Key Agreement Protocol for EPON

TaeWhan Yoo (twyoo)*, KyungSoo Han (kshan)*, KwangJo Kim (kkj)**,

SuGil Choi (sooguri)**, SungJun Min (sjmin)**

* Electronics and Telecommunications Research Institute (@etri.re.kr)
** Information and Communications University (@icu.ac.kr)

Contents

1. Introduction

- Problems
  - Single OLT is connected to many ONUs
  - Each ONU can listen to all downstream traffic
  - There is a possibility that each ONU may listen to upstream traffic
  - Each communication needs to be secured

- Goals
  - Mutual Authentication between OLT and each ONU
  - Agreement of key to protect communication in EPON

- Definition of Key Agreement
  - A key establishment protocol whose resultant keying material is a function of information contributed by two or more participants, so that no party can predetermine the value of the keying material

2. Requirements (1)

- Cryptographic Requirements
  - Key Freshness
    - » A key is fresh if it can be guaranteed to be new, as opposed to possibly an old key being reused through the actions of either an adversary or authorized party
  - Forward Secrecy and Backward Secrecy
    - » The exposure of current key doesn't affect the security of past and future protocol
  - Encryption of Secret Information in Message
    - » Secret message should be encrypted if there is a possibility of eavesdropping on the message

2. Requirements (2)

- Cryptographic Requirements (contd.)
  - Message Protection
    - » Source authentication
    - » Integrity check
    - » Replay attack prevention

- System Requirements
  - Compatibility with EPON
    - » EPON is a link layer protocol and EPON system has the functionality only up to link layer, so key agreement protocol should operate at link layer
    - » Existing PKI (Public Key Infrastructure) can't be used since CRL (Certificate Revocation List) checking happens at application layer

2. Requirements (3)

- Performance Requirements
  - Computational Efficiency
    - » Protocol is run infrequently. Slight increase of computational complexity doesn't cause a problem
    - » If possible, precomputes parts of protocol elements
  - Communication Efficiency
    - » Transmission of one frame is very quick
    - » Fragmentation and Reassembly of frame degrade performance. So, the size of one message must be less than EPON MTU (Maximum Transmission Unit)
  - Fast Reconnection
    - » Ability to create new key with lower overhead

3. Protocol (Master Key Creation)

- Master Key Sharing
  - Offline setting
- Master Key Usage
  - Encryption of keys before storing
  - Input value of keyed hash function in rekeying and key update process
- Assumptions
  - Master key update is done manually without causing service interruption
  - In the protocol, master key update is not considered

3. Protocol (Key Sharing, 1)

- Key Types
  - Master key (*MK*)
    - » A master key is used for a long time
    - » Master key is used only for key derivation
  - Primary key (PK_i: *i*-th primary key derived from *MK*)
    - » A primary key is derived from a master key through rekeying process
    - » A primary key is relatively long-lived and generates multiple secondary keys
  - Secondary key (SK_j: *j*-th secondary key derived from current *PK*)
    - » Secondary key (session key) is used for communication protection
    - » The cryptoperiod of a secondary key is short, e.g., a communication session

3. Protocol (Key Sharing, 2)

- Our key sharing protocol consists of Rekeying and Key Update

- Reasons for Replacing a Key
  - Key compromise or key exposure
  - Expiration of the key's cryptoperiod
  - Limited amount of data protected with any given key

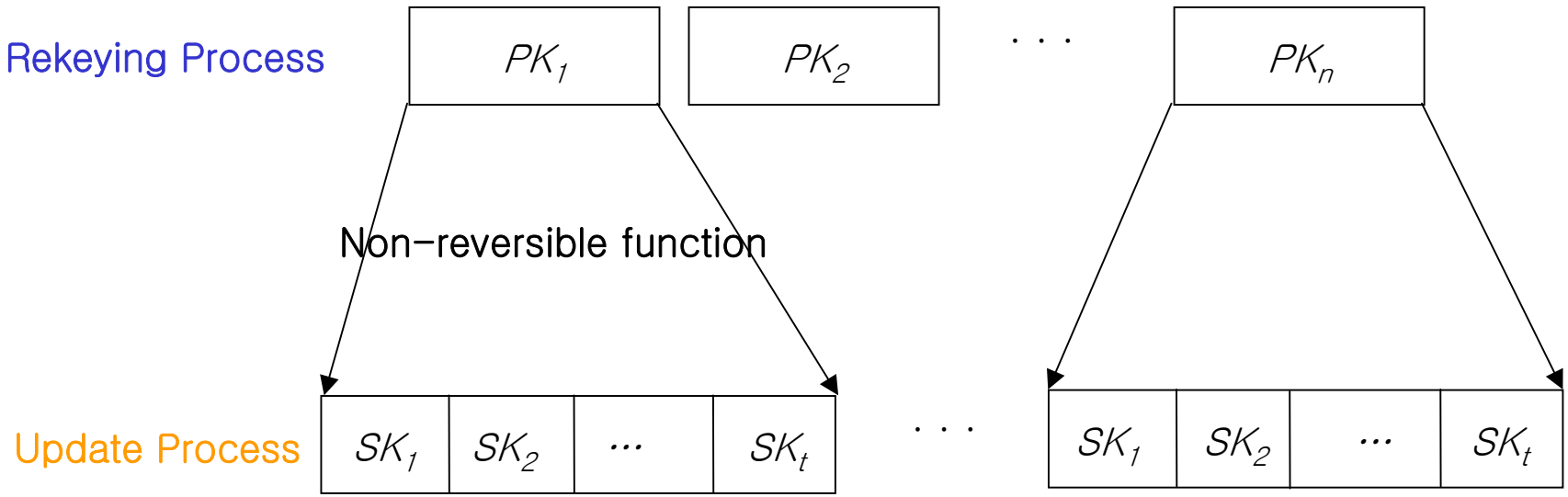
- Rekeying
  - It is used when a key has been compromised or cryptoperiod is almost over

3. Protocol (Key Sharing, 3)

- Key Update
  - Employ a non-reversible function to the old key and other data
  - do not require the exchange of any new information between the entities
  - limit the amount of data protected by a single key

3. Protocol (Key Sharing, 4)

❑ Relationship between Rekeying and Key Update



- PK_i : i-th primary key derived from given master key through rekeying process
- SK_j : j-th secondary key derived from current PK through update process

3. Protocol (Key Sharing, 5)

❑ Notation

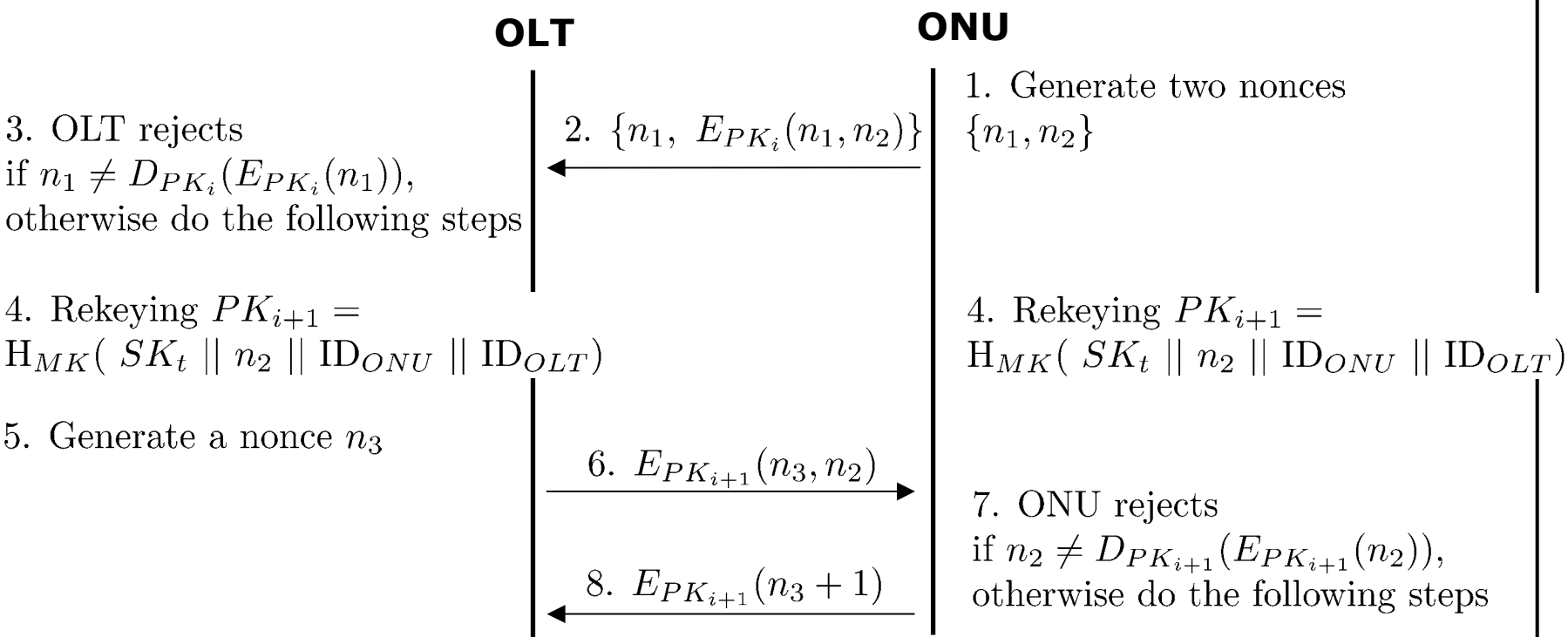
- n : nonce (random value)
- $||$: concatenation
- $E_K(M)$: encryption of message M using key K
- $D_K(M)$: decryption of message M using key K
- $\mathrm{H}_K(M)$: Keyed hash function. It accepts a secret key K and a message M, and generates a fixed length pseudo random output
- ID_x : Identity of x, where $x \in \{\mathrm{OLT}, \mathrm{ONU}_1, \mathrm{ONU}_2, \cdots, \mathrm{ONU}_n\}$

❑ Protocol Assumption

- Our protocol describes key agreement between OLT and one ONU only using symmetric key cryptosystem
- OLT shares Master key (*MK*) and primary key (*PM*) with ONU

3. Protocol (Key Sharing, 6)

❑ Rekeying Process



- PK_{i+1} : $(i+1)$-th primary key derived from i-th primary key
- SK_j : j-th secondary key derived from current primary key

3. Protocol (Key Sharing, 7)

- Key Update Process
  - OLT and ONU should synchronize the *index information* to update secondary key
  - The algorithm for *j*-th secondary key generation from *i*-th primary key

$$SK_j = \mathrm{H}_{MK}(SK_{j-1} \,||\, j \,||\, \mathrm{ID}_{ONU} \,||\, \mathrm{ID}_{OLT} \,||\, Others)$$

- $1 \leq j \leq t$ (j denotes *index information*)
- t denotes maximum number of secondary keys derived from one primary key
- $SK_0 = \text{current} \;\; PK$
- $Others$ means any other shared information

4. Protocol Analysis (1)

- Cryptographic Requirements
  - Key Freshness
    - » Achieved by rekeying and key update process
    - » Nonces used in rekeying process serve to offer this property
  - Forward Secrecy
    - » Assume that the current secondary key is compromised
      - – The attacker cannot compute the previous secondary keys because the function (H) used in the key update process is non-reversible.
      - – Previous primary key has nothing to do with current secondary key
    - » Assume that the current primary key is compromised
      - – The attacker cannot compute the previous secondary keys and primary keys because the function (H) is used in key update process and rekeying is non-reversible

4. Protocol Analysis (2)

- Cryptographic Requirements (contd.)
  - Backward Secrecy
    - » Attacker cannot compute the following primary keys and secondary keys without the knowledge of master key although he knows current primary key or secondary key
    - » This protocol guarantees *partial backward secrecy*, because backward secrecy depends on only master key
  - Encryption of Secret Information in Message
    - » There is no secret message transmitted over network. To enhance security, n_2 is encrypted and never appears in plaintext during transmission

4. Protocol Analysis (3)

- Cryptographic Requirements (contd.)
  - Message Protection
    - » Source authentication (Y): the only one who knows primary key can encrypt or decrypt message correctly
    - » Integrity check (Y): if a message is modified illegally, the decryption of the message will not contain expected nonces like n_1, n_2, and n_3
    - » Replay attack prevention (Y): When OLT receives a rekeying request message, it generates new primary key. The authentication of replayed message fails, as OLT applies new primary key. OLT and ONU must record received nonces to prevent replaying of confirmation message

4. Protocol Analysis (4)

- Performance Requirements
  - Communication Efficiency
    - » The length of each message is less than 1000 bits (< MTU), so fragmentation and reassembly are unnecessary (assuming nonce is 128 bit long)
  - Computational Efficiency
    - » Requires just one symmetric key cryptography operation or keyed pseudo random function
    - » *t* secondary keys are derived from one primary key
    - » It is possible to precompute *t* secondary keys
  - Fast Reconnection
    - » There is no time delay for computing new secondary key

4. Protocol Analysis (5)

- System Requirement
  - Compatibility with EPON
    - » Can be performed at link layer
    - » Excluded the use of PKI

5. Conclusions

- Proposed a key agreement protocol only using symmetric key cryptosystem
- Our protocol is a key agreement protocol between OLT and a specified ONU
- This protocol satisfies important cryptographic requirements as well as performance requirements in EPON system
- Group key agreement protocol is open problem