|  |  |
| --- | --- |
| Input Contribution | |
| Meeting ID\* | SEC #27 |
| Title:\* | Solution of Decentralized Authentication for Peer-to-Peer Communication |
| Source:\* | Jie Shi, Huawei, [shi.jie1@huawei.com](mailto:shi.jie1@huawei.com)  Guilin Wang, Huawei, [wang.guilin@huawei.com](mailto:wang.guilin@huawei.com)  Yanjiang Yang, Huawei, [yang.yanjiang@huawei.com](mailto:yang.yanjiang@huawei.com) |
| Uploaded Date:\* | 2017/02/06 |
| Document(s)  Impacted\* | TR-xxxx Decentralized Authentication |
| Intended purpose of  document:\* | Decision  Discussion  Information  Other <specify> |
| Decision requested or recommendation:\* | It is proposed to agree the content in this contribution. |
| Template Version:23 February 2015 (Dot not modify) | |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

### -----------------------start of change 1---------------------------------------------

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Dan Boneh, Matthew K. Franklin: “Identity-Based Encryption from the Weil Pairing”, Proceedings of CRYPTO 2001.

[i.2] RFC 6507: “Elliptic Cureve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)”, February, 2012.

### -----------------------End of change 1---------------------------------------------

### -----------------------start of change 2---------------------------------------------

# X Avaliable Options

## X.1 Decentralized Authentication for Peer-to-Peer Communication

### X.1.1 The sequence of events in decentralized authentication

This clause describes an decentralized authentication framework for peer-to-peer communication. The framework enables mutual authentication of two entities (AE or CSE). This framework employs an identity based cryptography (IBC) credential that has been provisioned into the entities. The provisioning of an IBC credential could be a pre-provisioing or a remote provisioning. The entities authenticate each other by using (D)TLS handshake with PKS ciphersuite.

Figure X.1-1 shows the sequence of events in this decentralized authentication framework using IBC credential. In the description, “Entity A” and “Entity B” could be a CSE or AE.



Figure X.1-1: The sequence of events in decentralized authentication using IBC credential

**Credential Configuration:** IBC based credentials (IdA, SkA, MPK) and (IdB, SkB, MPK) are provisioned in entities A and B respectively, where IdA and IdB are the identities of entities A and B, SkA and SkB are the corresponding private keys for IdA and IdB, and the MPK is the master public key. Particular instantiations are introduced in the Section X.1.2.

**Association Security Handshake**: The entities shall perform an IBC based (D)TLS-PSK handshake to establish a secure session. The details are described as follows, where TLS messages without special description remain.

1. Entity A initiates TLS with B using a TLS\_PSK ciphersuite, i.e., Entity A sends a TLS ClientHello message with PSK ciphersuite to entity B.
2. Entity B returns TLS message ServerKeyExchange, in which the psk\_identity\_hint in ServerKeyExchange is set to IdB.
3. After receiving ServerKeyExchange message, entity A retrieves IdB, computes a key K = KeyGen(IdB, SkA, MPK) and sets TLS psk parameter to be K, which will be used to authenticate entity B. Here, the function KeyGen can be implemented using different techniques, and two particular implementations are described in the following Section X.1.2.
4. Entity A sends TLS message ClientKeyExchange and Finished, where the psk\_identity in ClientKeyExchange is set to IdA;
5. After receiving ClientKeyExchange, entity B retrieves IdA, computes a Key K = KeyGen(IdA, SkA, MPK), and sets TLS psk parameter to be K, which will be used to authenticate entity A. Here, the function KeyGen can be implemented using different techniques, and two particular implementations are described in the following Section X.1.2.
6. Entity B sends TLS message Finished to A and completes the TLS-PSK handshake.

### X.1.2 Instantiation of IBC based credentials

Two instantiations of IBC based credentials are introduced below.

**Instantiation 1**: This instantiation of IBC based credential is adopted from IBC implementation using pairing [i.1].

Let G1 and G2 be respectively an additive and multiplicative groups of the same order q, which is a large prime. A bilinear map ê: G1 × G1 → G2 is defined as follows.

* Bilinear: ê(aP, bQ) = ê(P, Q)ab for all P, Q in G1 and all a, b in Z (the set of all integers);
* Non-degenerate: ê(P, P) ≠ 1 for a generator P of G1;
* Computability: there exists an efficient algorithm to compute ê;

The bilinear map can be implemented using Weil or Tate pairings.

The master public key is MPK = <q, G1, G2, ê, n, P, Ppub, H1, H2> and master key sk = s in Zq\*, where H1:{0,1}\*→G1\* and H2:G2→{0,1}n are two hash functions, n is the binary length of output of hash function H2, P is a random generator of G1 and Ppub = sP, and s is a random value selected from Zq\* (the set of all non-zero residuals of modular q). Given an identity ID, the corresponding secret key is computed as SkID = sH1(ID).

Thus, for entity A and entity B with identities IdA and IdB respectively, their IBC based credentials are (IdA, SkA, MPK) and (IdB, SkB, MPK), respectively, where MPK = <q, G1, G2, ê, n, P, Ppub, H1, H2>, SkA = sH1(IdA), and SkB = sH(IdB)

The key generation function KeyGen is defined as follows.

K = KeyGen(IdB, SkA, MPK) = e(H1(IdB), SkA) = e(H1(IdB), sH1(IdA)) = e(H1(IdA), H1(IdB))s;

K = KeyGen(IdA, SkB, MPK) = e(H1(IdA), SkB) = e(H1(IdA), sH1(IdB)) = e(H1(IdA), H1(IdB))s;

**Instantiation 2**: This instantiation of IBC based credential is adopted from IBC derived from Schnorr Signature [i.2].

Let Fp denote the finite field with p elements, where p is a large prime. The scheme works on an elliptic curve defined over Fp, having a subgroup of prime order q. Let G be a point on the elliptic curve that generates the subgroup of order q. The master private key s is a random secret selected from Zq\*, and P is computed by P = sG.

The master public key is MPK = <Fp, q, P, H>, where H is a hash function. Given an identity ID, the corresponding secret key is computed as SkID = (sk, V), in which V = vG for a random v selected from Zq\*, and sk = s+vH(G||P||ID||V) mod q;

Thus, for entity A and entity B with identities are IdA and IdB respectively, their IBC based credentials are (IdA, SkA, MPK) and (IdB, SkB, MPK). Here, MPK = < Fp, q, P, H >, SkA = kA, SkB = kB, IdA =(A, VA), and IdB =(B, VB), in which VA = vAG, VB = vBG, kA = s+vAH(G||P||IdA||VA) mod q, kB = s+vBH(G||P||IdB||VB) mod q, vA and vB are two random numbers selected from Zq\*

The key generation function KeyGen is defined as follows.

K = KeyGen(IdB, SkA, MPK) = kA(P + HBVB) = (s + vAHA)(sG+HB(vBG)) = (s + vAHA)(s+vBHB)G

K = KeyGen(IdA, SkB, MPK) = kB(P + HAVA) = (s + vBHB)(sG+HA(vAG)) = (s + vAHA)(s+vBHB)G

where HA = H(G||P||IdA||VA) and HB = H(G||P||IdB||VB);

### -----------------------End of change 2---------------------------------------------