Workgroup: I2NSF Working Group
Internet-Draft:
draft-yang-i2nsf-remote-attestation-interface-
dm-01
Published: 6 June 2022
Intended Status: Standards Track
Expires: 8 December 2022
Authors: P. Yang M. Chen L. Su
China Mobile China Mobile China Mobile
D. Lopiz J. Jeong L. Dunbar
Telefonica Sungkyunkwan University Futurewei

I2NSF Remote Attestation Interface YANG Data Model

Abstract

This document describes the architecture and corresponding interfaces of NSF remote attestation in I2NSF framework. Remote attestation of NSFs could provide integrity assurence of NSFs deployed in remote environment. The interfaces involved are I2NSF remote attestation evidence interface, I2NSF remote attestation reference value interface, and I2NSF remote attestation result interface. This document complies with I2NSF architecture and Remote Attestation ProcedureS (RATs) architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 December 2022.

Table of Contents

1. Introduction
2. Terminology
  2.1. Terms
  2.2. Requirements Language
3. Scope and Motivation
  3.1. Scope
  3.2. Motivation
4. Information Model
  4.1. Architecture of I2NSF Remote Attestation
  4.2. Root of Trust in I2NSF
  4.3. Verifier in I2NSF
  4.4. Reference Value Provider in I2NSF
  4.5. Relying Party in I2NSF
  4.6. Endorser in I2NSF
5. Data Model of I2NSF Remote Attestation
  5.1. I2NSF Remote Attestation Evidence Interface
    5.1.1. YANG Tree Diagram of I2NSF Remote Attestation Evidence Interface
    5.1.2. YANG Data Model of I2NSF Remote Attestation Interface
  5.2. I2NSF Remote Attestation Reference Value Interface
    5.2.1. YANG Tree Diagram of I2NSF Remote Attestation Reference Value Interface
    5.2.2. YANG Data Model of I2NSF Remote Attestation Reference Value Interface
  5.3. I2NSF Remote Attestation Result Interface
6. IANA Considerations
7. Security Considerations
8. References
  8.1. Normative Reference
  8.2. Informative Reference
Authors' Addresses

1. Introduction

In terms of implementation, NSFs are usually deployed in remote scenarios, where it is hard to guarantee if the deploy environment is secure and the NSFs are properly deployed. If the deploy environment or the NSF is compromised, the behavior and the feedback of NSFs cannot be trusted.

Remote attestation procedure [I-D.ietf-rats-architecture] provides an efficient mechanism that a verifier likes Network Operator Mgmt System could appraise if the NSFs and the basic platforms are trusted. The general remote attestation procedure has been defined by RATs working group, however specific interfaces and implementations still need to be determined in I2NSF. This document aims to create a unified remote attestation architecture for I2NSF to enable remote attestation of NSF.

This document follows the definition of I2NSF framework[RFC8329] and also could be treated as a usecase of RATs. In detail, this document refers to the definition of RATs architecture to add new remote attestation components in I2NSF. This document refers to [I-D.ietf-rats-yang-tpm-charra] and [I-D.ietf-rats-tpm-based-network-device-attest]to define I2NSF YANG model for evidence in TPM-based platform. This document refers to [I-D.ietf-rats-eat] to define I2NSF YANG model for evidence in TEE-based platform. This document refers to [I-D.ietf-rats-ar4si] to defien I2NSF YANG model for attestation result.

2. Terminology

2.1. Terms

RATs: Remote Attestation Procedure

RoT: Root of Trust

TPM: Trusted platform module

TEE: Trust Execution Environment

RVP: Reference Value Provider

2.2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Scope and Motivation

3.1. Scope

The scope of this document focuses on the architecture and expanded interfaces of NSF remote attestation in I2NSF. The details of how to implement measurement or how to collect remote attestation evidence is out of scope.

3.2. Motivation

The architecture of I2NSF aims to provide network security functions to network users. Usually the NSFs are in remote environment and the platforms to deploy these NSFs may not be trusted. As a consequence this will bring several potential threats to I2NSF framework, some examples are shown below. The first threat is malfunction of NSF. The inappropriate deployment of NSF or the defective platform in where runs NSF will affect the behaviour of NSF directly. The second threat is the leak of digital asset like policy rules and security intelligence which is provided by the Network Operator Mgmt System. Consider a secuiry company provides NSF in where contains lots of policy rules such as DDoS prevention, traffic filter, AI module, etc. If the platform who carrys the NSF is malicious, it could steal this digital asset and provide to other rivals or attackers. The third threat is the potential spoofing or penetration attack to the I2NSF framework. The attackers in NSF platfom could disturb the action of NSF, and feedback the fake notification or even penetrate into Network Operator Mgmt System.

The solution of these kinds of threats is also straight, which is using remote attestation to make sure the remote platform is trusted and the NSFs are well deployed. While it is true that any environment is vulnerable to malicious activity with full physical access (and this is obviously beyond the scope of this document), the application of remote attestation raises the degree of physical control necessary to perform an untraceable malicious modification of that environment.

When designing remote attestation in I2NSF, three aspects need to be considered. First, determine the remote attestation architecture of I2NSF. Second, refer to the appropriate specifications defined in RATs to create I2NSF remote attestation interfaces and YANG data models. Third, cover the heterogeneity architecture of specific trusted hardware like TPM and TEE.

4. Information Model

4.1. Architecture of I2NSF Remote Attestation

As shown in figure 1 is the remote attestation architecture in I2NSF. In order to fit into the I2NSF framework, this is not a typical background check model or passport model mentioned in RATs architecture. In this figure, the relevant interfaces are I2NSF remote attestation evidence interface, I2NSF remote attestation registration interface and I2NSF remote attestation result interface.

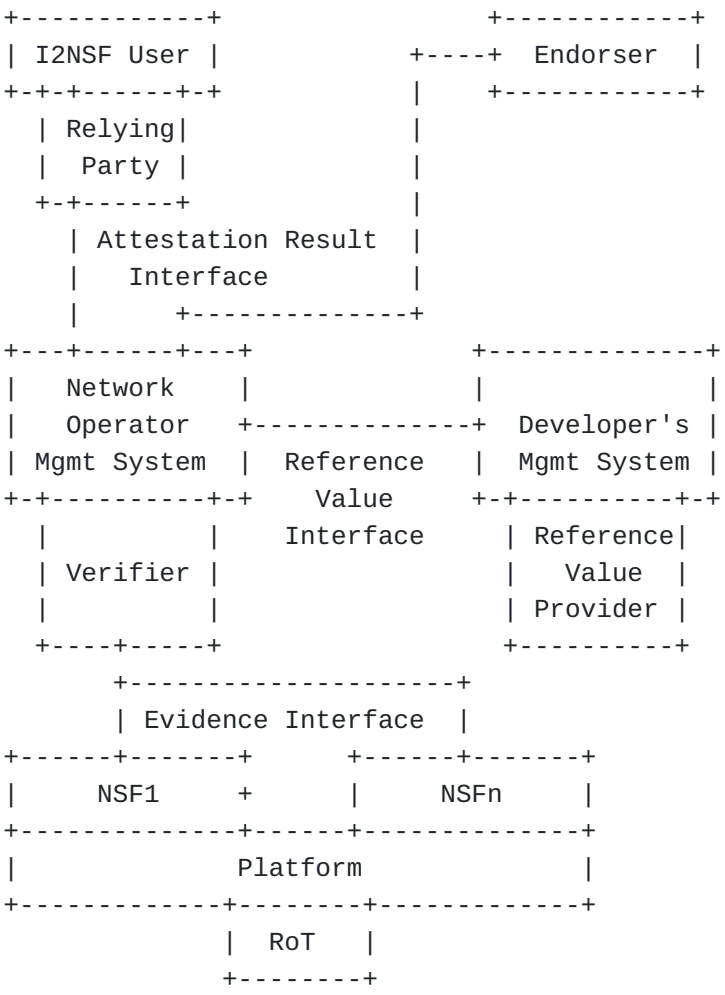

Figure 1: Architecture of Remote Attestation of I2NSF

In physical environment, NSFs exist as applications. In virtualization environment, the level-of-assurence of NFV is separated as two components: hardware and virtualization platform, VNF sotftware package[NFV-SEC-018]. When using this concept of VNF sotfware package in I2NSF, the granularity of NSF in virtualization environment is NSF and the VM it loaded in. As a result, the trust chain in virtualization environment would be RoT->virtualization layer->VM and NSF. And the trust chain in physical environment would be RoT->physical platform->NSF. Hence, we define the granularity of remote attestation in I2NSF as in Figure 2. The remote attestation procedure in I2NSF could challenge RoT, Platform and NSFs separately.

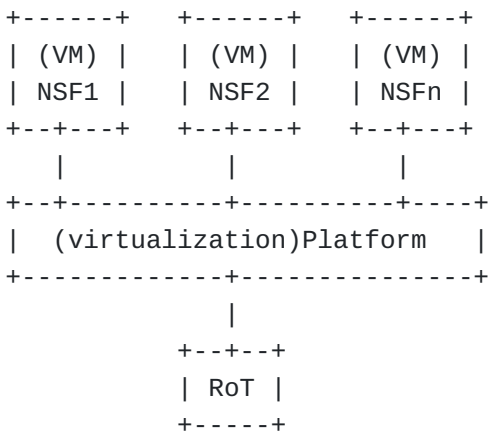

Figure 2: granularity of remote attestation in I2NSF

4.2. Root of Trust in I2NSF

Root of Trust is a hardware-based component that could provide endorsement information and relevant functions that cannot be stolen, tampered, or repudiated. RoT must be deployed in the basic hardware platform of NSF. Technologies like [TCGRoT] and [TEE] could act as RoT.

The architecture of specific RoT is out of scope of this document. But in order to depict RoT more clearly, the following segment uses TPM [tpm12][tpm20] as an example to explain how RoT works. TPM keeps an EK(Endorsement Key) to identify its identity. EK is an asymmetric root key pair, which never exposes its secret key to public. TPM derives certain AIKs(Attestation Identity Key) from EK to avoid the exposure of TPM's real identity(EK) during remote attestation. In the booting period, the TPM will record the Hash of measurement of bootloader, OS kernel and applications to a series of PCRs (Platform Configuration Registers). The value in PCRs can only be extended and cannot be tampered. If a remote attestation procedure is initiated, the PCR value will be signed by AIK and send to the verifier for attestation. The verifier then transfer the attestation result to Relying party for final decision.

4.3. Verifier in I2NSF

Verifier is deployed in Network Operator Mgmt System. Verifier is in charge of receiving attestation evidence from NSFs. Also, Verifier receives reference value from Developer's Mgmt system as benchmark.

4.4. Reference Value Provider in I2NSF

Reference Value Provider (RVP) brings the reference value of NSF remote attestation to Network Operation Mgmt System. In some conditions, the RVP could be some other venders like a blockchain, a third party security provider. So the RVP component may be an

interface that receive RVP form the third party. Or, the RVP could be deployed in Developer's Mgmt System. The reference value will be conveyed to Network Operator Mgmt System as the benchmark when verifying remote attestation evidence from attester. When the reference value needs to be collected by third party, the Reference Value Interface or other out-of-band methods in Developer's Mgmt System could be used.

4.5. Relying Party in I2NSF

Relying Party is deployed in I2NSF User. Relying Party is in charge of receiving attestation result from Verifier and generate user friendly policies to NSF User. The Relying Party does not have to know the detail of remote attestation evidence and could only focus on the final attestation result and making policies.

4.6. Endorser in I2NSF

Endorser provides the endorsement of RoT. And the verifier could use Endorser to verify the evidence information from NSF. For example, both EK and AIK in TPM are endorsed by Endorser. The communication between RoT and Endorser is based on specific RoT hardware, and usually has been setup during manufactureing. The Network Operator Mgmt System also needs to communicate with Endorser to get the endorsement of RoT before appraising attestation evidence.

5. Data Model of I2NSF Remote Attestation

The following section depicts the YANG tree diagram and YANG data model of I2NSF remote attestation interfaces.

5.1. I2NSF Remote Attestation Evidence Interface

Evidence interface focuses on the remote attestation evidence information between NSF and Network Operator Mgmt system. This interface defines notification and RPC for RoT, platform and NSFs. At present, the RoT type has two categories, one is TPM-based and the other is TEE-based like TrustZone and SGX[SGX]. The TPM-based RoT is split into TPM12 and TPM20 versions. When design this interface with TPM-based RoT, this document refers to the existing document[I-D.ietf-rats-yang-tpm-charra] to avoid unnecessary alignment work. And about the TEE-based RoT, this document refers to the EAT document[I-D.ietf-rats-eat] and uses binary format to express JWT[RFC7519]or CWT [RFC8392] in YANG data model.

In order to decouple the remote attestation result to NSF granularity, the following table defines the mapping between differnt layers. In the TPM-based remote attestation, the PCRs are arranged by specific purpose[I-D.ietf-rats-tpm-based-network-device-attest]. PCR 0-10 are responsible for the platform related remote

attestation. PCR 11-31 are responsible for the NSF remote attestation. If the platform is a virtual machine architecture, PCR 11-31 will be responsible for each virtual machine and its NSF. If the platform is a physical machines architecture, PCR 10-31 will be responsible for each NSF functions. The EAT-based platform uses Token to realize different remote attestation layers. EAT SYS Token and EAT NSF TOKEN are responsible for platform and NSF respectively.

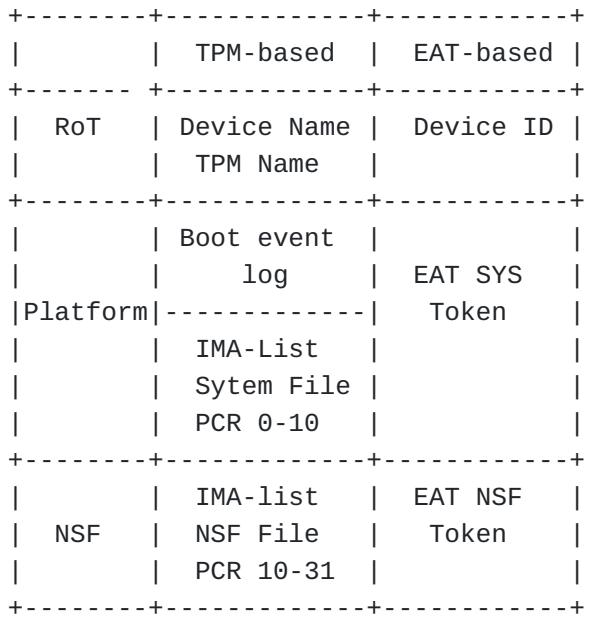

Figure 3: the mapping between different RoTs

5.1.1. YANG Tree Diagram of I2NSF Remote Attestation Evidence Interface

The YANG tree of i2nsf remote attestation evidence interface is shown below.

```
module: ietf-i2nsf-remote-attestation-evidence
  +--rw nsf-pcr-set {tpm:tpms}?
  |  +--rw nsf-name?    nsf-name
  |  +--rw pcr-index?   tpm:pcr
  +--rw eat-set {TEE}?
     +--rw algorithm?        enumeration
     +--rw cwt-uwt-choose?   int32

  rpcs:
    +---x nsf-challenge-response
    |  +---w input
    |  |  +---w nsf-name?   nsf-name
    |  |  +---w token?      binary
    |  |  +---w nonce?      uint32
    |  +--ro output
    |     +--ro (RoT-type)?
    |        +--:(TPM12) {TPM12}?
    |        |  +--ro tpm12-ra
    |        |     +--ro event-number?             uint64
    |        |     +--ro ima-template?             string
    |        |     +--ro filename-hint?            string
    |        |     +--ro filedata-hash?            binary
    |        |     +--ro filedata-hash-algorithm?  string
    |        |     +--ro template-hash-algorithm?  string
    |        |     +--ro template-hash?            binary
    |        |     +--ro pcr-index?                pcr
    |        |     +--ro signature?                binary
    |        |     +--ro up-time?                  uint32
    |        |     +--ro TPM_QUOTE2?               binary
    |        +--:(TPM20) {TPM20}?
    |        |  +--ro tpm20-ra
    |        |     +--ro event-number?             uint64
    |        |     +--ro ima-template?             string
    |        |     +--ro filename-hint?            string
    |        |     +--ro filedata-hash?            binary
    |        |     +--ro filedata-hash-algorithm?  string
    |        |     +--ro template-hash-algorithm?  string
    |        |     +--ro template-hash?            binary
    |        |     +--ro pcr-index?                pcr
    |        |     +--ro signature?                binary
    |        |     +--ro TPMS_QUOTE_INFO           binary
    |        |     +--ro quote-signature?          binary
    |        |     +--ro up-time?                  uint32
    |        |     +--ro unsigned-pcr-values* []
    |        |        +--ro tpm20-hash-algo?   identityref
    |        |        +--ro pcr-values* [pcr-index]
    |        |           +--ro pcr-index    pcr
    |        |           +--ro pcr-value?   binary
    |        +--:(TEE)
```

```
|           +--ro header-NSF?      binary
|           +--ro payload-NSF?     binary
|           +--ro signature-NSF?   binary
+---x platform-challenge-response
|  +---w input
|  |  +---w token?      binary
|  |  +---w nsf-name?   nsf-name
|  |  +---w nonce?      int32
|  +--ro output
|     +--ro (RoT-type)?
|        +--:(TPM12) {TPM12}?
|        |  +--ro tpm12-pra
|        |     +--ro event-number?              uint64
|        |     +--ro ima-template?              string
|        |     +--ro filename-hint?             string
|        |     +--ro filedata-hash?             binary
|        |     +--ro filedata-hash-algorithm?   string
|        |     +--ro template-hash-algorithm?   string
|        |     +--ro template-hash?             binary
|        |     +--ro pcr-index?                 pcr
|        |     +--ro signature?                 binary
|        |     +--ro bios-event-entry* [event-number]
|        |     |  +--ro event-number    uint32
|        |     |  +--ro event-type?     uint32
|        |     |  +--ro pcr-index?      pcr
|        |     |  +--ro digest-list* [hash-alog]
|        |     |  |  +--ro hash-algo?   identityref
|        |     |  |  +--ro digest*      binary
|        |     |  +--ro event-size?     uint32
|        |     |  +--ro event-data*     uint8
|        |     +--ro up-time?                   uint32
|        |     +--ro TPM_QUOTE2?                binary
|        +--:(TPM20) {TPM20}?
|        |  +--ro tpm2-pra
|        |     +--ro event-number?              uint64
|        |     +--ro ima-template?              string
|        |     +--ro filename-hint?             string
|        |     +--ro filedata-hash?             binary
|        |     +--ro filedata-hash-algorithm?   string
|        |     +--ro template-hash-algorithm?   string
|        |     +--ro template-hash?             binary
|        |     +--ro pcr-index?                 pcr
|        |     +--ro signature?                 binary
|        |     +--ro bios-event-entry* [event-number]
|        |     |  +--ro event-number    uint32
|        |     |  +--ro event-type?     uint32
|        |     |  +--ro pcr-index?      pcr
|        |     |  +--ro digest-list* [hash-alog]
|        |     |  |  +--ro hash-algo?   identityref
```

```
  |        |     |  |  +--ro digest*      binary
  |        |     |  +--ro event-size?     uint32
  |        |     |  +--ro event-data*     uint8
  |        |     +--ro TPMS_QUOTE_INFO           binary
  |        |     +--ro quote-signature?          binary
  |        |     +--ro up-time?                  uint32
  |        |     +--ro unsigned-pcr-values* []
  |        |        +--ro tpm20-hash-algo?   identityref
  |        |        +--ro pcr-values* [pcr-index]
  |        |           +--ro pcr-index    pcr
  |        |           +--ro pcr-value?   binary
  |        +--:(TEE) {TEE}?
  |           +--ro header-platform?     binary
  |           +--ro payload-platform?    binary
  |           +--ro signature-platform?  binary
  +---x RoT-challenge-response
     +---w input
     |  +---w token?      binary
     |  +---w nsf-name?   nsf-name
     |  +---w nonce?      int32
     +--ro output
        +--ro (RoT-type)?
           +--:(TPM12)
           |  +--ro rot-tpm12 {TPM12}?
           |     +--ro rot-name?           ->
             /tpm:rats-support-structures/tpms/tpm/
             certificates/certificate/name
           |     +--ro certificate-name    certificate-name-ref
           |     +--ro tpm12-hash-algo?    identityref
           +--:(TPM20) {TPM20}?
           |  +--ro rot-tpm20
           |     +--ro rot-name?           ->
             /tpm:rats-support-structures/tpms/tpm/
             certificates/certificate/name
           |     +--ro certificate-name    certificate-name-ref
           |     +--ro tpm20-hash-algo?    identityref
           +--:(TEE-general) {TEE}?
              +--ro TEE-UEID?    binary

notifications:
  +---n NSF-remote-attestation-event
  |  +--ro event-description?     string
  |  +--ro acquisition-method?    identityref
  |  +--ro emission-type?         identityref
  |  +--ro dampening-type?        identityref
  |  +--ro user                   string
  |  +--ro group*                 string
  |  +--ro ip-address             inet:ip-address
  |  +--ro authentication?        identityref
```

```
|  +--ro message?               string
|  +--ro vendor-name?           string
|  +--ro nsf-name?              union
|  +--ro severity?              severity
|  +--ro (RoT-type)?
|     +--:(TPM12) {TPM12}?
|     |  +--ro tpm12-ra
|     |     +--ro event-number?             uint64
|     |     +--ro ima-template?             string
|     |     +--ro filename-hint?            string
|     |     +--ro filedata-hash?            binary
|     |     +--ro filedata-hash-algorithm?  string
|     |     +--ro template-hash-algorithm?  string
|     |     +--ro template-hash?            binary
|     |     +--ro pcr-index?                pcr
|     |     +--ro signature?                binary
|     |     +--ro up-time?                  uint32
|     |     +--ro TPM_QUOTE2?               binary
|     +--:(TPM20) {TPM20}?
|     |  +--ro tpm20-ra
|     |     +--ro event-number?             uint64
|     |     +--ro ima-template?             string
|     |     +--ro filename-hint?            string
|     |     +--ro filedata-hash?            binary
|     |     +--ro filedata-hash-algorithm?  string
|     |     +--ro template-hash-algorithm?  string
|     |     +--ro template-hash?            binary
|     |     +--ro pcr-index?                pcr
|     |     +--ro signature?                binary
|     |     +--ro TPMS_QUOTE_INFO           binary
|     |     +--ro quote-signature?          binary
|     |     +--ro up-time?                  uint32
|     |     +--ro unsigned-pcr-values* []
|     |        +--ro tpm20-hash-algo?   identityref
|     |        +--ro pcr-values* [pcr-index]
|     |           +--ro pcr-index    pcr
|     |           +--ro pcr-value?   binary
|     +--:(TEE)
|        +--ro header-NSF?      binary
|        +--ro payload-NSF?     binary
|        +--ro signature-NSF?   binary
+---n Platform-remote-attestation-event
|  +--ro event-description?          string
|  +--ro acquisition-method?         identityref
|  +--ro emission-type?              identityref
|  +--ro dampening-type?             identityref
|  +--ro user                        string
|  +--ro group*                      string
|  +--ro ip-address                  inet:ip-address
```

```
|  +--ro authentication?            identityref
|  +--ro message?                   string
|  +--ro vendor-name?               string
|  +--ro nsf-name?                  union
|  +--ro severity?                  severity
|  +--ro (RoT-type)?
|     +--:(TPM12) {TPM12}?
|     |  +--ro tpm12-pra
|     |     +--ro event-number?             uint64
|     |     +--ro ima-template?             string
|     |     +--ro filename-hint?            string
|     |     +--ro filedata-hash?            binary
|     |     +--ro filedata-hash-algorithm?  string
|     |     +--ro template-hash-algorithm?  string
|     |     +--ro template-hash?            binary
|     |     +--ro pcr-index?                pcr
|     |     +--ro signature?                binary
|     |     +--ro bios-event-entry* [event-number]
|     |     |  +--ro event-number    uint32
|     |     |  +--ro event-type?     uint32
|     |     |  +--ro pcr-index?      pcr
|     |     |  +--ro digest-list* [hash-alog]
|     |     |  |  +--ro hash-algo?   identityref
|     |     |  |  +--ro digest*      binary
|     |     |  +--ro event-size?     uint32
|     |     |  +--ro event-data*     uint8
|     |     +--ro up-time?                  uint32
|     |     +--ro TPM_QUOTE2?               binary
|     +--:(TPM20) {TPM20}?
|     |  +--ro tpm2-pra
|     |     +--ro event-number?             uint64
|     |     +--ro ima-template?             string
|     |     +--ro filename-hint?            string
|     |     +--ro filedata-hash?            binary
|     |     +--ro filedata-hash-algorithm?  string
|     |     +--ro template-hash-algorithm?  string
|     |     +--ro template-hash?            binary
|     |     +--ro pcr-index?                pcr
|     |     +--ro signature?                binary
|     |     +--ro bios-event-entry* [event-number]
|     |     |  +--ro event-number    uint32
|     |     |  +--ro event-type?     uint32
|     |     |  +--ro pcr-index?      pcr
|     |     |  +--ro digest-list* [hash-alog]
|     |     |  |  +--ro hash-algo?   identityref
|     |     |  |  +--ro digest*      binary
|     |     |  +--ro event-size?     uint32
|     |     |  +--ro event-data*     uint8
|     |     +--ro TPMS_QUOTE_INFO           binary
```

```
|     |     +--ro quote-signature?           binary
|     |     +--ro up-time?                   uint32
|     |     +--ro unsigned-pcr-values* []
|     |        +--ro tpm20-hash-algo?   identityref
|     |        +--ro pcr-values* [pcr-index]
|     |           +--ro pcr-index    pcr
|     |           +--ro pcr-value?   binary
|     +--:(TEE) {TEE}?
|        +--ro header-platform?      binary
|        +--ro payload-platform?     binary
|        +--ro signature-platform?   binary
+---n RoT-remote-attestation-event
   +--ro event-description?    string
   +--ro acquisition-method?   identityref
   +--ro emission-type?        identityref
   +--ro dampening-type?       identityref
   +--ro user                  string
   +--ro group*                string
   +--ro ip-address            inet:ip-address
   +--ro authentication?       identityref
   +--ro message?              string
   +--ro vendor-name?          string
   +--ro nsf-name?             union
   +--ro severity?             severity
   +--ro (RoT-type)?
      +--:(TPM12)
      |  +--ro rot-tpm12 {TPM12}?
      |     +--ro rot-name?           ->
        /tpm:rats-support-structures/tpms/tpm/
        certificates/certificate/name
      |     +--ro certificate-name    certificate-name-ref
      |     +--ro tpm12-hash-algo?    identityref
      +--:(TPM20) {TPM20}?
      |  +--ro rot-tpm20
      |     +--ro rot-name?           ->
        /tpm:rats-support-structures/tpms/tpm/
        certificates/certificate/name
      |     +--ro certificate-name    certificate-name-ref
      |     +--ro tpm20-hash-algo?    identityref
      +--:(TEE-general) {TEE}?
         +--ro TEE-UEID?       binary
```

5.1.2. YANG Data Model of I2NSF Remote Attestation Interface

The YANG Model of I2NSF Remote Attestation Evidence Interface is shown below.

```
module ietf-i2nsf-remote-attestation-evidence{
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-
                attestation-evidence";
  prefix
    nsfra;
  import ietf-i2nsf-nsf-monitoring{
    prefix nsfmi;
  reference
    "Section 9 of draft-ietf-i2nsf-nsf-facing-interface";
  }
  import ietf-tpm-remote-attestation{
      prefix tpm;
  }
  import ietf-inet-types{
    prefix inet;
    reference
      "section 4 of RFC 6991";
  }
  organization
    "IETF I2NSF (Interface to Network Security Functions)
     Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/i2nsf>
     WG List: <mailto:i2nsf@ietf.org>

     Editor: Penglin Yang
       <mailto:yangpenglin@chinamobile.com>";

  description
    "This module is a YANG module for I2NSF Remote
     Attestation Interface.";
  feature TPM12{
    description
      "This device is for TPM12 remote attestation";
  }
  feature TPM20{
    description
      "This device is for TPM20 remote attestation";
  }
  feature TEE{
    description
      "This device is for general TEE remote attestation";
  }
  typedef nsf-name{
    type union{
      type string;
      type inet:ip-address-no-zone;
```

```
  }
  description
    "nsf-name for remote attestation";
}
identity RoT-type{
  description
    "RoT have different types, like TPM, TEE, etc.";
}
identity TPM12{
  base RoT-type;
  description
    "RoT type is TPM1.2";
}
identity TPM20{
  base RoT-type;
  description
    "RoT type is TPM2.0";
}
identity TEE{
  base RoT-type;
  description
    "RoT type is TEE";
}
identity cwt{
  description
    "cbor web token for remote attestation";
}
identity jwt{
  description
    "json web token for remote attestation";
}
identity nsf-name{
  description
    "nsf name";
}

grouping nsf-remote-attestation{
  description
    "This grouping is for certain nsf's remote attestation result.";
  choice RoT-type{
    case TPM12{
      if-feature "TPM12";
      description
        "The filename hint of IMA log item is NSF name. The range
         of PCR index is defined as 11~32.";
      container tpm12-ra{
        uses tpm:ima-event;
        uses tpm:tpm12-attestation;
      }
```

```
    }
    case TPM20{
      if-feature "TPM20";
      container tpm20-ra{
        uses tpm:ima-event;
        uses tpm:tpm20-attestation;
      }
    }
    case TEE{
      description
        "EAT for NSF remote attestation";
      leaf header-NSF{
        type binary;
      }
      leaf payload-NSF{
        type binary;
      }
      leaf signature-NSF{
        type binary;
      }
    }
  }
}
grouping platform-remote-attestation{
  description
    "this item is for platform remote attestation";
  choice RoT-type{
    case TPM12{
      if-feature "TPM12";
      container tpm12-pra{
        uses tpm:ima-event;
        uses tpm:bios-event-log;
        uses tpm:tpm12-attestation;
      }
    }
    case TPM20{
      if-feature "TPM20";
      container tpm20-pra{
        uses tpm:ima-event;
        uses tpm:bios-event-log;
        uses tpm:tpm20-attestation;
      }
    }
    case TEE{
      if-feature "TEE";
      description
        "EAT for Platform Remote Attestation";
      leaf header-platform{
        type binary;
```

```
      }
      leaf payload-platform{
        type binary;
      }
      leaf signature-platform{
        type binary;
      }
    }
  }
}
grouping RoT-remote-attestation{
  description
    "this item is for the identity of platform and RoT";
  choice RoT-type{
    case TPM12{
      container rot-tpm12{
        if-feature "TPM12";
        description
          "the identity of TPM could be represented by
           TPM-name and certificate";
        leaf rot-name{
          type leafref {
            path "/tpm:rats-support-structures/tpm:tpms/tpm:tpm"
               + "/tpm:certificates/tpm:certificate/tpm:name";
            }
        }
        uses tpm:certificate-name-ref;
        uses tpm:tpm12-hash-algo;
      }
    }
    case TPM20{
      if-feature "TPM20";
      description
        "the identity of TPM could be represented
         by TPM-name and certificate";
      container rot-tpm20{
        leaf rot-name{
          type leafref {
          path "/tpm:rats-support-structures/tpm:tpms/tpm:tpm"
             + "/tpm:certificates/tpm:certificate/tpm:name";
          }
        }
        uses tpm:certificate-name-ref;
        uses tpm:tpm20-hash-algo;
      }
    }
    case TEE-general{
      if-feature "TEE";
      leaf TEE-UEID{
```

```
          type binary;
        }
      }
          }
  }
  notification NSF-remote-attestation-event{
    description
      "Event that triggered the NSF remote attestation
       will use this notification";
    leaf event-description{
      description
        "describe the reason why notification was triggered";
      type string;
    }
    uses nsfmi:characteristics;
    uses nsfmi:i2nsf-system-event-type-content;
    uses nsfmi:common-monitoring-data;
    uses nsf-remote-attestation;
  }
  notification Platform-remote-attestation-event{
    description
      "Event that triggered the platform remote attestation
       will use this notification ";
    leaf event-description{
      description
        "describe why this notification was triggered";
      type string;
    }
    uses nsfmi:characteristics;
    uses nsfmi:i2nsf-system-event-type-content;
    uses nsfmi:common-monitoring-data;
    uses platform-remote-attestation;
  }
  notification RoT-remote-attestation-event{
    description
      "Event that triggered the rot remote attestation
       will use this notification";
    leaf event-description{
      description
        "describe why this notification was triggered";
      type string;
    }
    uses nsfmi:characteristics;
    uses nsfmi:i2nsf-system-event-type-content;
    uses nsfmi:common-monitoring-data;
    uses RoT-remote-attestation;
  }

//token in RPC is for specify the identity of RPC caller. //
```

```
grouping token{
  description
    "this token is for identify rpc caller. How to define
     this token, oauth, JWT, or other? Or not necessary, TBD";
  leaf token{
    type binary;
  }
}
rpc nsf-challenge-response{
  description
    "this is the unified rpc for nsf remote attestation";
  input{
    leaf nsf-name{
      type nsf-name;
    }
    uses token;
    leaf nonce{
      type uint32;
    }
  }
  output{
    uses nsf-remote-attestation;
  }
}
rpc platform-challenge-response{
  description
    "this rpc is for platform challenge ";
  input{
    uses token;
    leaf nsf-name{
      type nsf-name;
    }
    leaf nonce{
      type int32;
    }
  }
  output{
    uses platform-remote-attestation;
  }
}
rpc RoT-challenge-response{
  input{
    uses token;
    leaf nsf-name{
      type nsf-name;
    }
    leaf nonce{
      type int32;
    }
```

```
    }
    output{
      uses RoT-remote-attestation;
    }
  }

//***********************************************//
//    configuration about PCR and NSF set.       //
//***********************************************//
  container nsf-pcr-set{
    description
      "this container is used for NSF-name, IMA log and
       PCR index setting. NSF-name needs to be set as the
       IMA item filename-hint, the pcr value need to be
       set as the IMA pcr index.";
    if-feature "tpm:tpms";
    leaf nsf-name{
      type nsf-name;
    }
    leaf pcr-index{
      type tpm:pcr;
    }
  }

//***********************************************//
//         configuration about EAT set.          //
//***********************************************//
  container eat-set{
    description
      "this container is for NSF-name set in EAT environment";
    if-feature "TEE";
    leaf algorithm{
      description
        "set the signing algorithm for generating EAT";
      type enumeration{
        enum HS256;//hmac with sha256
        enum RS256;//rsa with sha256
      }
    }
    leaf cwt-uwt-choose{
      type int32;
      description
        "0 is cwt, 1 is uccs, 2 is jwt";
    }
  }
}
```

5.2. I2NSF Remote Attestation Reference Value Interface

The reference value of a NSF needs to be conveyed by I2NSF remote attestation reference value interface. The interface works between Network Operator Mgmt System and Developer's Management System.

5.2.1. YANG Tree Diagram of I2NSF Remote Attestation Reference Value Interface

```
module: ietf-i2nsf-remote-attestation-reference-value
  +--rw nsf-tpm-reference-value-registration
  |  +--rw nsf-name              nsf-name
  |  +--rw ima-template?         string
  |  +--rw nsf-hash?             binary
  |  +--rw nsf-hash-algorithm?   string
  |  +--rw pcr-index?            pcr
  +--rw nsf-tee-reference-value-registration
  |  +--rw nsf-name         nsf-name
  |  +--rw header-NSF?      binary
  |  +--rw payload-NSF?     binary
  |  +--rw signature-NSF?   binary
  +--rw rot-reference-value-registration
  |  +--rw (RoT-type)?
  |     +--:(TPM12) {TPM12}?
  |     |  +--rw rot-tm12
  |     |     +--rw rot-tpm12-name?     string
  |     |     +--rw certificate-name    certificate-name-ref
  |     |     +--rw tpm12-hash-algo?    identityref
  |     +--:(TPM20) {TPM20}?
  |     |  +--rw rot-tpm20
  |     |     +--rw rot-tpm20-name?     string
  |     |     +--rw certificate-name    certificate-name-ref
  |     |     +--rw tpm20-hash-algo?    identityref
  |     +--:(TEE) {TEE}?
  |        +--rw TEE-UEID?    binary
  +--rw platform-tpm-reference-value-registration
  |  +--rw platform-name         string
  |  +--rw ima-template?         string
  |  +--rw nsf-hash?             binary
  |  +--rw nsf-hash-algorithm?   string
  |  +--rw pcr-index?            pcr
  +--rw platform-tee-remote-attestation-reference-value
     +--rw platform-name    nsf-name
     +--rw header-NSF?      binary
     +--rw payload-NSF?     binary
     +--rw signature-NSF?   binary
```

5.2.2. YANG Data Model of I2NSF Remote Attestation Reference Value Interface

The YANG Model of I2NSF remote attestation reference value interface is shown below. The registration information will refer to the measurement logs and algorithms of remote attestation. The log infomation contains all the information needed by Network Operator Mgmt System to apparaise attester's evidence.

```
module ietf-i2nsf-remote-attestation-reference-value {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-
     attestation-reference-value";
  prefix
    nsfrarv;

  import ietf-tpm-remote-attestation{
    prefix tpm;
  }
  import ietf-inet-types{
    prefix inet;
  }
  organization
    "IETF I2NSF (Interface to Network Security Functions)
     Working Group";
  contact
    "WG Web: <http://tools.ietf.org/wg/i2nsf>
    WG List: <mailto:i2nsf@ietf.org>
    Editor: Penglin Yang
    <mailto:yangpenglin@chinamobile.com>";

  description
    "This module is a YANG module for I2NSF NSF remote
     attestation reference value.";

  identity RoT-type{
    description
      "RoT have different types, like TPM, TEE, etc.";
  }

  identity TPM12{
    base RoT-type;
    description
      "RoT type is TPM1.2";
  }
  identity TPM20{
    base RoT-type;
    description
      "RoT type is TPM2.0";
  }
  identity TEE{
    base RoT-type;
    description
      "RoT type is TEE general";
  }
  feature TPM12{
    description
```

```
    "tpm 1.2 version";
}
feature TPM20{
  description
    "tpm 2.0 version";
}
feature TEE{
  description
    "TEE version";
}
typedef nsf-name{
  type union{
    type string;
    type inet:ip-address-no-zone;
  }
  description
    "nsf-name for regular expression";
}
typedef pcr{
  type uint8{
    range "0..31";
  }
}

container nsf-tpm-reference-value-registration{
  description
    "the reference value is for nsf in tpm20 platform";
  leaf nsf-name{
    type nsf-name;
    mandatory true;
    description
      "The name of nsf";
  }
  leaf ima-template {
    type string;
    description
      "Name of the template used for event logs
       for e.g. ima, ima-ng, ima-sig";
  }
  leaf nsf-hash {
    type binary;
    description
      "Hash of nsf file/image";
  }
  leaf nsf-hash-algorithm {
    type string;
    description
      "Algorithm used for filedata-hash";
  }
```

```
  leaf pcr-index {
    type pcr;
    description
      "Defines the PCR index that stores this nsf";
  }
}
container nsf-tee-reference-value-registration{
  description
    "the reference value is for nsf in TEE platform";
  leaf nsf-name{
    type nsf-name;
    mandatory true;
    description
      "The name of nsf";
  }
  leaf header-NSF{
    type binary;
  }
  leaf payload-NSF{
    type binary;
  }
  leaf signature-NSF{
    type binary;
  }
}
container rot-reference-value-registration{
  description
    "this container is for root of trust reference value";
  choice RoT-type{
    case TPM12{
      if-feature "TPM12";
      description
        "the identity of TPM could be represented by
         TPM-name and certificate";
      container rot-tm12{
        leaf rot-tpm12-name{
          type string;
        }
        uses tpm:certificate-name-ref;
        uses tpm:tpm12-hash-algo;
      }
    }
    case TPM20{
      if-feature "TPM20";
      description
        "the identity of TPM could be represented by
         TPM-name and certificate";
      container rot-tpm20{
        leaf rot-tpm20-name{
```

```
          type string;
        }
        uses tpm:certificate-name-ref;
        uses tpm:tpm20-hash-algo;
      }
    }
    case TEE{
      if-feature "TEE";
      leaf TEE-UEID{
        //provide a UEID to identify TEE
      type binary;
      }
    }
  }
}
container platform-tpm-reference-value-registration{
  description
    "this container is for platform reference value";
  leaf platform-name{
    type string;
    mandatory true;
    description
      "The name of nsf";
  }
  leaf ima-template {
    type string;
    description
      "Name of the template used for event logs
       for e.g. ima, ima-ng, ima-sig";
  }
  leaf nsf-hash {
    type binary;
    description
      "Hash of nsf file/image";
  }
  leaf nsf-hash-algorithm {
    type string;
    description
      "Algorithm used for filedata-hash";
  }
  leaf pcr-index {
    type pcr;
    description
      "Defines the PCR index that stores this nsf";
  }
}
container platform-tee-remote-attestation-reference-value{
  description
    "the reference value is for platform in TEE platform";
```

```
    leaf platform-name{
      type nsf-name;
      mandatory true;
      description
        "The name of nsf";
    }
    leaf header-NSF{
      type binary;
    }
    leaf payload-NSF{
      type binary;
    }
    leaf signature-NSF{
      type binary;
    }
  }
}
```

5.3. I2NSF Remote Attestation Result Interface

This interface is used to transfer attestation result from Verifier to Network User. The definition and data format refer to [I-D.ietf-rats-ar4si], TBD.

6. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" RFC 3688 [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-attestation-evidence Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-attestation-reference-value Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document requests IANA to register the following YANG module in the "YANG Module Names" registry RFC 7950 [RFC7950] RFC 8525 [RFC8525]:

Name: ietf-i2nsf-remote-attestation-evidence-interface

Namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-attestation-evidence

Prefix: nsfra

Reference: RFC XXXX

Name: ietf-i2nsf-remote-attestation-reference-value-interface

Namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-remote-attestation-reference-value

Prefix: nsfteri

Reference: RFC XXXX

7. Security Considerations

This document introduces the architecture of I2NSF remote attestation and designs related interfaces. Different RoT architectures have different trust ability and different appearance. Network Operator Mgmt System will determine if it will trust these remote attestation results by customized policy rules. The I2NSF

remote attestation interfaces need to be protected by secure channel when transmission occurs. Meanwhile, the remote attestation results in interfaces are protected by their own mechanisms like TPM signature or token.

8. References

8.1. Normative Reference

[I-D.ietf-rats-ar4si] Voit, E., Birkholz, H., Hardjono, T., Fossati, T., and V. Scarlata, "Attestation Results for Secure Interactions", Work in Progress, Internet-Draft, draft-ietf-rats-ar4si-02, 7 March 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-ar4si-02.txt>.

[I-D.ietf-rats-architecture] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote Attestation Procedures Architecture", Work in Progress, Internet-Draft, draft-ietf-rats-architecture-17, 1 June 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-architecture-17.txt>.

[I-D.ietf-rats-eat] Lundblade, L., Mandyam, G., and J. O'Donoghue, "The Entity Attestation Token (EAT)", Work in Progress, Internet-Draft, draft-ietf-rats-eat-13, 20 May 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-eat-13.txt>.

[I-D.ietf-rats-tpm-based-network-device-attest] Fedorkow, G., Voit, E., and J. Fitzgerald-McKay, "TPM-based Network Device Remote Integrity Verification", Work in Progress, Internet-Draft, draft-ietf-rats-tpm-based-network-device-attest-14, 22 March 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-tpm-based-network-device-attest-14.txt>.

[I-D.ietf-rats-yang-tpm-charra]
Birkholz, H., Eckel, M., Bhandari, S., Voit, E., Sulzen, B., (Frank), L. X., Laffey, T., and G. C. Fedorkow, "A YANG Data Model for Challenge-Response-based Remote Attestation Procedures using TPMs", Work in Progress, Internet-Draft, draft-ietf-rats-yang-tpm-charra-21, 18 May 2022, <https://www.ietf.org/archive/id/draft-ietf-rats-yang-tpm-charra-21.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <https://www.rfc-editor.org/info/rfc8329>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

8.2. Informative Reference

[NFV-SEC-018] ETSI, "Report on NFV Remote Attestation Architecture", November 2019, <https://www.etsi.org/deliver/etsi_gr/NFV-SEC/001_099/018/01.01.01_60/gr_NFV-SEC018v010101p.pdf>.

[SGX] Intel, "Overview of Intel Software Guard Extension", June 2016, <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html>.

[TCGRoT] Trust Computing Group, "DRAFT: TCG Roots of Trust Specification", October 2018, <https://trustedcomputinggroup.org/wp-content/uploads/TCG_Roots_of_Trust_Specification_v0p20_PUBLIC_REVIEW.pdf>.

[TEE] Global Platform Technology, "Global Platform Technology TEE System Architecture Version 1.2", December 2018, <https://globalplatform.org/specs-library/tee-system-architecture-v1-2/>.

[tpm12] Trusted Computing Group, "TPM Main Specification Level 2 Version 1.2, Revision 116", March 2011, <https://

trustedcomputinggroup.org/resource/tpm-main-specification/>.

[tpm20] Trusted Computing Group, "Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.59", November 2019, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

Authors' Addresses

Penglin Yang
China Mobile
32 Xuanwumen West Street, Xicheng District
Beijing
100053
China

Email: yangpenglin@chinamobile.com

Meiling Chen
China Mobile
32 Xuanwumen West Street, Xicheng District
Beijing
100053
China

Email: chenmeiling@chinamobile.com

Li Su
China Mobile
32 Xuanwumen West Street, Xicheng District
Beijing
100053
China

Email: suli@chinamobile.com

Diego Lopiz
Telefonica

Email: diego.r.lopez@telefonica.com

Jaehoon Paul Jeong
Sungkyunkwan University

Email: jaehoon.paul@gmail.com

Linda Dunbar
Futurewei

Email: linda.dunbar@futurewei.com