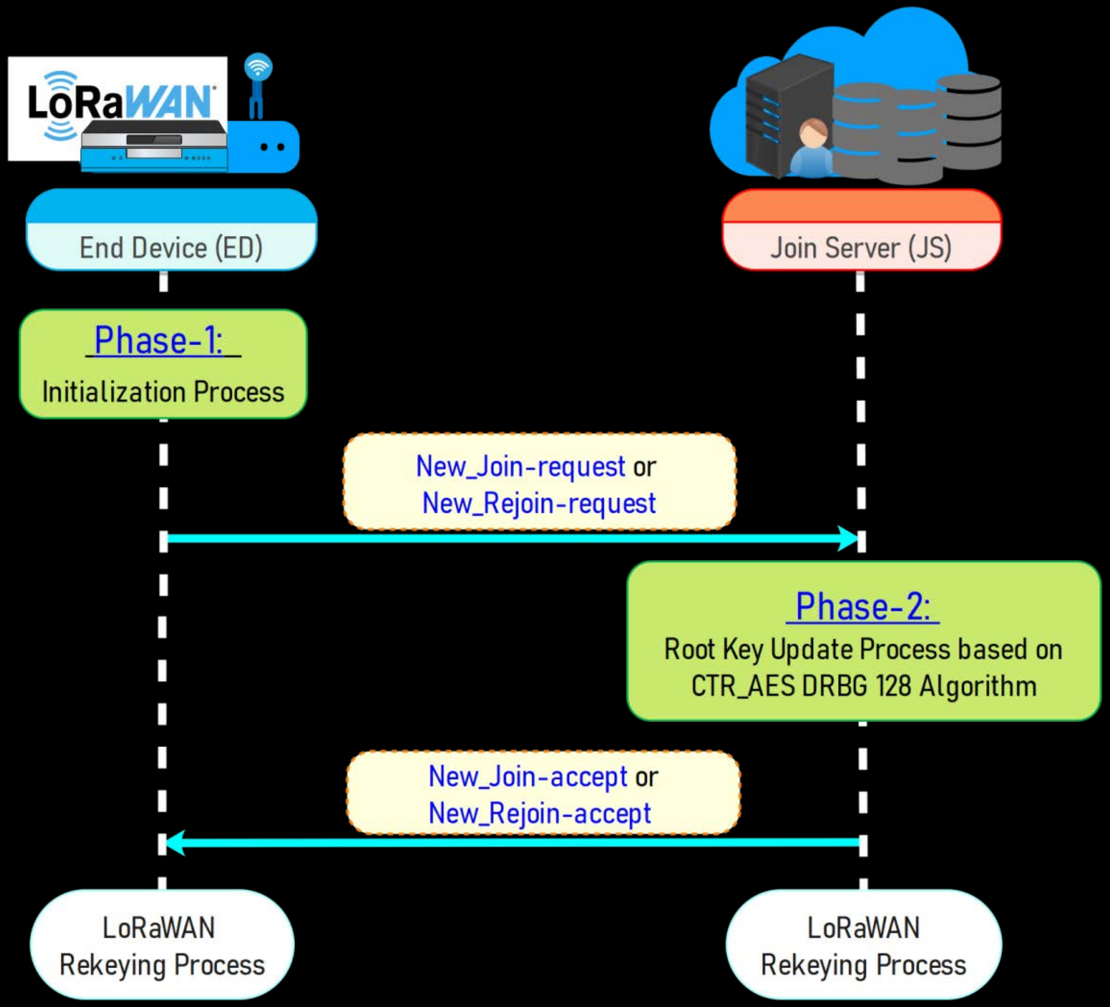
b. Cryptoperiod: An appropriate cryptoperiod for a symmetric master key might be one year, depending on its usage environment, the sensitivity/criticality of the information protected by the derived keys, and the number of keys derived from the master key.

- Cryptoperiod of **Session Key** → NIST recommends that the session key should be applied **only once** in every communication or should be **unique to each session** (NIST Special Publication 800-57 Part 3 Rev. 1)

NIST SP 800-57

A Novel Secure Root Key Updating Scheme Based on CTR_AES DRBG 128

Novel Secure Root Key Updating Scheme for LoRaWANs Based on CTR_AES DRBG 128



Pict 3.1 General Architecture of LoRaWAN's root key update

- The involved Entities
  1. ED
  2. JS
- Scheme
  - ❖ *Time-driven: Periodic Update*
- Phases
  - ❖ Phase-1: Initialization at ED
  - ❖ Phase-2: Root Key update process at JS
- Communication Protocol
  - ❖ *New_Join-request & New_Rejoin-request*
  - ❖ *New_Join-accept & New_Rejoin-accept*
- Root Key Update Algorithm
  - ❖ CTR_AES DRBG 128 bit
  - ❖ *Input: Key + Counter* generated by RBG module complied to FIPS 140 standard
  - ❖ *Output: New Root Key*

Phase 1: Initialization Process of *Root key Update*

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
EST. 1849

End Device (ED) | **Join Server (JS)**

Phase-1: Initialization Process

1. Retrieve scheduled *ED*'s Timestamp, *Ts*
2. Retrieve counter's value ($0 <= 2^{16} - 1$)
   If ($Count = 0$) ; $Count = DevNonce$
   else ($0 < Count < 2^{16} - 1$) ; $Count = RJount1$

3. Calculate MIC_{EJ} of New_Join-request or New_Rejoin-request message
   if $Count = DevNonce$
   - $cmac_j = aes128cmac(NwkKey, MHDR\text{-}ED \mid JoinEUI \mid DevEUI \mid DevNonce \mid Ts)$
   - $MIC_{EJj} = cmac_j[0..3]$

   else
   - $JSIntKey = aes128\_encrypt(NwkKey, 0x06 \mid DevEUI \mid pad16)$
   - $cmac_r = aes128\_cmac(JSIntKey, MHDR_{ED} \mid ReJoin\ Type1 \mid JoinEUI \mid DevEUI \mid RJcount1 \mid Ts)$
   - $MIC_{EJr} = cmac_r[0..3]$

4. Send the *New_Join-request* or *New_Rejoin-Request* message
   - *New_Join-request* = $\{MHDR_{ED}, (JoinEUI, DevEUI, DevNonce, Ts), MIC_{EJj}\}$
   - *New_Rejoin-request* = $\{MHDR_{ED}, (ReJoin\ Type1, JoinEUI, DevEUI, RJCount1, Ts), MIC_{EJr}\}$

$\{MHDR_{ED}, (JoinEUI, DevEUI, DevNonce, Ts), MIC_{EJj}\}$ or
$\{MHDR_{ED}, (ReJoin\ Type1, JoinEUI, DevEUI, RJCount1, Ts), MIC_{EJr}\}$

Phase 2: Root key Update Process

- *New_Root_Key=CTR_AES DRBG_128bits (Key, Nonce_Count|DevNonce)*

  or

- *New_Root_Key=CTR_AES DRBG_128bits(Key, Nonce_Count|RJCount1)*

Phase-2: Root Key Update Process based on *CTR_AES DRBG 128*

1. Calculate the MIC_{EJj} or MIC_{EJr}

2. Retrieve *Ts'*, *JS*'s scheduled timestamp of the related *ED*, and check *Ts'-Ts* $\leq \Delta Ts$
   - if the MIC calculation and ΔTs is correct, then
     -- Store current *NwkKey* as *NwkKey_old*;
     -- Store current *JSIntKey* as *JSIntKey_old*;
     -- Retrieve a counter value from *DevNonce* or *RJCount1*
     -- Store the *JSEncKey* of the *ED* as *JSEncKey_old* ; *JSEncKey* = *aes128_encrypt(NwkKey, 0x05 | DevEUI | pad16)*
   - if incorrect send notification to ED to retry the New_Join-request or New_Rejoin-request procedure.

3. Instruct Random Bit Generator to generate 2 value *Pseudo Random Bit Sequence*: 128 bits and 112 bits (*Nonce_Count*)

4. Assign the input parameter
   - *Key* = 128 *Pseudo Random Bit Sequence*
   - *Counter* = 112 bits *Nonce_Count* | 16 bits value of *DevNonce* or *RJCount1*

5. Calculate
   - *New_Root_Key* = *CTR_AES DRBG 128(Key, Nonce_Count | DevNonce)* or
   - *New_Root_Key* = *CTR_AES DRBG 128(Key, Nonce_Count | RJCount1)*

6. Calculate Context and MIC_{JE}
   - *JContext* = *JoinEUI | DevNonce |* $MHDR_{JS}$ *| JoinNonce | NetID | DevAddr | DLSettings | RxDelay | CFList*
   - *RContext* = *JoinEUI | RJCount1 |* $MHDR_{JS}$ *| JoinNonce | NetID | DevAddr | DLSettings | RxDelay | CFList*

   To respond *New_Join-request*:
   - $cmac_j$ = *aes128_cmac(JSIntKey_old, 0xFF| JContext | New_Root_Key)*
   - MIC_{JEj} = $cmac_j$*[0..3]*

   To respond New_Rejoin-request:
   - $cmac_r$ = *aes128_cmac(JSIntKey_old, 0x01| RContext | New_Root_Key)*
   - MIC_{JEr} = $cmac_r$*[0..3]*

7. Calculate *JMessage* and Encrypt the *New_Join-accept* or *New_Rejoin-accept* using AES 128 decrypt operation in ECB mode
   - *JMessage* = *JoinNonce | NetID | DevAddr | DLSettings | RxDelay | CFList*
   - *New_Join-accept* = *aes128_decrypt(NwkKey_old, JMessage | New_Root_Key |* MIC_{JEj}*)*
   - *New_Rejoin-accept* = *aes128_decrypt(JSEncKey_old, JMessage | New_Root_Key |* MIC_{JEr}*)*

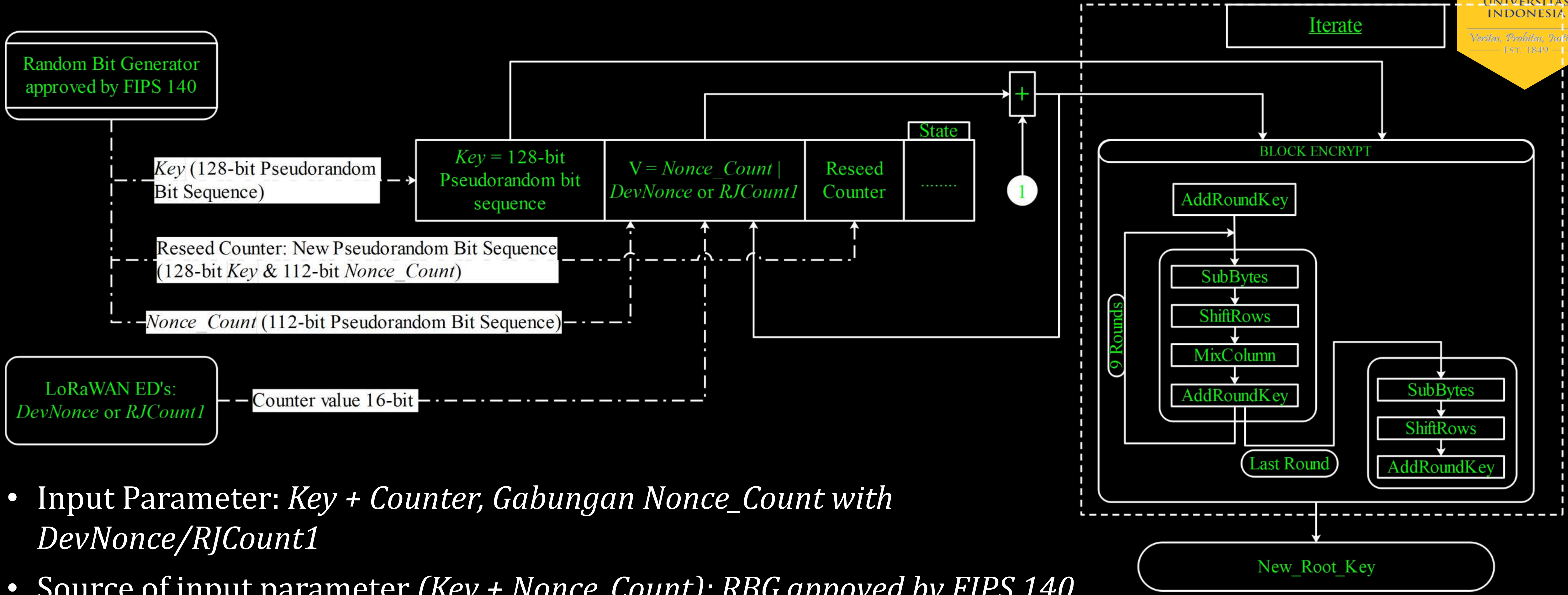
8. Send the encrypted *New_Join-accept* or *New_Rejoin-accept*

{$MHDR_{JS}$, *aes128_decrypt(NwkKey_old, JMessage | New_Root_Key |* MIC_{JEj}*)*} or
{$MHDR_{JS}$, *aes128_decrypt(JSEncKey_old, JMessage | New_Root_Key |* MIC_{JEr}*)*}

LoRaWAN Rekeying Process

LoRaWAN Rekeying Process

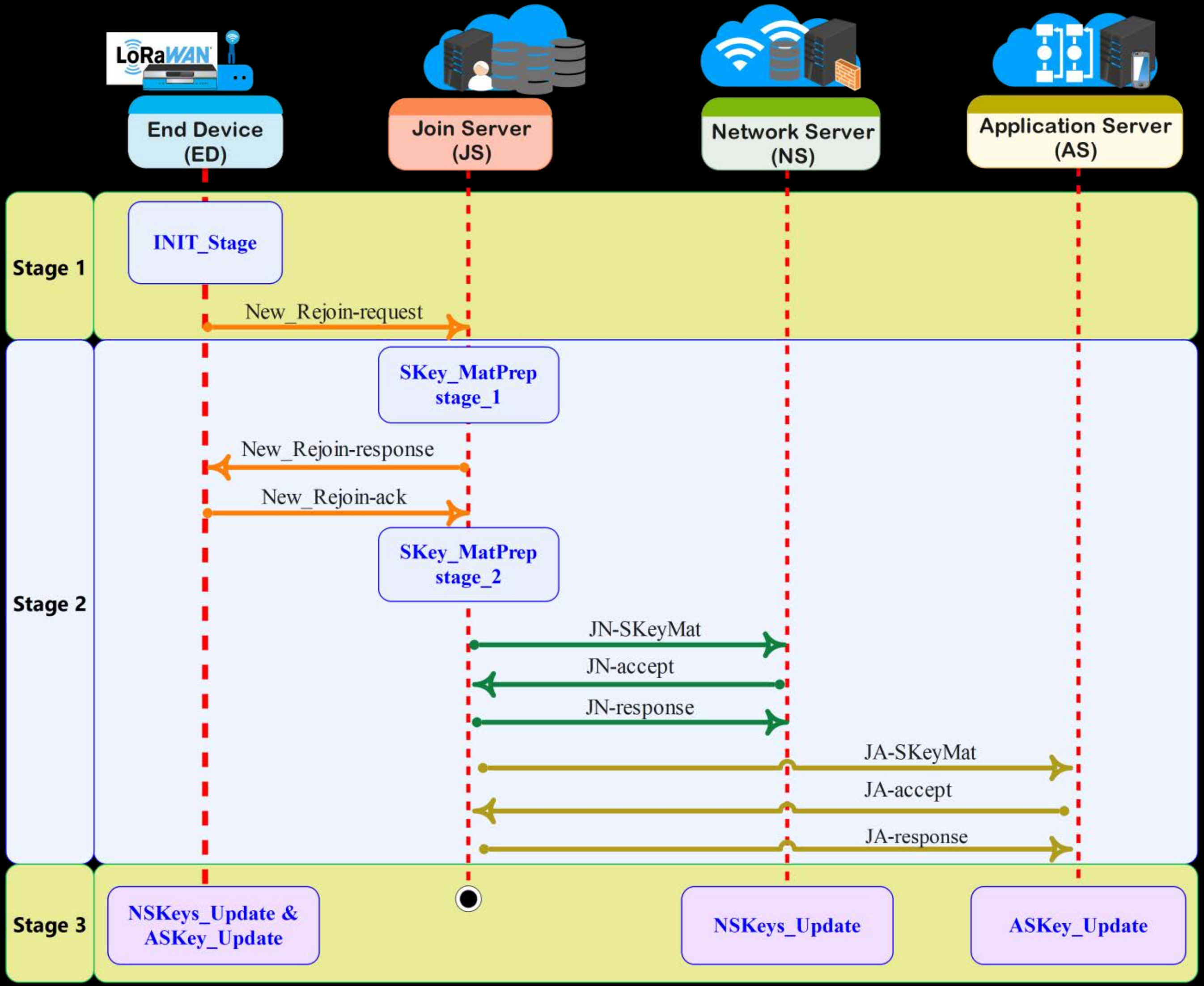
Algorithm Design of The CTR_AES DRBG 128-bits



- Input Parameter: *Key + Counter, Gabungan Nonce_Count with DevNonce/RJCount1*
- Source of input parameter *(Key + Nonce_Count): RBG appoved by FIPS 140*
- Reseed counter dijalankan setiap 2^16 – 1
- *Internal state (block encrypt) : CTR_AES 128-bits*
- *Algorithm output*: *New_Root_Key*

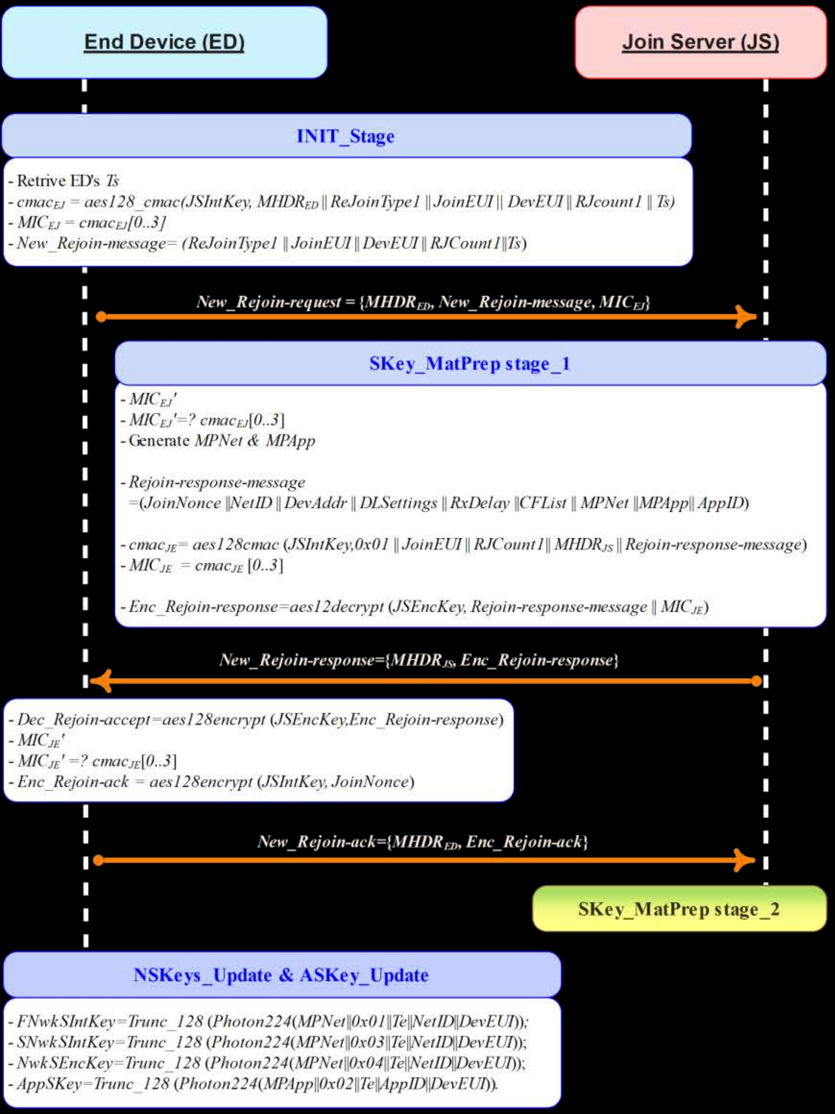
A Novel Session Key Update Scheme Based on Truncated Photon-256

General Architecture: Session Key Update Scheme based on Truncated Photon-256

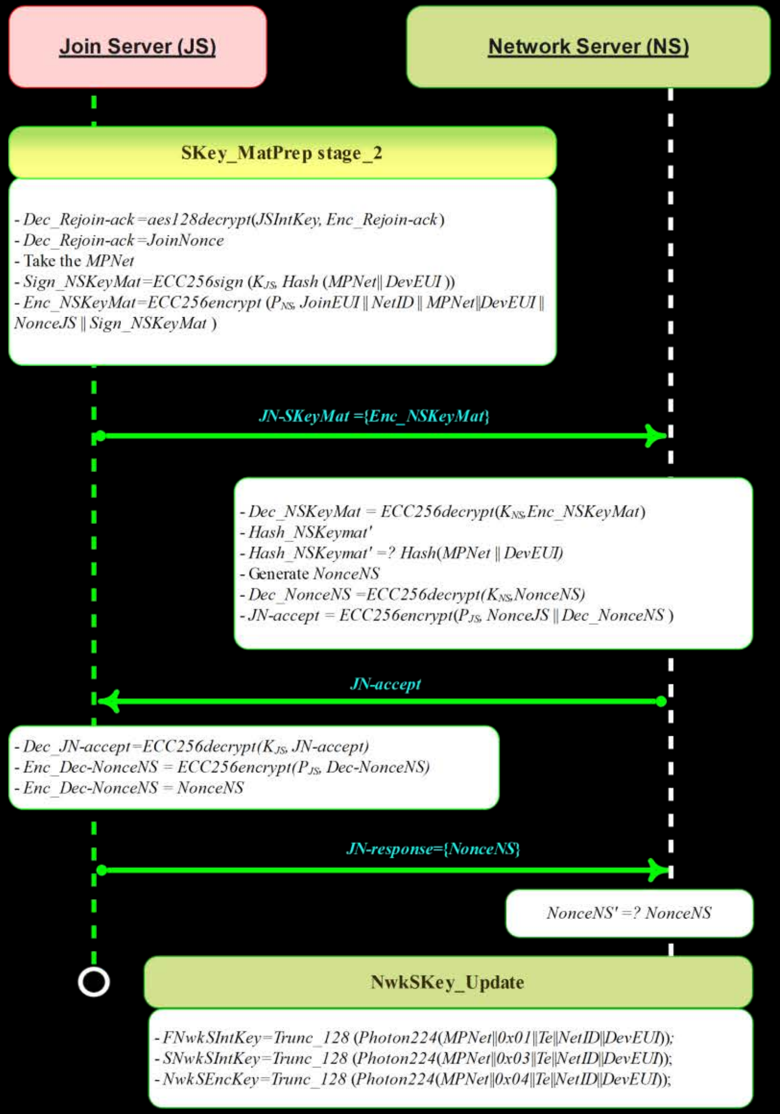


- Proposed Approach
  - ❖ *Time-driven: Periodic Update*
- Entities involved in the scheme:
  - ❖ End Device (ED), Join Server (JS), Network Server (NS), Application Server (AS)
- The scheme consists of three stages
  1. *INIT_Stage* occurs at ED
  2. *SKey_MatPrep* occurs at JS
  3. *NSKey_Update* & *AS_KeyUpdate* occur at ED, NS, AS
- Communication Protocol between ED-JS
  - ❖ *New_Rejoin-request*
  - ❖ *New_ReJoin-response*
  - ❖ *New_Rejoin-ack*
- Communication Protocol between JS-NS & JS-AS
  - ❖ *JN-SKeyMat & JA-SKeyMat*
  - ❖ *JN-accept & JA-accept*
  - ❖ *JN-response & JA-response*

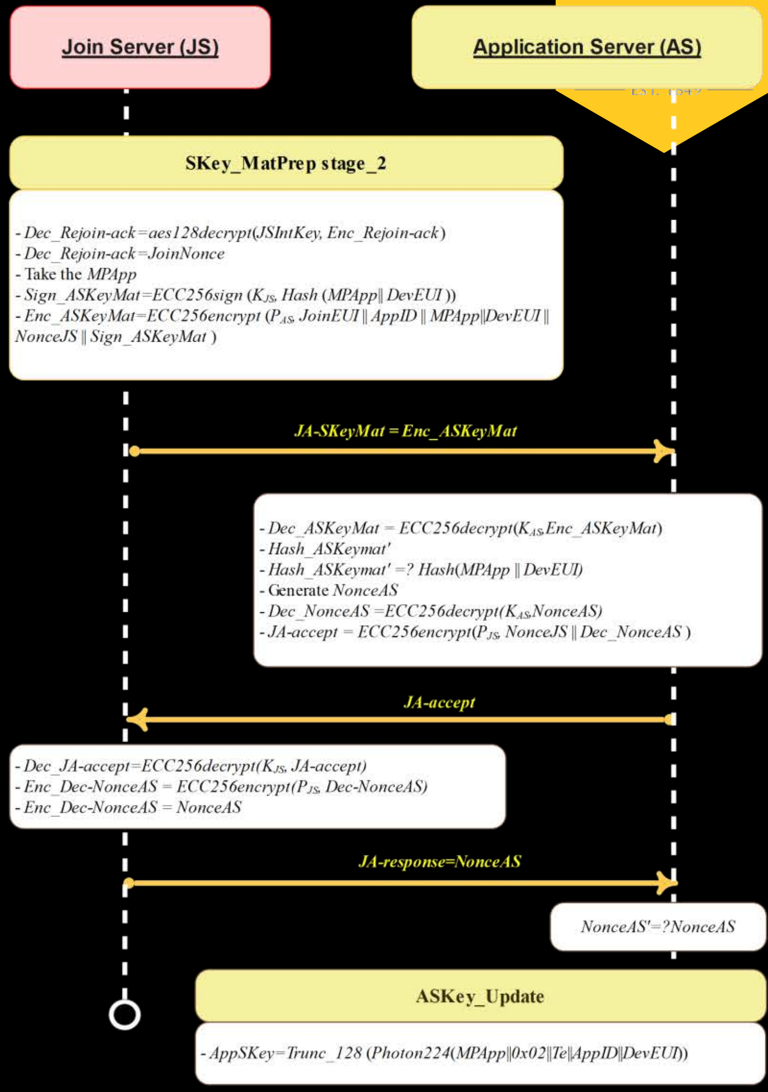
Communication Session between ED-JS, JS-NS & JS-AS



Between ED-JS



Between JS-NS



Between JS-AS

Truncated Photon-256 Algorithm of NSKey_Update & ASKey_Update

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
EST. 1849

➢NSKey_Update

- *FNwkSIntKey=Trunc_128 (Photon-256 (MPNet||0x01||Te||NetID||DevEUI));*
- *SNwkSIntKey=Trunc_128 (Photon- 256 (MPNet||0x03||Te||NetID||DevEUI));*
- *NwkSEncKey=Trunc_128 (Photon-256 (MPNet||0x04||Te||NetID||DevEUI));*

➢ASKey_Update

- *AppSKey=Trunc_128 (Photon- 256 (MPApp||0x02||Te||AppID||DevEUI)).*

Research overview and Conclusions

Problem Statements

Section 1

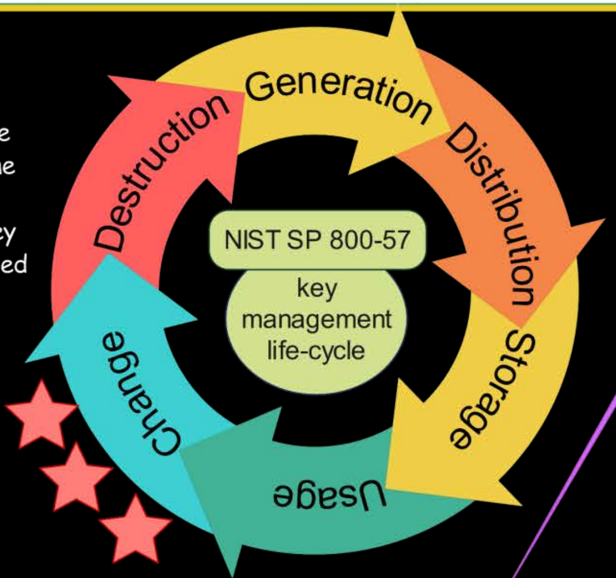
Manajemen kunci kriptografi LoRaWAN

Root Key: Static
The Value is never change during device's lifespan

Session Keys: Dynamic
Used to Secure ≥ 1x Communication Session

Endanger LoRaWAN Security Protocol: potential for key compromises.

to maintain the strength of the cryptographic system, the key must be changed



Potential Solution: Key Update

Proposed Solutions

Section 2.1
A Novel Secure Root Key Updating Scheme Based on CTR_AES DRBG 128

→ Section 5
Change the Root Key from Static to Dynamic

Section 2.2
A Novel Session Key Update Scheme Based on Truncated Photon-256

→ Section 5
Change Session Key from used ≥ 1x session to unique key for each communication session.

Result & Discussion

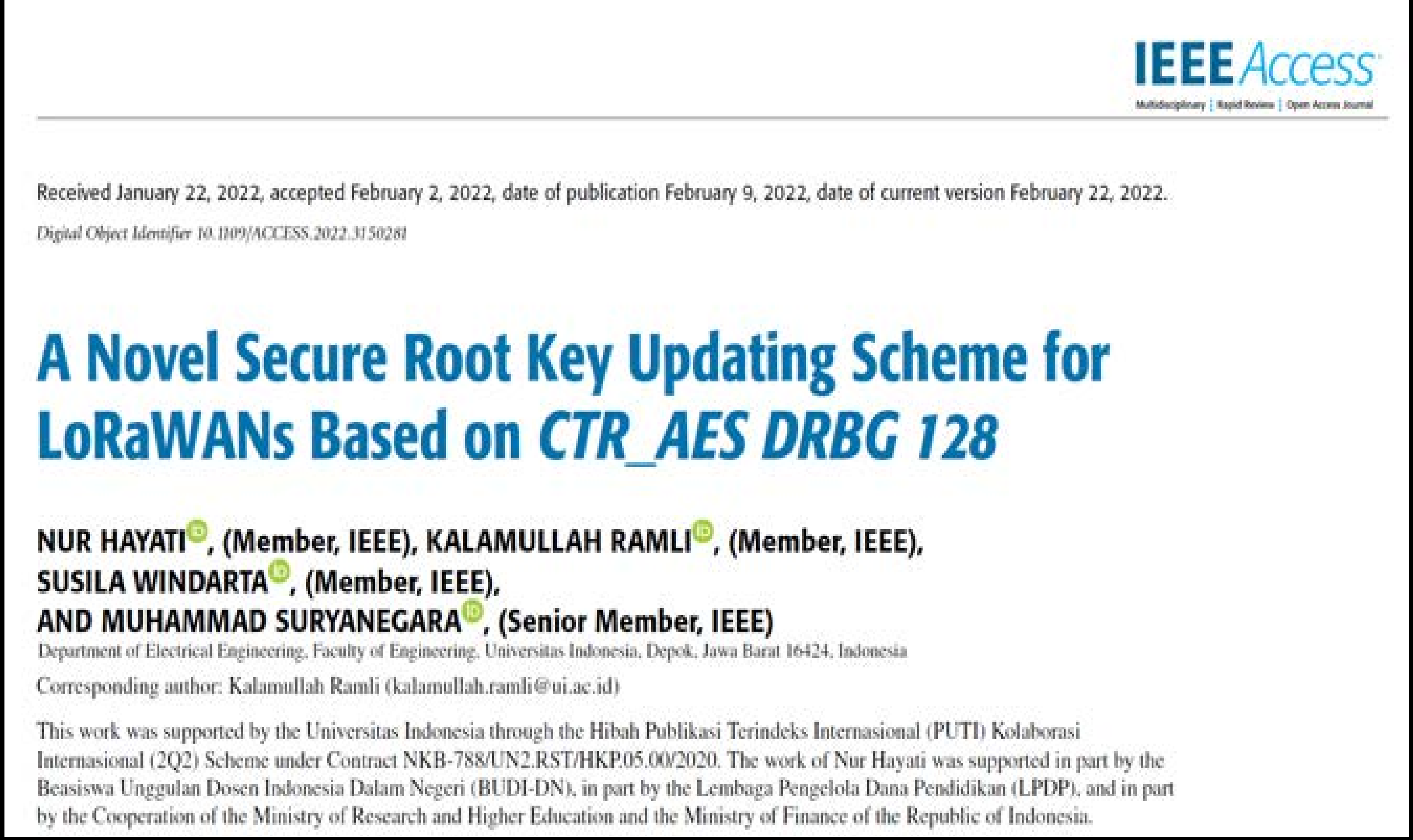
Section 3 & 4

The Analysis of Cryptographic Key Update

- Randomness test of Key bit sequences - NIST 802-22
- Security Communication Protocol Test - Scyther tools
- Formal Security Protocol Analysis - GNY Logic
- Informal Security Protocol (Integrity Protection, Replay attack resistance, Perfect forward secrecy, End-to-end data secrecy, Mutual authentication)
- Performance (Computation cost, Updating method, Implementation compatibility, Communication cost, Storage)

N. Hayati, K. Ramli, S. Windarta, M. Suryanegara, “A Novel Secure Root Key Updating Scheme for LoRaWANs Based on CTR_AES DRBG 128,” IEEE Access, vol. 10, pp. 18807–18819, 2022,

Doi: 10.1109/ACCESS.2022.3150281.



IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 22, 2022, accepted February 2, 2022, date of publication February 9, 2022, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150281

A Novel Secure Root Key Updating Scheme for LoRaWANs Based on *CTR_AES DRBG 128*

NUR HAYATI, (Member, IEEE), KALAMULLAH RAMLI, (Member, IEEE), SUSILA WINDARTA, (Member, IEEE), AND MUHAMMAD SURYANEGARA, (Senior Member, IEEE)

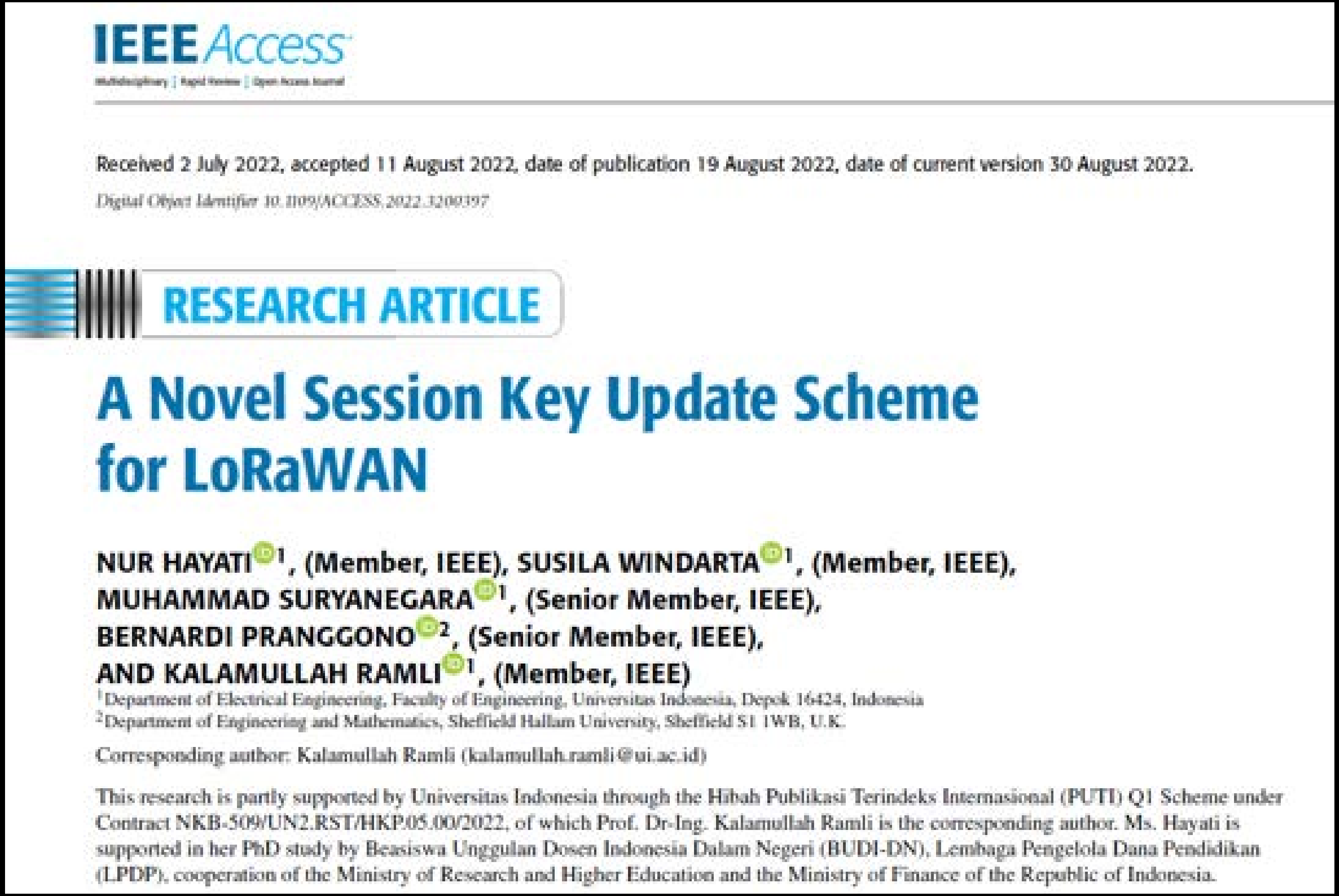
Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Jawa Barat 16424, Indonesia

Corresponding author: Kalamullah Ramli (kalamullah.ramli@ui.ac.id)

This work was supported by the Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Kolaborasi Internasional (2Q2) Scheme under Contract NKB-788/UN2.RST/HKP.05.00/2020. The work of Nur Hayati was supported in part by the Beasiswa Unggulan Dosen Indonesia Dalam Negeri (BUDI-DN), in part by the Lembaga Pengelola Dana Pendidikan (LPDP), and in part by the Cooperation of the Ministry of Research and Higher Education and the Ministry of Finance of the Republic of Indonesia.

N. Hayati, S. Windarta, M. Suryanegara, B. Pranggono and K. Ramli, "A Novel Session Key Update Scheme for LoRaWAN," in IEEE Access, vol. 10, pp. 89696-89713, 2022,

Doi: 10.1109/ACCESS.2022.3200397



IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 July 2022, accepted 11 August 2022, date of publication 19 August 2022, date of current version 30 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3200397

RESEARCH ARTICLE

A Novel Session Key Update Scheme for LoRaWAN

NUR HAYATI[1], (Member, IEEE), SUSILA WINDARTA[1], (Member, IEEE), MUHAMMAD SURYANEGARA[1], (Senior Member, IEEE), BERNARDI PRANGGONO[2], (Senior Member, IEEE), AND KALAMULLAH RAMLI[1], (Member, IEEE)

[1]Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok 16424, Indonesia
[2]Department of Engineering and Mathematics, Sheffield Hallam University, Sheffield S1 1WB, U.K.

Corresponding author: Kalamullah Ramli (kalamullah.ramli@ui.ac.id)

This research is partly supported by Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Q1 Scheme under Contract NKB-509/UN2.RST/HKP.05.00/2022, of which Prof. Dr-Ing. Kalamullah Ramli is the corresponding author. Ms. Hayati is supported in her PhD study by Beasiswa Unggulan Dosen Indonesia Dalam Negeri (BUDI-DN), Lembaga Pengelola Dana Pendidikan (LPDP), cooperation of the Ministry of Research and Higher Education and the Ministry of Finance of the Republic of Indonesia.

THANK YOU