S J P N Trust's

Hirasugar Institute of Technology, Nidasoshi.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka and Affiliated to VTU Belagavi

ECE Dept.
NS
VIII Sem
2017-18

Department of Electronics & Communication Engg.

Course : Network Security **Sem.: 8th (2017-18 EVEN)**

Course Coordinator:

Prof. Nyamatulla M Patel

CRYPTOGRAPHY AND NETWORK SECURITY

Unit-04

Digital Signatures & Authentication Protocols

Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents at the time of signature
  - Must be verifiable by third parties to resolve disputes

Digital Signature Properties

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize & verify
- be computationally infeasible to forge
  - with new message for existing digital signature
  - with fraudulent digital signature for given message
- be practical save digital signature in a storage

Direct Digital Signatures

- involves only the parties: sender and receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message & signature
- security depends on sender's private-key

Arbitrated Digital Signatures

- involves use of arbiter A
  - Sender sends the signed message to arbiter
  - validates any signed message
  - then dated and sent to recipient
- requires suitable level of trust in arbiter
- can be implemented with either private or public-key algorithms
- arbiter may or may not be able to see message

Authentication Protocols

- used to convince parties of each others identity and to exchange session keys
- may be one-way or mutual
- key issues in authenticated key exchange:
  - confidentiality – to protect session keys
  - timeliness – to prevent replay attacks
- published protocols are often found to have flaws and need to be modified

Replay Attacks

- where a valid signed message is copied and later resent
  - simple replay (simply copy and replay later)
  - repetition that can be logged (replay a timestamped message within its valid time window)
  - repetition that cannot be detected (the original message is suppressed and only replayed message arrives at the destination)
  - backward replay without modification (a message is replayed back to the sender; can work if symmetric encryption is used)

Replay Attacks

- countermeasures include
  - use of sequence numbers (generally impractical– each party must remember the last sequence for every other person)
  - timestamps (needs synchronized clocks)
  - challenge/response (using unique nonce)

Using Symmetric Encryption

- as discussed previously, we can use a two-level hierarchy of keys
- usually with a trusted Key Distribution Center (KDC)
  - each party shares own master key with KDC
  - KDC generates session keys used for connections between parties
  - master keys used to distribute these to them

Needham-Schroeder Protocol

- does key distribution using a KDC
- Also performs authentication
- for session between A and B mediated by KDC, protocol overview is:

1. A->KDC: $ID_A \,||\, ID_B \,||\, N_1$
2. KDC -> A: $E_{Ka}[Ks \,||\, ID_B \,||\, N_1 \,||\, E_{Kb}[Ks||ID_A]\,]$
3. A -> B: $E_{Kb}[Ks||ID_A]$
4. B -> A: $E_{Ks}[N_2]$
5. A -> B: $E_{Ks}[f(N_2)]$

Needham-Schroeder Protocol

- used to securely distribute a new session key for communications between A & B
- vulnerable to a replay attack if an old session key has been compromised
  - then message 3 can be resent convincing B that is communicating with A
- modifications to address this require:
  - timestamps (Denning 81)
  - using an extra nonce (Neuman 93)

Using Public-Key Encryption

- have a range of approaches based on the use of public-key encryption
- need to ensure have correct public keys for other parties
- using a central Authentication Server (AS)
- various protocols exist using timestamps or nonces

Denning AS Protocol

- Denning 81 presented the following:
  1. A -> AS: ID_A || ID_B
  2. AS -> A: $E_{PRas}[ID_A||PU_a||T]$ || $E_{PRas}[ID_B||PU_b||T]$
  3. A -> B: $E_{PRas}[ID_A||PU_a||T]$ || $E_{PRas}[ID_B||PU_b||T]$ || $E_{PUb}[E_{PRas}[K_s||T]]$
- note session key is chosen by A, hence AS need not be trusted to protect it
- timestamps prevent replay but requires synchronized clocks

One-Way Authentication

- required when sender & receiver are not in communications at same time (e.g., email)
- have header in clear so can be delivered by email system
- may want contents of body protected & sender authenticated

Using Symmetric Encryption

- One-way authentication protocol:

  1. A->KDC: ID_A || ID_B || N_1

  2. KDC -> A: E_{Ka}[Ks || ID_B || N_1 || E_{Kb}[Ks||ID_A]]

  3. A -> B: E_{Kb}[Ks||ID_A] || E_{Ks}[M]

- does not protect against replays
  - could rely on timestamp in message, though email delays make this problematic

Public-Key Approaches

- if confidentiality is a major concern, can use:

  A->B: $E_{PUb}[Ks] || E_{Ks}[M]$

  – has encrypted session key, encrypted message
- if authentication needed, use a digital signature with a digital certificate:

  A->B: $M || E_{PRa}[H(M)] || E_{PRas}[T||ID_A||PU_a]$

  – with message, signature, certificate

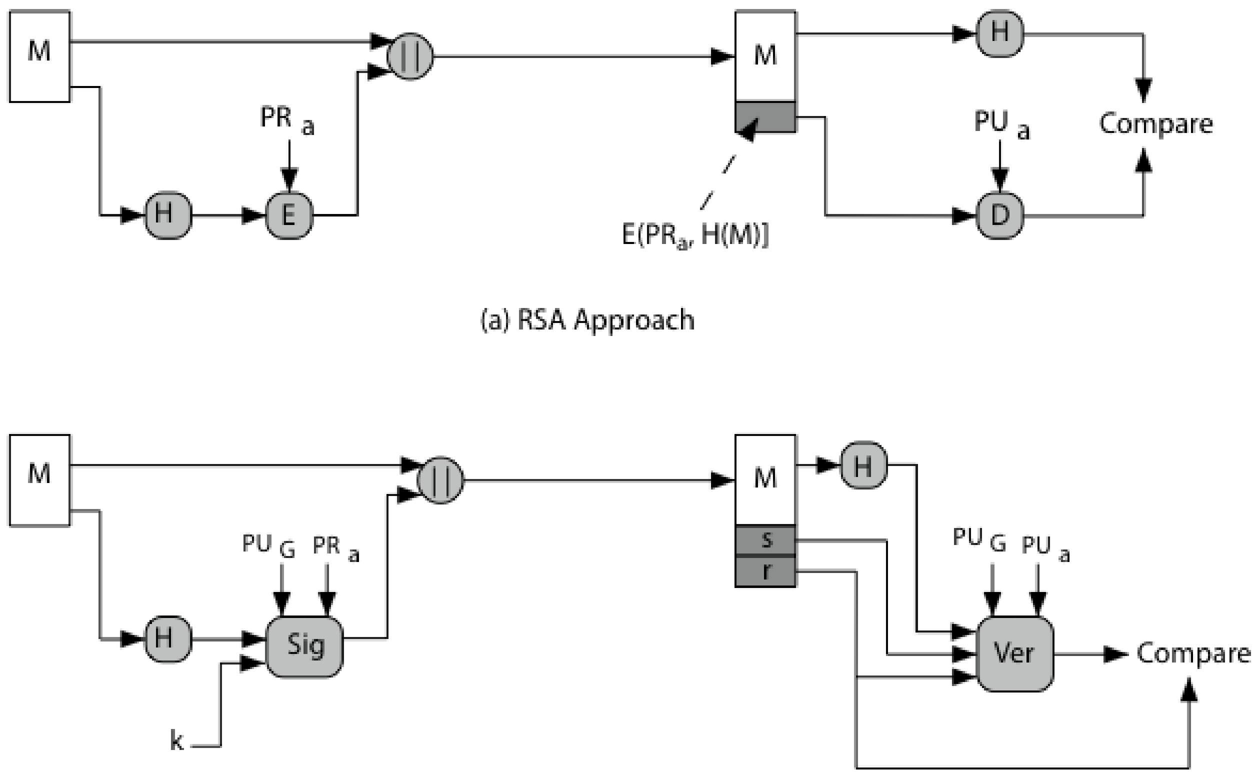
Digital Signature Standard (DSS)

- A digital signature function (can not be used for encryption or key exchange)
- US Govt approved signature scheme
- designed by NIST & NSA in early 90's
- published as FIPS-186 in 1991
- revised in 1993, 1996 & then 2000
- uses the SHA hash algorithm
- DSS is the standard, DSA is the algorithm
- FIPS 186-2 (2000) includes alternative RSA & elliptic curve signature variants

Digital Signature Algorithm (DSA)

- creates a 320 bit signature
- smaller and faster than RSA
- a digital signature scheme only
- security depends on difficulty of computing discrete logarithms

Digital Signature Algorithm (DSA)



(a) RSA Approach

(b) DSS Approach

Digital Signature Algorithm (DSA)

- Sig: a signature function that has four inputs:

a. Hash H
b. A random number k
c. Private key of the sender
d. A global public key (known to a group of communicating principals)

- The signature consists of two parts, r and s.
- At the receiver side, verification is done.

Summary

- have discussed:
  - digital signatures
  - authentication protocols (mutual & one-way)
  - digital signature algorithm and standard

fppt.com

Queries ...?