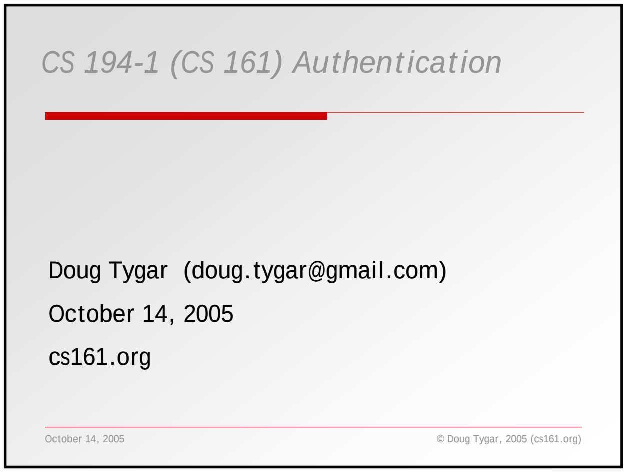
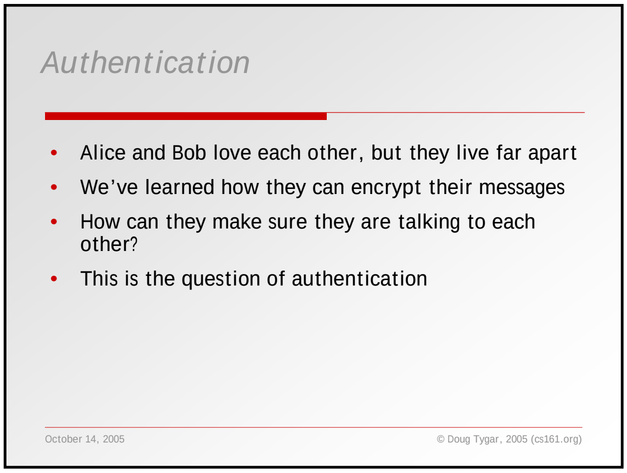


CS 194-1 (CS 161) Authentication

Doug Tygar (doug.tygar@gmail.com)

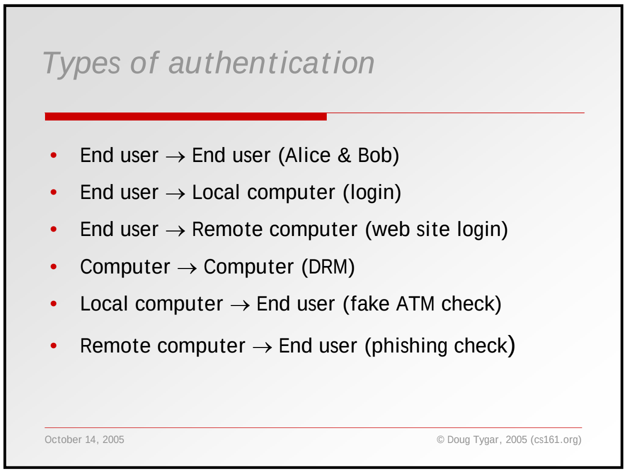
October 14, 2005

cs161.org



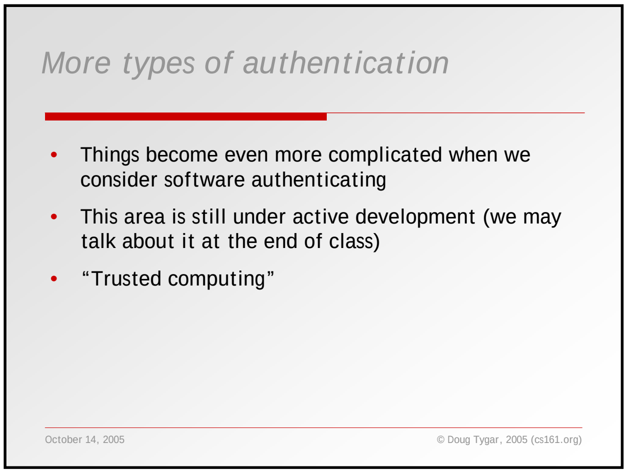
Authentication

- 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



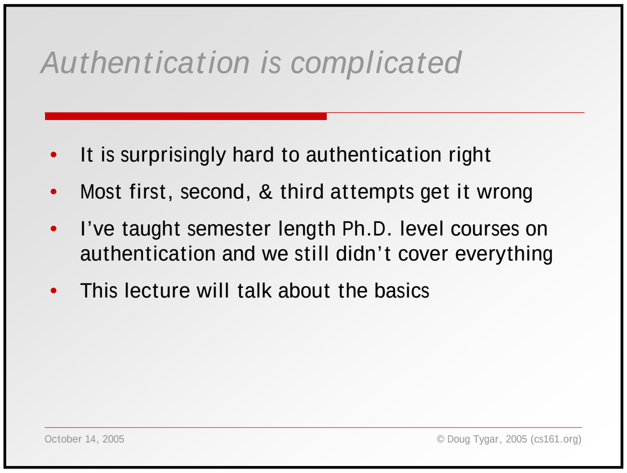
Types of authentication

- 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)



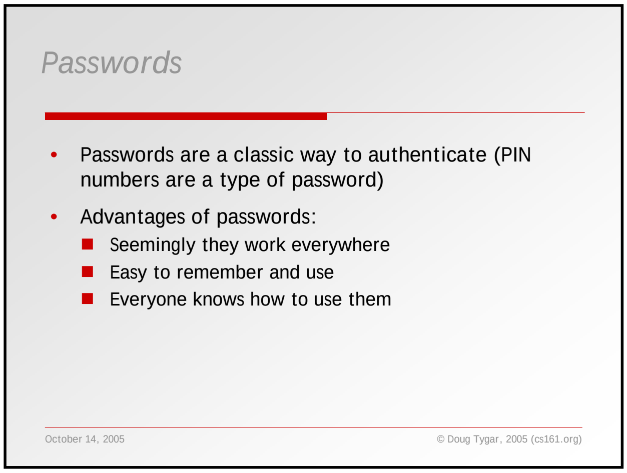
More types of authentication

- Things become even more complicated when we consider software authenticating
- This area is still under active development (we may talk about it at the end of class)
- "Trusted computing"



Authentication is complicated

- It is surprisingly hard to authentication right
- Most first, second, & third attempts get it wrong
- I've taught semester length Ph.D. level courses on authentication and we still didn't cover everything
- This lecture will talk about the basics



Passwords

- Passwords are a classic way to authenticate (PIN numbers are a type of password)
- Advantages of passwords:
  - Seemingly they work everywhere
  - Easy to remember and use
  - Everyone knows how to use them

Problems with passwords

- If password is sent in the clear, can be intercepted
- If password is encrypted, requires establishment of encryption key
- People choose bad passwords
  - E.g., “susan” or “f***you”
- Passwords are easily observed
- Passwords can be sniffed by spyware

Two-factor authentication

- Use passwords plus something else
- Biometrics
  - Retinal or iris scans, hand geometry, voice prints, handwriting analysis, etc.
  - Not clear this works very well
- One time passwords/token
  - RSA Inc makes these
  - Something else to lose!

Notation

$A \rightarrow B : \{m\}_K$ means Alice sends to Bob message m encrypted with key K (symmetric or asymmetric)

A & B are Alice and Bob’s public keys

a & b are Alice and Bob’s private keys

sa (& sb) are symmetric keys between Alice (Bob) and a trusted Server S

t is a temporary key

Public key review

- Public keys can be served by a directory
  - (this never works – why?)
- Or Public keys can be served through certificates:

{ *Doug Tygar’s public key is …*
Love, Arnold Schwartzeneger } Arnold

What’s wrong with asymmetry

- It is slower than symmetric crypto
- It is more expensive to build than symmetric crypto
- Revocation

Revocation problems

- January 2001: Verisign issued two bogus Class 3 certificates for “Microsoft Corporation”
- However the recipient was not Microsoft
- Windows had no way to revoke bogus certificates
- Ultimate solution: issue a patch to revoke

Revocation problems redux

- The motion picture industry (MPAA) wants to protect high-definition versions of movies
- Communication for digital transmission of video: HDMI (a superset of DVI)
- Current DVDs have data at 480i (480 scan lines – interlaced)
- High definition DVDs (and broadcasts) will be 720p (720 scan lines – progressive) or 1080i (1080 scan lines – interlaced)

Encrypting digital content

- To prevent people from copying digital content, contemporary high-definition TV sets accept HDMI with HDCP (high definition copy protection)
- This uses a handshake to authenticate the 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) has made a DVI (HDMI equivalent) repeater called DVI Magic that strips HDCP:

HDCP strippers continued

- To address this, MPAA can revoke SPATZ-TECH's key so SPATZ-TECH can no longer authenticate
- Revocation list is contained in every high-definition broadcast; every high-definition DVD.
- Equipment suddenly stops working

Public key authentication is tricky

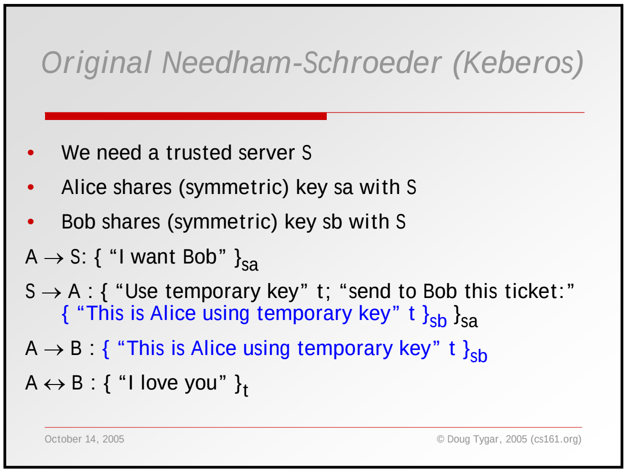
$A \rightarrow B$: {random message}$_B$

$B \rightarrow A$: {random message}

What's wrong with this?

Ultimate public key authentication

- Prof. Vazirani discussed ultimate asymmetric authentication method
  - zero-knowledge authentication.
- But that technique is patented, slow, and requires extensive infrastructure
- What if we want something more streamlined?



Original Needham-Schroeder (Keberos)

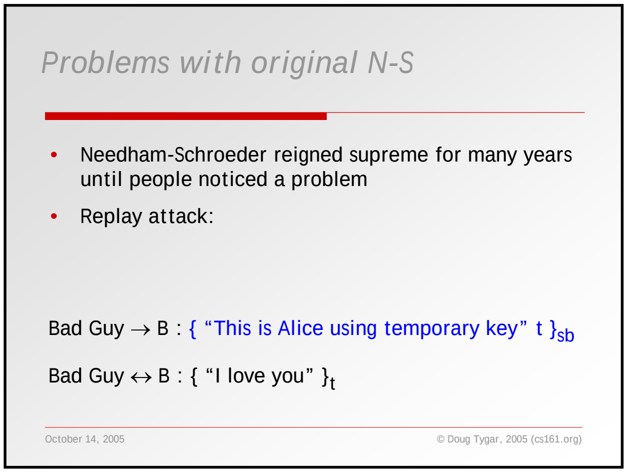
- We need a trusted server S
- Alice shares (symmetric) key sa with S
- Bob shares (symmetric) key sb with S

A → S: { "I want Bob" $\}_{sa}$

S → A : { "Use temporary key" t; "send to Bob this ticket:"
{ "This is Alice using temporary key" t $\}_{sb}$ $\}_{sa}$

A → B : { "This is Alice using temporary key" t $\}_{sb}$

A ↔ B : { "I love you" $\}_{t}$

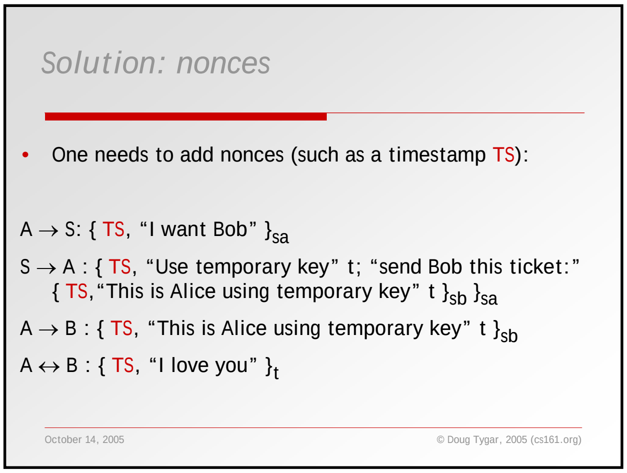


Problems with original N-S

- Needham-Schroeder reigned supreme for many years until people noticed a problem
- Replay attack:

Bad Guy → B : { "This is Alice using temporary key" t $\}_{sb}$

Bad Guy ↔ B : { "I love you" $\}_{t}$



Solution: nonces

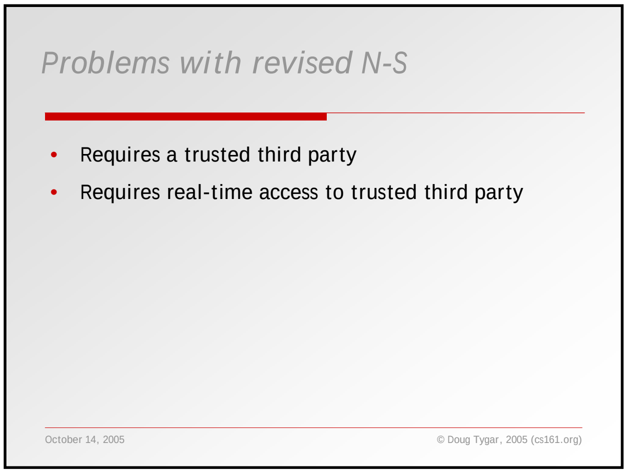
- One needs to add nonces (such as a timestamp TS):

A → S: { TS, "I want Bob" $\}_{sa}$

S → A : { TS, "Use temporary key" t; "send Bob this ticket:"
{ TS,"This is Alice using temporary key" t $\}_{sb}$ $\}_{sa}$

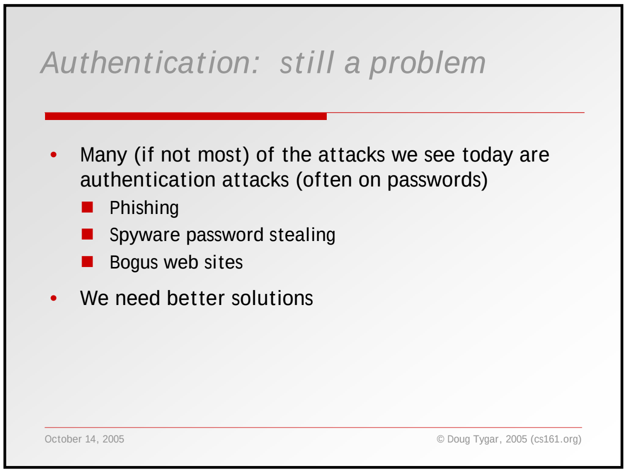
A → B : { TS, "This is Alice using temporary key" t $\}_{sb}$

A ↔ B : { TS, "I love you" $\}_{t}$



Problems with revised N-S

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



Authentication: still a problem

- Many (if not most) of the attacks we see today are authentication attacks (often on passwords)
  - Phishing
  - Spyware password stealing
  - Bogus web sites
- We need better solutions