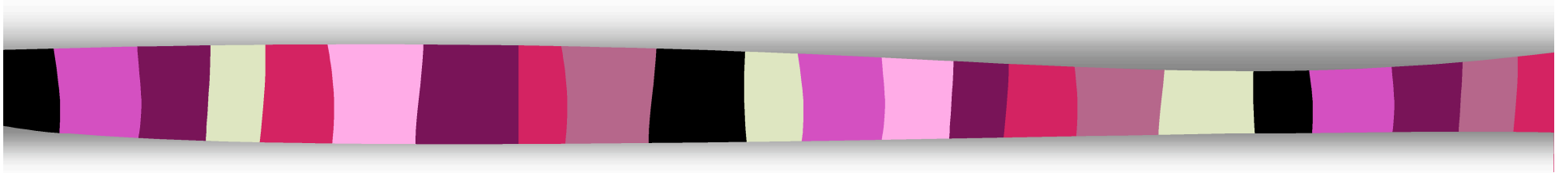


# Cryptography CS 555



## Lecture 19: Authentication Protocols

Department of Computer Sciences  
Purdue University

Cristina Nita-Rotaru

Spring 2005/Lecture 19

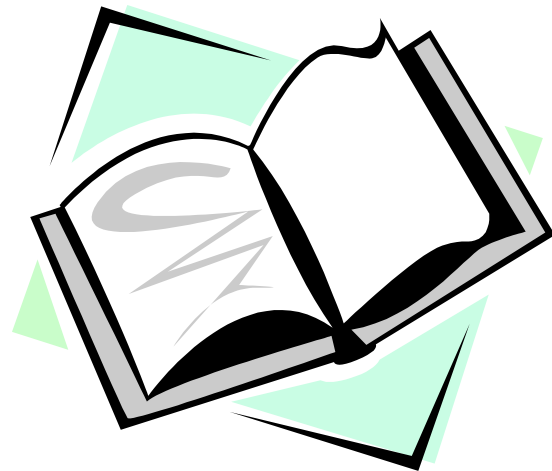
# Lecture Outline

- Authentication protocols: password based, challenge-response, zero-knowledge



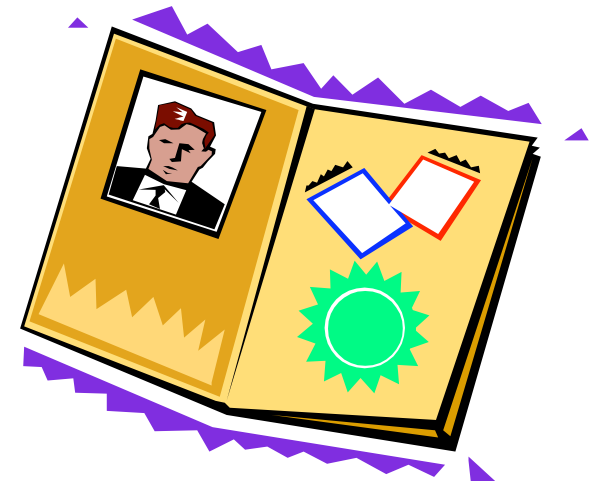
# Recommended Reading

- HAC: Chapter 10 for authentication protocols



# Authentication

- **Entity authentication (identification):**  
the process whereby one party is assured of the identity of a second party involved in a protocol and that the second has actually participated.
- **Data source authentication:**  
represents an indication about the source of the data.



# Properties of Identification Protocols

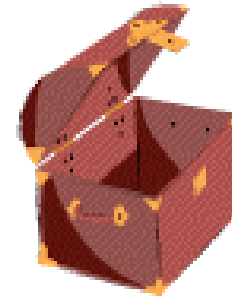
- Reciprocity of identification
- Computational efficiency
- Communication efficiency
- Involvement of a third party
- Nature of trust in the third party
- Nature of security: provable security, zero-knowledge security
- Storage of secrets

# Authentication Using Fixed Passwords

- Client authenticates to a server using a password.
- Passwords must be kept in encrypted password files or as digests
- Strengthen passwords by “salting”
- Passphrases, more complex passwords
- Attacks:
  - Replay of fixed passwords
  - Exhaustive password search
  - Password-guessing and dictionary attacks

# Unix crypt Algorithm

- Used to store Unix passwords
- Information stored in /etc/passwd is:
  - Iterated DES encryption of 0 (64 bits), using the first 8 characters of the password as key
  - 12 bit random salt taken from the system clock time at the password creation
- Why use the salt: to alter the expansion function E of DES, to defend against attacks on DES using off-the-shelf hardware that can crack DES



# Lamport's One-Time Password

Stronger authentication than password-based

- One-time setup:
  - A selects a value  $w$ , a hash function  $H()$ , and an integer  $t$ , computes  $w_0 = H^t(w)$  and sends  $w_0$  to B
  - B stores  $w_0$
- Protocol: to identify to B for the  $i^{\text{th}}$  time,  $1 \leq i \leq t$ 
  - A sends to B:  $A, i, w_i = H^{t-i}(w)$
  - B checks  $i = i_A, H(w_i) = w_{i-1}$
  - if both holds,  $i_A = i_A + 1$

# Challenge-Response Protocols

- Goal: one entity authenticates to other entity proving the knowledge of a secret, 'challenge'
- Time-variant parameters used to prevent replay, interleaving attacks, provide uniqueness and timeliness : nonce (used only once)
- Three types:
  - Random numbers
  - Sequences
  - Timestamp

# Challenge-Response Protocols

- **Random numbers:**
  - pseudo-random numbers that are unpredictable to an adversary;
  - vulnerable to birthday attacks, use larger sample;
  - must maintain state;
  - do not prevent interleaving attacks (parallel sessions)
- **Sequences:**
  - serial number or counters;
  - long-term state information must be maintained by both parties+ synchronization
- **Timestamp:**
  - provides timeliness and detects forced delays;
  - requires synchronized clocks.

# Challenge-Response Protocols Using Digital Signatures

- unilateral authentication with timestamp  
A  $\rightarrow$  B:  $\text{cert}_A, t_A, B, S_A(t_A, B)$
- unilateral authentication with random numbers  
A  $\rightarrow$  B:  $r_B$   
A  $\rightarrow$  B:  $\text{cert}_A, r_A, B, S_A(r_A, r_B, B)$
- mutual authentication with random numbers  
A  $\rightarrow$  B:  $r_B$   
A  $\rightarrow$  B:  $\text{cert}_A, r_A, B, S_A(r_A, r_B, B)$   
A  $\rightarrow$  B:  $\text{cert}_B, A, S_B(r_B, r_A, A)$

# Attacks: Examples

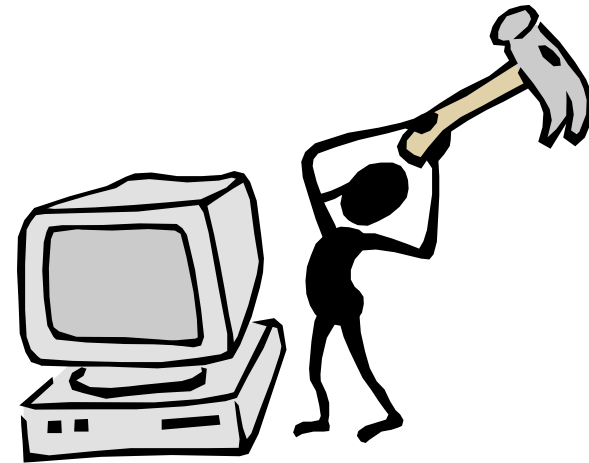
- E1: “Man-in-the-middle” attack on unauthenticated DH
- E2: Reflection attack

## Protocol

1.  $A \rightarrow B : r_A$
2.  $B \rightarrow A : E_k(r_A, r_B)$
3.  $A \rightarrow B : r_B$

## Attack

1.  $A \rightarrow E : r_A$
2.  $E \rightarrow A : r_A$  : Starting a new session
3.  $A \rightarrow E : E_k(r_A, r_A')$  : Reply of (2)
4.  $E \rightarrow A : E_k(r_A, r_A')$  : Reply of (1)
5.  $A \rightarrow E : r_A'$



# Attacks: Examples (cont.)

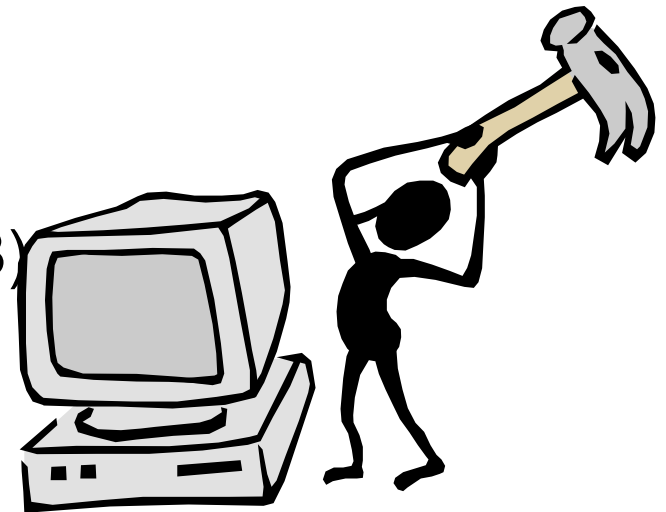
- E3: Interleaving attacks

## Protocol

1. A  $\square$  B :  $r_A$
2. B  $\square$  A :  $r_B, S_B(r_B, r_A, A)$
3. A  $\square$  B :  $r_A', S_A(r_A', r_B, B)$

## Attack

1. E  $\square$  B :  $r_A$
2. B  $\square$  E :  $r_B, S_B(r_B, r_A, A)$
3. E  $\square$  A :  $r_B$
4. A  $\square$  E :  $r_A', S_A(r_A', r_B, B)$
5. E  $\square$  B :  $r_A', S_A(r_A', r_B, B)$



# Zero-Knowledge Protocols

- **Motivation:**
  - Password-based protocols: when Alice authenticates to a server, she gives her password, so the server can then impersonate her.
  - Challenge-response improves on this, but still reveals partial information.
- **Zero-knowledge protocols:** allows a prover to prove that it possesses a secret without revealing any information of use to the verifier.

# General Mechanism

- Many protocols consists of repeating n times the following three message rounds:
  - **Prover sends to Verifier a Witness**  
The prover selects a random element from a pre-defined set as its *secret commitment* and from this computes a *public witness*. The randomness creates unrepeatable execution histories.
  - **Verifier sends to Claimant a Challenge**  
The verifier probabilistically tests this by asking questions. If the claimant is the one it claims to be, then it can answer all questions successfully.
  - **Prover sends to Verifier a Response**  
The verifier checks the answer for accuracy. Answers do not provide information about the secret commitment

# Fiat-Shamir Identification Protocol

## Setup:

- 1) A trusted server T makes public  $n = pq$ , keeping secrets  $p$  and  $q$  (selected as in RSA)
- 2) Every claimant A selects  $0 < s < n$ ,  $\gcd(s, n) = 1$ , computes  $v = s^2 \bmod n$  and registers  $v$  with the server T

## Protocol run $t$ rounds, proof accepted if all $t$ rounds succeed

1. A selects  $r$ ,  $0 < r < n$ , and sends to B  $x = r^2 \bmod n$
2. B randomly selects  $e = 0$  or  $e = 1$ , and sends  $e$  to A
3. If  $e = 0$ , A sends  $y = r$ , else sends  $y = rs \bmod n$
4. B rejects  $y = 0$ , otherwise accept if  $y^2 \equiv x v^e \bmod n$

Based on difficulty of extracting square roots modulo large numbers for which the factorization is not known

# Schnorr Identification Protocol

Like DSA select public  $(p, q, \alpha, \beta)$ , and  $t$ ,  
 $t$  defines the security of the system

**Setup:** a claimant  $A$  has an identifier  $I_A$ , private  $a$ ,  $0 < a < q$ ,  $v = \alpha^{-a} \bmod p$ , obtain certificate that provides authenticity of  $v$ ,  $\text{cert}_A$

**Protocol:**

$A$  selects  $0 < r < q$ , computes  $x = \alpha^r \bmod p$ ,

$B$  returns challenge  $1 \leq e \leq 2^t \leq q$

$A$  sends  $y = (ae + r) \bmod q$

$B$  computes  $z = \alpha^y v^e \bmod p$ , accept  $A$  if  $z = x$

$1 \leq e \leq 2^t \leq q$

# Summary

- Replay and interleaving attacks prevented using timestamps and unique protocol run identifiers



# Next Lecture...

- Network Authentication services: Kerberos
- Secure communication protocols: SSL and IPSec.

