SNMP over (D)TLS
IETF-75

Wes Hardaker
ietf@hardakers.net

Overview

- Recap of Current Draft Status (-04)
- SSH Identity / securityName refresher
- (D)TLS X.509 / securityName overview
- subjectAltName details
- Other (D)TLS Considerations

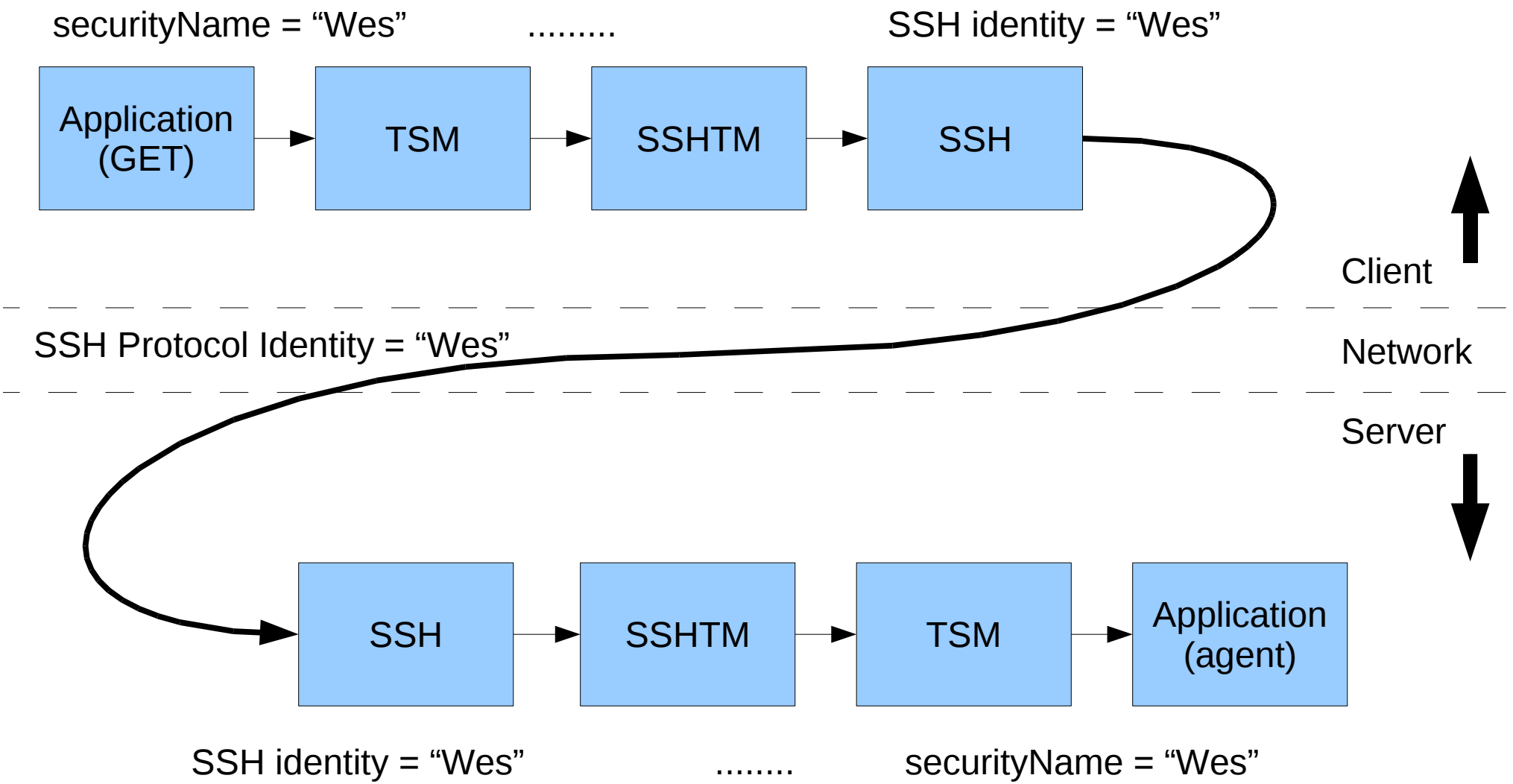
Current Draft Status

- draft-hardaker-isms-dtls-tm-04
- Updates since -03:
  - Added support for TLS
    - Brings list to TLS, DTLS/UDP and DTLS/SCTP
    - Uses (D)TLS to speak generically about any of them
    - Uses TLS or DTLS over XXX to speak about individuals
  - Other minor wording changes
- Mostly Done!
  - The biggest area for critique is the MIB tables (IMHO)
  - (and is most of the open issues to discuss today)

Review: SSH Identity / securityNames

- SSH has an implicit “identity” that is sent through the protocol.
  - Maps dircetly to a securityName
  - Traditionally short (“login names”)
- Simple and Easy, mostly
  - TSM optionally adds a “xxxx:” prefix
  - We provide “otheruser@” prefix support to securityNames for non-1:1 mappings

Review:
SSH Identity / securityName



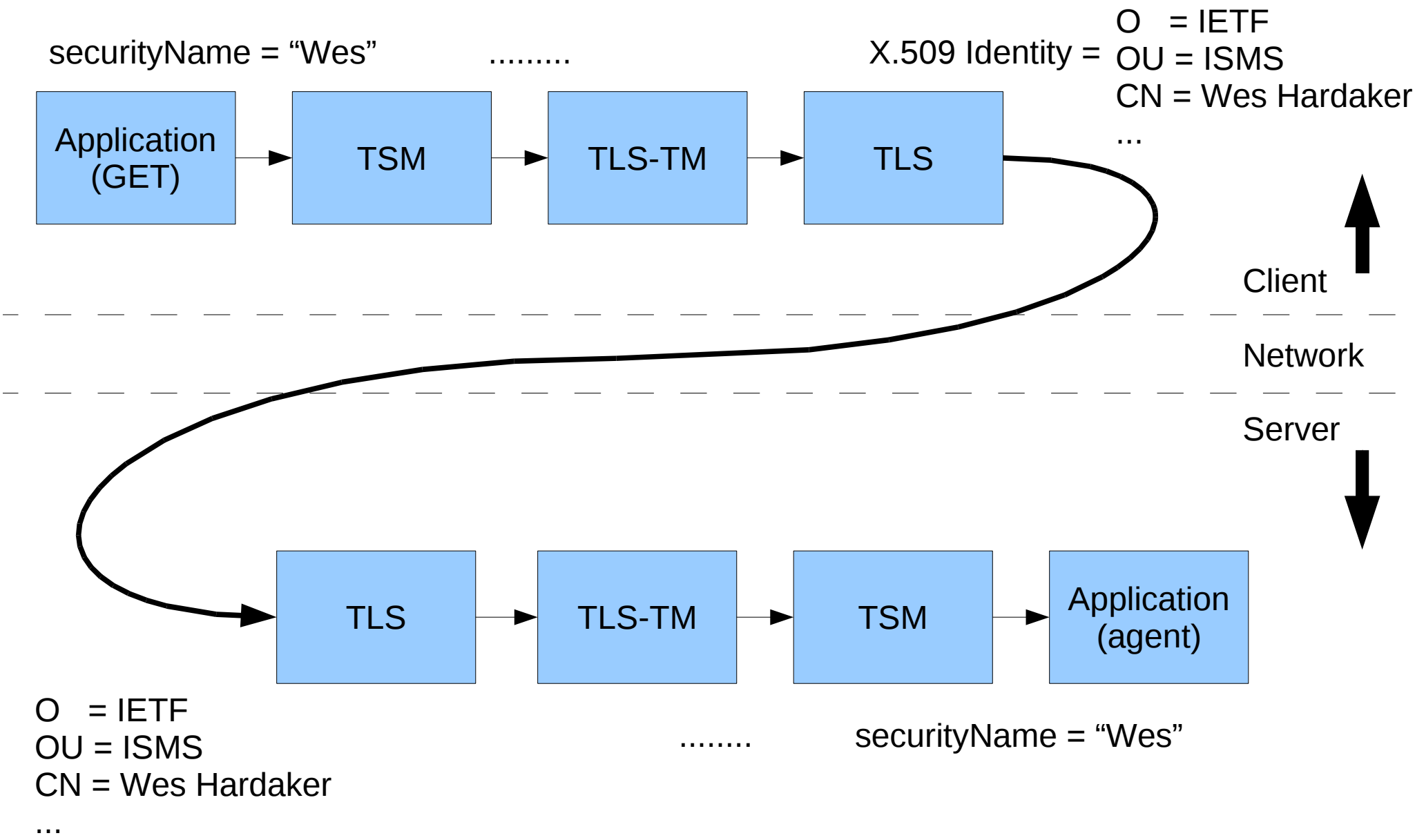
Review:
In other words...

- SSH was fairly straight forward
- An identity string is passed directly in SSH
- ISMS relied on pre-existing SSH configuration
  - SSH already knew where user certificates were
  - SSH already knew a list of remote address and server certificate bindings were
  - IE, configuration was entirely pre-existing

Now on to (D)TLS...

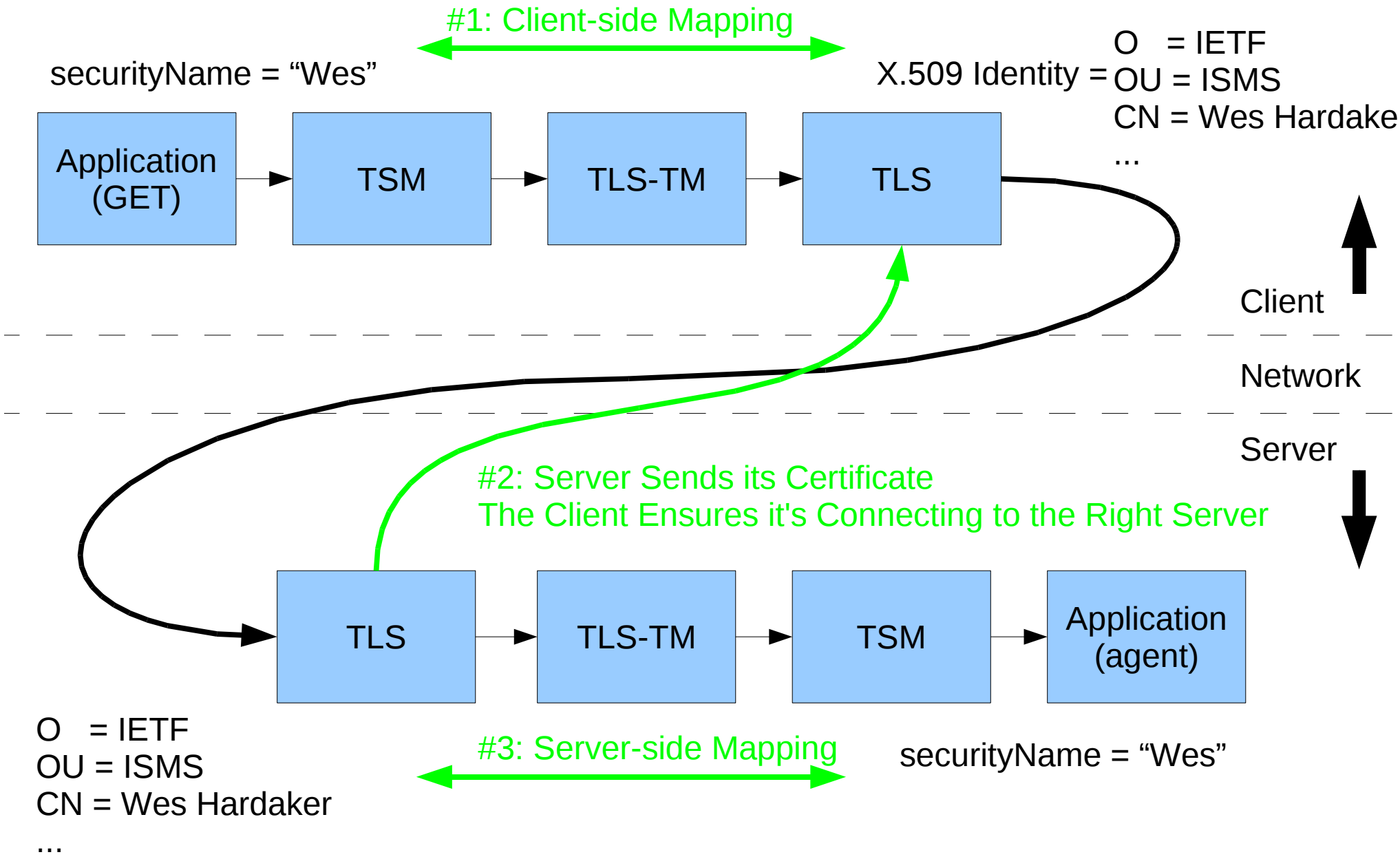
- (D)TLS is:
  - Provides no “I'm Wes” identity field
  - Uses X.509 certificate based authentication
  - Any needed identity information is expected to come from the certificate
- X.509 certificates provide a lot of data:
  - Location, Organizational Information, Name(s), ...
  - No direct easy 1:1 mapping choice

X.509 Identity / securityName



X.509 Identity / securityName

3 issues:



3 Issues

Client Side Certificate Usage

(1) SNMP-TARGET-MIB outputs: securityName

- Which client certificate should be used?

(2) What server certificate should be expected?

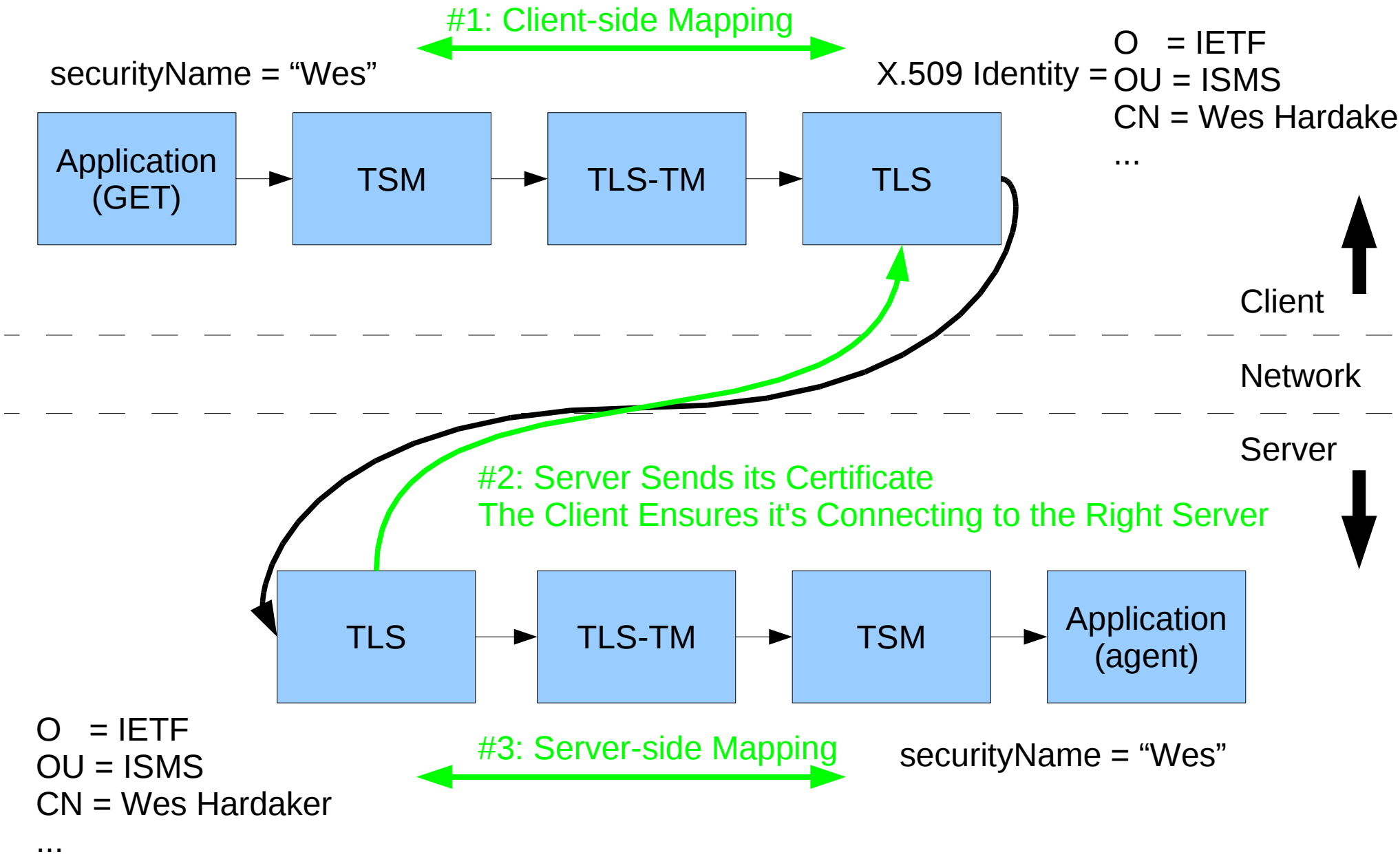
- Can I be sure I'm connecting to the right server?

Server Side Certificate Usage

(3) How to map a client's certificate to a securityName?

X.509 Identity / securityName

3 issues:



(1) Client Sending: tlstmParamsTable

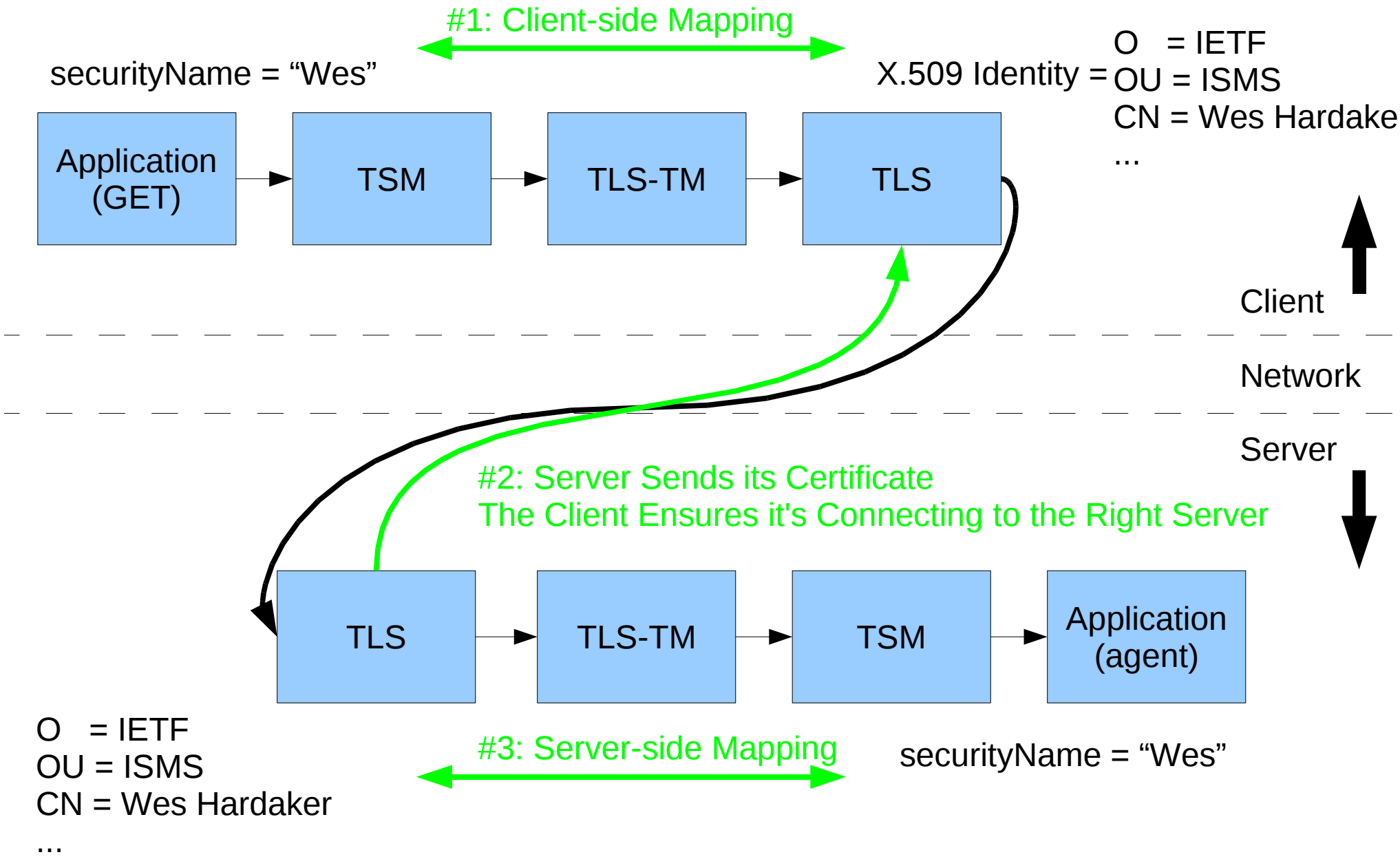
- Extention table to the snmpTargetParamsTable
- Adds Certificate hash type and hash value
- Used to look up a certificate in an implementation dependent certificate store
- (D)TLS connects using this certificate

(1) Client Sending: tlstmParamsTable

- Discussed on the mailing list
  - General agreement that this was the right way to go
  - Minor disagreements about the RowStatus wording
  - Believed Resolved

X.509 Identity / securityName

3 issues:



(2) Client Receiving: Server Certificate Expectations

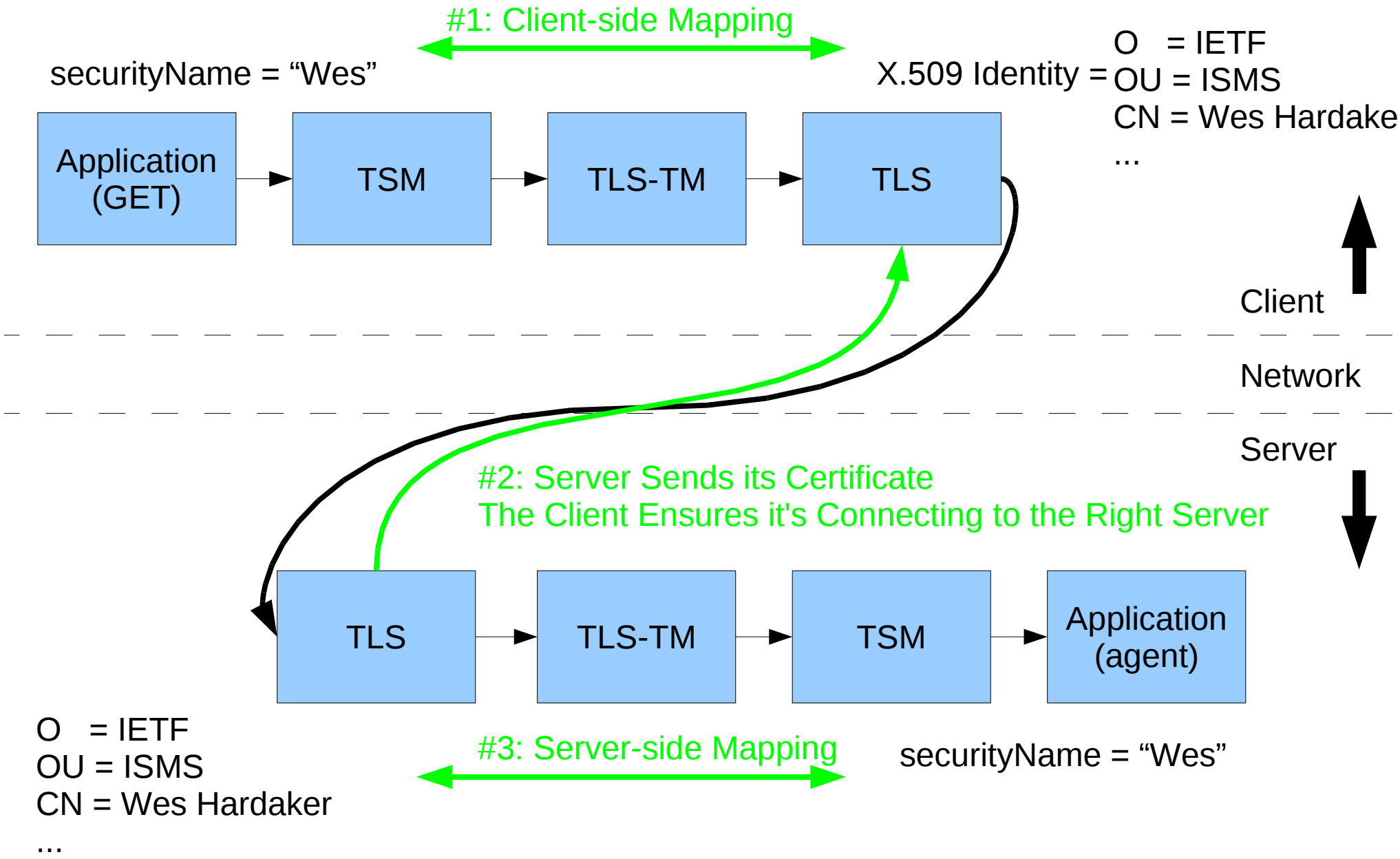
- In SSHTM we assumed known_hosts exists
- (D)TLS MAY use certificate hierarchies
- In (D)TLSTM we can:
  - a) Decide that the CommonName must match
    - (though common, this usage is being deprecated)
  - b) Decide that one subjectAltName must match
  - c) Configure a single certificate hash per server
    - (Would extend the snmpTargetAddrTable)
  - d) Optional a, b, and/or c
  - e) Assume something exists already

(2) Client Receiving: Server Certificate Expectations

- Discussed on the mailing list
  - Not fully resolved?
  - Current agreement seems to be:
    - Text to discuss subjectAltName mapping
      - Our addressType needs to be converted to subjectAltName types
      - (referencing external documentation)
    - Don't standards-support but don't prohibit certificate hash per address
  - Any discussion today?

X.509 Identity / securityName

3 issues:



(3) Server Receiving: Client X.509 Certificates

- Servers will receive a client's X.509 certificate
- Need to map this to a securityName

- Not yet discussed on the mailing list
- (some problems are handled by X.509 handling already, but are referenced here for education; some problems ISMS needs to handle directly)

(3) Server Receiving: Client X.509 Certificates

- Usable X.509 Certificate Fields:
  - Direct Map (doesn't scale well)
  - CommonName (maybe long; deprecating)
  - SubjectAltName (is the future)
- Compounded By Multiple Certificate Issuers
  - Issuer1 CN=”IETF”, User CN=”Wes”
  - Issuer2 CN=”EvilHacker”, User CN=”Wes”
- Result:
  - A certificate to securityName system is needed
  - The good news is that a solution is fairly simple

(3) tlstmCertificateToSNTable

- Ordered list of mapping rules
- Mapping Types:
  - Direct Certificate Hash SN = specified string
  - TrustAnchor Hash SN = CommonName
  - TrustAnchor Hash SN = SubjectAltName
- Very Simple Table
  - 8 columns including index and storage/rowstatus
  - But flexible for small-nets or enterprise-wide

(3) tlstmCertificateToSNTable

TlstmCertificateToSNEntry ::= SEQUENCE {

tlstmCertID	*Unsigned32,*
tlstmCertHashType	X509IdentifierHashType,
tlstmCertHashValue	X509IdentifierHash,
tlstmCertMapType	INTEGER { specified(1), bySubjectAltName(2), byCN(3) },
tlstmCertSecurityName	SnmpAdminString,
tlstmCertStorageType	*StorageType,*
tlstmCertRowStatus	*RowStatus*

}

(3) subjectAltName Considerations

- RFC5280 SubjectAltName definition:

```
SubjectAltName ::= GeneralNames

   GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

   GeneralName ::= CHOICE {

        otherName                       [0]     OtherName,

        rfc822Name                      [1]     IA5String,

        dNSName                         [2]     IA5String,

        x400Address                     [3]     ORAddress,

        directoryName                   [4]     Name,

        ediPartyName                    [5]     EDIPartyName,

        uniformResourceIdentifier       [6]     IA5String,

        iPAddress                       [7]     OCTET STRING,

        registeredID                    [8]     OBJECT IDENT }
```

subjectAltName Considerations

- Choices when looking through subjectAltNames:
  1) Pick first of mappable types: rfc822Name, dNSName
     - What about IP Addresses?
  2) Add a selection column (rfc822Name or dNSName)
     - Again, picking first found if multiple exist
  3) Define our own extension OID for mapping
  4) A combination of the above

- Draft currently does #1
- What happens when length is too long (>32)?

Other (D)TLS Issues/Considerations

- DTLS over UDP provides no session identification
  - (resolved in draft)
  - IE, every packet that arrives on a port could belong to any session that is communicating over that port
  - DTLS-TM Rule: Must have only one session per source-addr, source-port, dest-addr, dest-port
    - (functionally requires clients to use unique port per server)
- Current draft provides a lot of overview text
  - X.509, DTLS, etc.
  - Keep or remove?

Questions?

Secret Slides.

- Shhhhhh
- Stop
- Don't go on.

Certificate Mapping Options

- Don't standardize mapping (ie, no MIB tables)
  - Not a complete solution and difficult deployment
- Standardize Mapping
  - Require conforming certificates
    - (e.g. must have a subjectAltName)
    - Still requires issuer configuration and ordering
    - Reduces reuse of existing infrastructure
  - Provide mapping tables
    - Best trade off