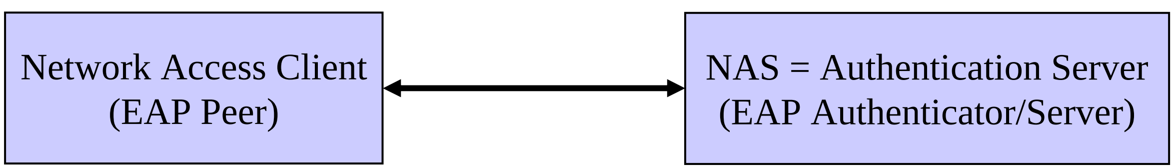
Problem Statement and Requirements on 3-Party Key Distribution Protocol for Handover Keying (draft-ohba-hokey-3party-keydist-01.txt)

Dan Harkins
Yoshihiro Ohba
Madjid Nakhjiri
Rafa Marin Lopez

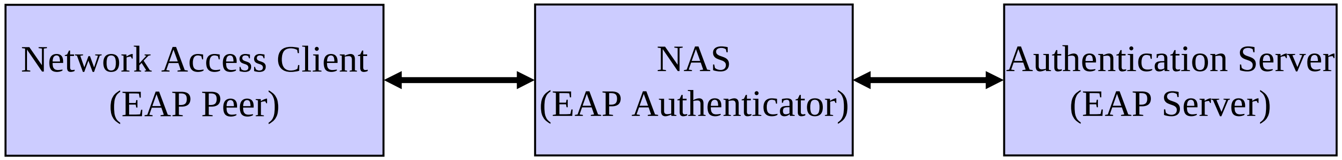
Two-Party Authentication Model

- Network access authentication and authorization services have been based on two-party trust model
  - Credentials are maintained by the authentication server
  - Initially, NAS was taking the role of authentication server, so it was really a two-party model



3-Party Key Distribution Problem

- Eventually auth/authz services were then extended to be more scalable
  - This created functional separation between EAP authenticator and server
- In many cases, an SA between peer and authenticator needs to be dynamically established
- A session key needs to be transferred from EAP server to authenticator
- This key distribution created a "**Channel Binding**" (aka **lying NAS**) problem
- Lack of Channel Binding can result in the session key to be bound to wrong context:
  - The authenticator advertises a forged identity to one of the peer and server
  - The authenticator advertises the same forged identities to the peer and server



3-Party Key Distribution Problem (cont'd)

- There are several Channel Binding mechanisms
- However, correct operation of this binding has depended on deployment AAA protocols
  - The identities of AAA protocol endpoints need to be same as or associated with the identities visible to the 3 parties
  - Mis-deployment of AAA protocol can break security properties
- What is needed: **3-party key distribution protocol** whose security properties do not solely depend on a particular deployment of AAA protocol

Requirements

1. **Confidentiality** -- disclosure of the keying material to passive and active attackers of the key distribution protocol MUST NOT be possible.

2. **Integrity protection** -- it MUST be possible to detect tampering of a network access credential.

3. **Validation of credential source** -- the recipient of a network access credential MUST be able to prove who it came from and for what context the credential was delivered.

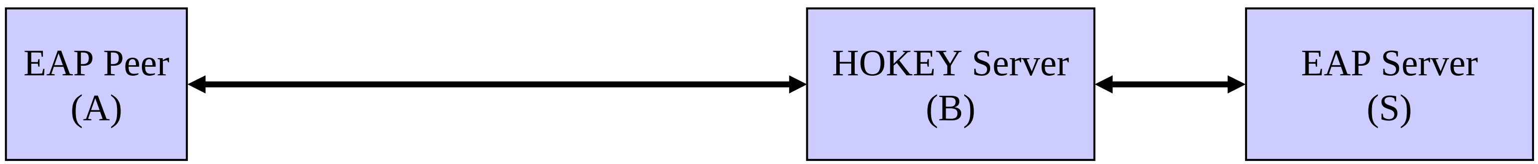
4. **Validation of authorization** -- the scope (intended users) of the network access credential MUST be distributed as part of the credential and MUST be protected to the same degree as the credential itself. The context (life time, labels, intended usage, etc) of the network access credentials MUST be distributed as part of the credentials and MUST be protected to the same degree.

5. **Resilience** -- It MUST NOT be possible for an active attacker to consent of the client.

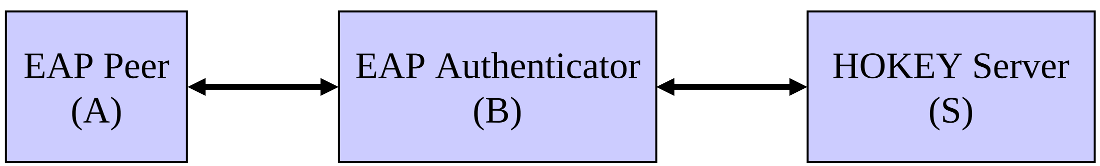
Requirements (cont’d)

6. **Peer consent** -- Either the credential MUST NOT be distributed without the consent of the client or it MUST be unusable without the consent of the client.
7. **Verification of identity** -- Identities of the three parties involved MUST be confirmed by all three parties.
8. **Agreement by all parties** -- If the protocol successfully completes all three parties MUST agree on the keying material disclosed and the identity of the entity to whom the keying material was disclosed.
9. **Replay protection** -- replay attacks MUST NOT effect the key distribution protocol.

Use Cases



Use Case 1



Use Case 2

- A(Alice), B(Bob), S(Server): Key distribution principals
- In both Use Cases, there is a pre-established SA between B and S

Notation

- {X}K: authenticated encryption of X with key K
- <X>K: encryption of X with K
- [X]K: Message Authentication Code of X with key K.
- H(x): Digest produced from a one-way hash function given x as input
- x|y: Concatenation of x and y

- Kas: A symmetric key shared between A and S
- Kbs: A symmetric key shared between B and S
- Kab: A symmetric key to be shared between A and B
- L : Key lifetime for Kab

- Tx : Timestamp generated by the party X
- Nx : Nonce provided by the party X

Candidate 3-Party Key Distribution Protocols

1. Kerberos
2. ISO/IEC 11770-2 mechanism 10
3. Improved 3PKD
4. Modified Otway-Rees

- Other protocols may be possible
- Validation against the requirements is needed

Call Flow of Candidate Solutions

Kerberos

A→S: A,B
S→A: {Ts, L, Kab, B, {Ts, L, Kab, A}Kbs}Kas
A→B: {Ts, L, Kab, A}Kbs, {A, Ta}Kab
B→A: {Ta+1}Kab

ISO/IEC 11770-2 mechanism 10

A→S: {Ta,B}Kas
S→A: {Ts,Kab,B}Kas
S→B: {Ts',Kab,A}Kbs

Improved 3PKD

A→S: Na
B→S: Na,Nb
S→A: <Kab>Kas,[A,B,Na,Nb,Ns,<Kab>Kas]Kas,Nb,Ns
S→B: <Kab>Kbs,[A,B,Na,Nb,Ns,<Kab>Kbs]Kbs

Modified Otway-Rees

A→B: A, {A, B, Na}Kas
B→S: A, B, {Nb}Kbs, {A, B, Na}Kas
S→B: {Na, Nt, B}Kas, {A, Na, Nb, Nt, Kab}Kbs
B→A: {Na, Nt, B}Kas, H(Na|Nt)

Comparison of the Candidate Protocols

- Some protocols allows A to directly retrieve key from S
  - Kerberos, ISO/IEC 11770-2 mechanism 10
  - This may be a good for proactive operation because A does not need to talk to B until it hands over to B
- Some protocol makes use of key hierarchy (hence no key distribution to A)
  - Modified Otway-Rees
  - This may be a good fit for HOKEY purpose
- Some protocol provides key confirmation betw. A and B
  - Kerberos
  - Key confirmation between A and B is also provided by lower layer
- Some protocols use timestamp (Kerberos, ISO) while others nonces
- Some protocol supports cross-realm operation (Kerberos)

Key Distribution Protocol Transport

- Defining a new EAP Code or Type
  - EAP does not really fit 3-party model because EAP is designed to be Mode Independent
- Lower-layer specific transport is most plausible approach
  - Encapsulating 3-party key distribution protocol in link-layer frames and AAA attributes
  - Lower-layer may be IP in some cases
    - Between A and B in Use Case 1 or in proactive Use Case 2

Summary

- HOKEY WG should work on a 3-party key distribution protocol
- It is recommend to do more investigation