CS 161 – Authentication Protocols

27 September 2006

Zero knowledge review

- Goal: authenticate without leaking any information
- What you need to know about Rabin signatures:
  - Squares mod pq have four square roots (r, -r, s, -s)
  - If we add together ±r and ±s
  - And take the greatest common divisor with pq
  - We get p or q
- GCD(pq, ±r + ±s) = p or q

Leaky protocols

- Many protocols leak information
- For example, consider the following authentication protocol:

A $\rightarrow$ B: Prove you are Bob, sign message M
B $\rightarrow$ A: Sign(M, B)

- Now Alice has some information she didn't have before
- She has Sign(M, B)
- Perfect for what kind of attack?

Zero-knowledge protocol

- Idea: interactive proof
- At the end of the proof, A is convinced B knows a proof of fact F
- But A has no information about that proof

How to prove identity using zero-knowledge

- B publishes b^2 mod pq
- B $\rightarrow$ A: r^2 mod pq (random r)
- A flips coin
- A $\rightarrow$ B: coin flip
- If heads
  - B $\rightarrow$ A: r mod pq
  - A verifies $(r \bmod pq)^2 = r^2 \bmod pq$
- If tails
  - B $\rightarrow$ A: rb mod pq
  - A verifies $(rb \bmod pq)^2 = (r^2)(b^2) \bmod pq$

Comments

1. This is an easy-to-perform protocol

2. After each round, convinced with 50% probability

   If B knows both rb & r (mod pq), he knows rb/r (mod pq)

   Fake-B will be caught 50% of the time

3. A learns nothing – if she does, she could just generate pairs <r, r^2>on her own. (Or, <rb, $(rb)^2$>.)

Authentication

- Alice and Bob love each other, but they live far apart
- We've learned how they can encrypt their messages
- How can they make sure they are talking to each other?
- This is the question of authentication

Types of authentication

- End user → End user (Alice & Bob)
- End user → Local computer (login)
- End user → Remote computer (web site login)
- Computer → Computer (DRM)
- Local computer → End user (fake ATM check)
- Remote computer → End user (phishing check)

More types of authentication

- Software authentication: tougher!
- Still under active development
  - (we may talk about it at the end of class)
- “Trusted computing”

Authentication is complicated

- It is surprisingly hard to get authentication right
- Most first, second, & third attempts get it wrong
- Ph.D. level courses on authentication don’t cover all
- This lecture will talk about the basics

Encrypting digital content

- Goal: prevent people from copying digital content:
  - Contemporary high-definition TV sets accept HDMI with HDCP
  - (high definition copy protection)
- Handshake to authenticate recipient
  - enforces copy protection
- Older HD TVs don't accept HDCP
- Rules say: HDCP cannot be converted to analogue.

HDCP strippers

- SPATZ-TECH (I am not making this up)
  - Makes DVI (HDMI equivalent) repeater
  - Called DVI Magic
  - Strips HDCP

HDCP strippers continued

- MPAA could revoke SPATZ-TECH's key
  - Then SPATZ-TECH could no longer authenticate
- Revocation list is contained in every HD broadcast
  - every HD DVD.
- Equipment suddenly stops working

Public key authentication is tricky

$A \rightarrow B : \{\text{random message}\}_B$

$B \rightarrow A : \{\text{random message}\}$

What's wrong with this?

Ultimate public key authentication

- We learned zero knowledge authentication
- But it is patented & slow
- What if we want something more streamlined?

Original Needham-Schroeder (Keberos)

- We need a trusted server S
- Alice shares (symmetric) key a with S
- Bob shares (symmetric) key b with S

A → S: { "I want Bob" }$_a$

S → A : { "Use temp key" t; "send to Bob this ticket:"
{ "This is Alice using temporary key" t }$_b$
}$_a$

A → B : { "This is Alice using temporary key" t }$_b$

A ↔ B : { "I love you" }$_t$

Problems with original N-S

- Needham-Schroeder reigned supreme for many years
- But then people noticed a problem
- Replay attack

Bad Guy $\rightarrow$ B : { "This is Alice using temporary key" t $\}_b$

Bad Guy $\leftrightarrow$ B : { "I love you" $\}_t$

Solution: nonces

- One needs to add nonces (such as a timestamp TS):

A $\rightarrow$ S: { "I want Bob", TS $\}_a$

S $\rightarrow$ A : { "Use temp key" t; "send to Bob this ticket:", TS
{ "This is Alice using temporary key" t, TS $\}_b$
$\}_a$

A $\rightarrow$ B : { "This is Alice using temporary key" t, TS $\}_b$

A $\leftrightarrow$ B : { "I love you", TS $\}_t$

Problems with revised N-S

- Requires a trusted third party
- Requires real-time access to trusted third party

Authentication: still a problem

- Most attacks we see today are authentication attacks
  - (often on passwords)
  - Phishing
  - Spyware password stealing
  - Bogus web sites
- We need better solutions