

# Principles of Information Security

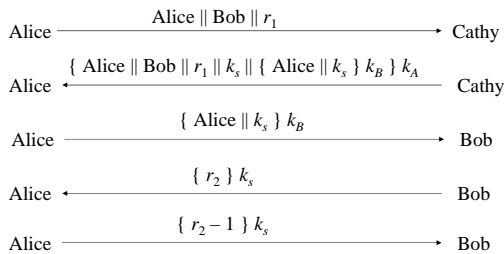
## CS 5323 Lecture 14

Prof. William Winsborough  
November 6, 2008

## Business

- We will have one more midterm near the end of the semester
  - The final will be cumulative and optional
  - If you have done well on the first two midterms, I will base your grade on those exams (plus homework and project)
  - If you need a chance to improve your performance, study the material you didn't master for the midterms and take the final
- Questions from previous lectures?
- Remainder of slides are ©2004 Matt Bishop

## Needham-Schroeder



## Argument: Alice talking to Bob

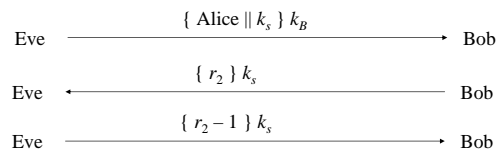
- Second message
  - Enciphered using key only she, Cathy knows
    - So Cathy enciphered it
  - Response to first message
    - As  $r_1$  in it matches  $r_1$  in first message
- Third message
  - Alice knows only Bob can read it
    - As only Bob can derive session key from message
  - Any messages enciphered with that key are from Bob

## Argument: Bob talking to Alice

- Third message
  - Enciphered using key only he, Cathy know
    - So Cathy enciphered it
  - Names Alice, session key
    - Cathy provided session key, says Alice is other party
- Fourth message
  - Uses session key to determine if it is replay from Eve
    - If not, Alice will respond correctly in fifth message
    - If so, Eve can't decipher  $r_2$  and so can't respond, or responds incorrectly

## Denning-Sacco Modification

- Assumption: all keys are secret
- Question: suppose Eve can obtain session key. How does that affect protocol?
  - In what follows, Eve knows  $k_s$



## Solution

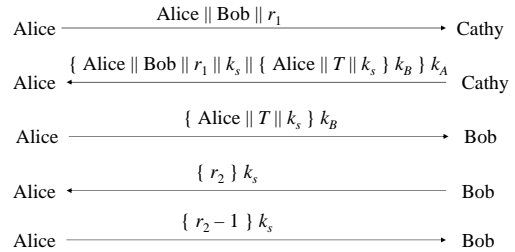
- In protocol above, Eve impersonates Alice
- Problem: replay in third step
  - First in previous slide
- Solution: use time stamp  $T$  to detect replay
- Weakness: if clocks not synchronized, may either reject valid messages or accept replays
  - Parties with either slow or fast clocks vulnerable to replay
  - Resetting clock does *not* eliminate vulnerability

6 November 2008

Winsborough CS 5323 Lecture 14

7

## Needham-Schroeder with Denning-Sacco Modification



6 November 2008

Winsborough CS 5323 Lecture 14

8

## Otway-Rees Protocol

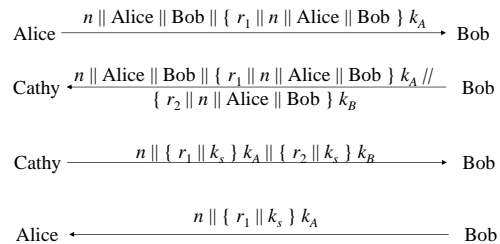
- Corrects problem
  - That is, Eve replaying the third message in the protocol
- Does not use timestamps
  - Not vulnerable to the problems that Denning-Sacco modification has
- Uses integer  $n$  to associate all messages with particular exchange

6 November 2008

Winsborough CS 5323 Lecture 14

9

## The Protocol



6 November 2008

Winsborough CS 5323 Lecture 14

10

## Argument: Alice talking to Bob

- Fourth message
  - If  $n$  matches first message, Alice knows it is part of this protocol exchange
  - Cathy generated  $k_s$  because only she, Alice know  $k_A$
  - Enciphered part belongs to exchange as  $r_1$  matches  $r_1$  in encrypted part of first message

6 November 2008

Winsborough CS 5323 Lecture 14

11

## Argument: Bob talking to Alice

- Third message
  - If  $n$  matches second message, Bob knows it is part of this protocol exchange
  - Cathy generated  $k_s$  because only she, Bob know  $k_B$
  - Enciphered part belongs to exchange as  $r_2$  matches  $r_2$  in encrypted part of second message

6 November 2008

Winsborough CS 5323 Lecture 14

12

## Replay Attack

- Eve acquires old  $k_s$ , message in third step
  - $n \parallel \{ r_1 \parallel k_s \} k_A \parallel \{ r_2 \parallel k_s \} k_B$
- Eve forwards appropriate part to Alice
  - Alice has no ongoing key exchange with Bob:  $n$  matches nothing, so is rejected
  - Alice has ongoing key exchange with Bob:  $n$  does not match, so is again rejected
    - If replay is for the current key exchange, and Eve sent the relevant part *before* Bob did, Eve could simply listen to traffic; no replay involved

6 November 2008

Winsborough CS 5323 Lecture 14

13

## Kerberos

- Authentication system
  - Based on Needham-Schroeder with Denning-Sacco modification
  - Central server plays role of trusted third party (“Cathy”)
- Ticket
  - Issuer vouches for identity of requester of service
- Authenticator
  - Identifies sender

6 November 2008

Winsborough CS 5323 Lecture 14

14

## Idea

- User  $u$  authenticates to Kerberos server
  - Obtains ticket  $T_{u,TGS}$  for ticket granting service (TGS)
- User  $u$  wants to use service  $s$ :
  - User sends authenticator  $A_u$ , ticket  $T_{u,TGS}$  to TGS asking for ticket for service
  - TGS sends ticket  $T_{u,s}$  to user
  - User sends  $A_u$ ,  $T_{u,s}$  to server as request to use  $s$
- Details follow

6 November 2008

Winsborough CS 5323 Lecture 14

15

## Ticket

- Credential saying issuer has identified ticket requester
- Example ticket issued to user  $u$  for service  $s$ 

$$T_{u,s} = s \parallel \{ u \parallel u's \text{ address} \parallel \text{valid time} \parallel k_{u,s} \} k_s$$
 where:
  - $k_{u,s}$  is session key for user and service
  - Valid time is interval for which ticket valid
  - $u$ 's address may be IP address or something else
    - Note: more fields, but not relevant here

6 November 2008

Winsborough CS 5323 Lecture 14

16

## Authenticator

- Credential containing identity of sender of ticket
  - Used to confirm sender is entity to which ticket was issued
- Example: authenticator user  $u$  generates for service  $s$ 

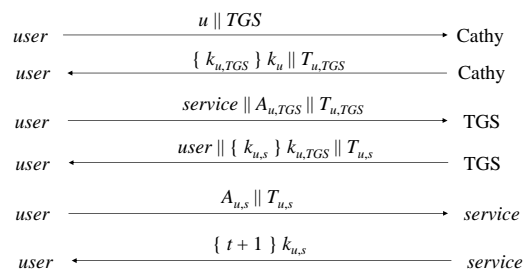
$$A_{u,s} = \{ u \parallel \text{generation time} \parallel k_t \} k_{u,s}$$
 where:
  - $k_t$  is alternate session key
  - Generation time is when authenticator generated
    - Note: more fields, not relevant here

6 November 2008

Winsborough CS 5323 Lecture 14

17

## Protocol



6 November 2008

Winsborough CS 5323 Lecture 14

18

## Analysis

- First two steps get user ticket to use TGS
  - User  $u$  can obtain session key only if  $u$  knows key shared with Cathy
- Next four steps show how  $u$  gets and uses ticket for service  $s$ 
  - Service  $s$  validates request by checking sender (using  $A_{u,s}$ ) is same as entity ticket issued to
  - Step 6 optional; used when  $u$  requests confirmation

6 November 2008

Winsborough CS 5323 Lecture 14

19

## Problems

- Relies on synchronized clocks
  - If not synchronized and old tickets, authenticators not cached, replay is possible
- Tickets have some fixed fields
  - Dictionary attacks possible
  - Kerberos 4 session keys weak (had much less than 56 bits of randomness); researchers at Purdue found them from tickets in minutes

6 November 2008

Winsborough CS 5323 Lecture 14

20

## Public Key Key Exchange

- Here interchange keys known
  - $e_A, e_B$  Alice and Bob's public keys known to all
  - $d_A, d_B$  Alice and Bob's private keys known only to owner
- Simple protocol
  - $k_s$  is desired session key

Alice  $\xrightarrow{\{k_s\} e_B}$  Bob

6 November 2008

Winsborough CS 5323 Lecture 14

21

## Problem and Solution

- Vulnerable to forgery or replay
  - Because  $e_B$  known to anyone, Bob has no assurance that Alice sent message
- Simple fix uses Alice's private key
  - $k_s$  is desired session key

Alice  $\xrightarrow{\{\{k_s\} d_A\} e_B}$  Bob

6 November 2008

Winsborough CS 5323 Lecture 14

22

## Notes

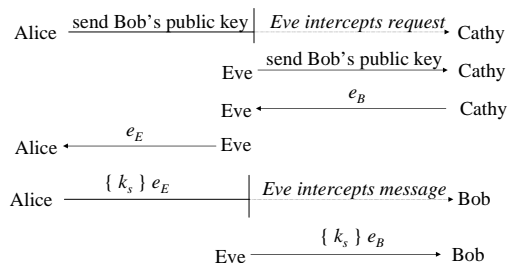
- Can include message enciphered with  $k_s$
- Assumes Bob has Alice's public key, and *vice versa*
  - If not, each must get it from public server
  - If keys not bound to identity of owner, attacker Eve can launch a *man-in-the-middle* attack (next slide; Cathy is public server providing public keys)
    - Solution to this (binding identity to keys) discussed later as public key infrastructure (PKI)

6 November 2008

Winsborough CS 5323 Lecture 14

23

## Man-in-the-Middle Attack



6 November 2008

Winsborough CS 5323 Lecture 14

24

## Cryptographic Key Infrastructure

- Goal: bind identity to key
- Classical: not possible as all keys are shared
  - Use protocols to agree on a shared key (see earlier)
- Public key: bind identity to public key
  - Crucial as people will use key to communicate with principal whose identity is bound to key
  - Erroneous binding means no secrecy between principals
  - Assume principal identified by an acceptable name

6 November 2008

Winsborough CS 5323 Lecture 14

25

## Certificates

- Create token (message) containing
  - Identity of principal (here, Alice)
  - Corresponding public key
  - Timestamp (when issued)
  - Other information (perhaps identity of signer)signed by trusted authority (here, Cathy)

$$C_A = \{ e_A \parallel \text{Alice} \parallel T \} d_C$$

6 November 2008

Winsborough CS 5323 Lecture 14

26

## Use

- Bob gets Alice's certificate
  - If he knows Cathy's public key, he can decipher the certificate
    - When was certificate issued?
    - Is the principal Alice?
  - Now Bob has Alice's public key
- Problem: Bob needs Cathy's public key to validate certificate
  - Problem pushed "up" a level
  - Two approaches: Merkle's tree, signature chains

6 November 2008

Winsborough CS 5323 Lecture 14

27

## Certificate Signature Chains

- Create certificate
  - Generate hash of certificate
  - Encipher hash with issuer's private key
- Validate
  - Obtain issuer's public key
  - Decipher enciphered hash
  - Recompute hash from certificate and compare
- Problem: getting issuer's public key

6 November 2008

Winsborough CS 5323 Lecture 14

28

## X.509 Chains

- Some certificate components in X.509v3:
  - Version
  - Serial number
  - Signature algorithm identifier: hash algorithm
  - Issuer's name; uniquely identifies issuer
  - Interval of validity
  - Subject's name; uniquely identifies subject
  - Subject's public key
  - Signature: enciphered hash

6 November 2008

Winsborough CS 5323 Lecture 14

29

## X.509 Certificate Validation

- Obtain issuer's public key
  - The one for the particular signature algorithm
- Decipher signature
  - Gives hash of certificate
- Recompute hash from certificate and compare
  - If they differ, there's a problem
- Check interval of validity
  - This confirms that certificate is current

6 November 2008

Winsborough CS 5323 Lecture 14

30

## Issuers

- *Certification Authority (CA)*: entity that issues certificates
  - Multiple issuers pose validation problem
  - Alice's CA is Cathy; Bob's CA is Don; how can Alice validate Bob's certificate?
  - Have Cathy and Don cross-certify
    - Each issues certificate for the other

6 November 2008

Winsborough CS 5323 Lecture 14

31

## Validation and Cross-Certifying

- Certificates:
  - Cathy<<Alice>>
  - Dan<<Bob>
  - Cathy<<Dan>>
  - Dan<<Cathy>>
- Alice validates Bob's certificate
  - Alice obtains Cathy<<Dan>>
  - Alice uses (known) public key of Cathy to validate Cathy<<Dan>>
  - Alice uses Cathy<<Dan>> to validate Dan<<Bob>>

6 November 2008

Winsborough CS 5323 Lecture 14

32

## PGP Chains

- OpenPGP certificates structured into packets
  - One public key packet
  - Zero or more signature packets
- Public key packet:
  - Version (3 or 4; 3 compatible with all versions of PGP, 4 not compatible with older versions of PGP)
  - Creation time
  - Validity period (not present in version 3)
  - Public key algorithm, associated parameters
  - Public key

6 November 2008

Winsborough CS 5323 Lecture 14

33

## OpenPGP Signature Packet

- Version 3 signature packet
  - Version (3)
  - Signature type (level of trust)
  - Creation time (when next fields hashed)
  - Signer's key identifier (identifies key to encipher hash)
  - Public key algorithm (used to encipher hash)
  - Hash algorithm
  - Part of signed hash (used for quick check)
  - Signature (enciphered hash)
- Version 4 packet more complex

6 November 2008

Winsborough CS 5323 Lecture 14

34