CSE 486/586 Distributed Systems
Security --- 2

Steve Ko
Computer Sciences and Engineering
University at Buffalo

Recap

- Three types of functions
  - Cryptographic hash, symmetric key crypto, asymmetric key crypto
- Cryptographic hash
  - Easy to compute *h(m)*
  - Hard to find an *m*, given *h(m)*
  - Hard to find two values that hash to the same *h(m)*
- How to find collisions?
  - Birthday paradox: for 50% prob. & m bits, ~ $2^{m/2}$ numbers
- Symmetric key crypto
  - MAC: Compute H = AES_K(SHA1 (M)) & Send <M, H>
- Asymmetric key crypto
  - Guarantees rely on computational hardness

Recap: Digital Signatures

- Method
  - Signer: compute H = RSA_{K}(SHA1(M)) & send <M, H>
  - Verifier: compute H' = $RSA_{K'}$(H) & verify H' == SHA1(M)
- Not just integrity, but also authenticity

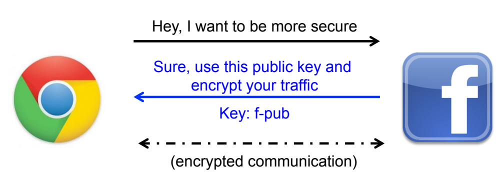
Heard of Firesheep?

- Firesheep
  - A Firefox extension
  - A packet sniffer to intercept unencrypted cookies from certain websites (such as Facebook and Twitter)
  - Allows the user to take on the log-in credentials of the victim
- Solution?
  - Encrypt your traffic!
  - This is before facebook started using https, but now facebook uses https.

"Securing" HTTP

- Threat model
  - Eavesdropper listening on conversation (confidentiality)
  - Man-in-the-middle modifying content (integrity)
  - Adversary impersonating desired website (authentication, and confidentiality)
- Enter HTTP-S
  - HTTP sits on top of secure channels
  - All (HTTP) bytes written to secure channel are encrypted and authenticated

Encrypted Communication



- What is wrong with this?
  - How do you know you're actually talking to facebook and f-pub belongs to facebook?

Digital Certificates

- A digital certificate is a statement signed by a third party principal, and can be reused
  - e.g., Verisign Certification Authority (CA)
- To be useful, certificates must have:
  - A standard format, for construction and interpretation
  - A protocol for constructing chains of certificates
  - A trusted authority at the end of the chain
- Example
  - When facebook sends you the public key, it also sends a signature for the public key signed by Verisign.
  - You pre-store Verisign's public keys & certificates (self-signed by Verisign), i.e., you have already established trust with Verisign.
  - Use Verisign's public key to verify facebook's public key.

On My Mac…

X.509 Certificates

- The most widely used standard format for certificates
- Format
  - Subject: Distinguished Name, Public Key
  - Issuer: Distinguished Name, Signature
  - Period of validity: Not Before Date, Not After Date
  - Administrative information: Version, Serial Number
  - Extended information
- Binds a public key to the subject
  - A subject: person, organization, etc.
- The binding is in the signature issued by an issuer.
  - You need to either trust the issuer directly or indirectly (by establishing a *root of trust*).

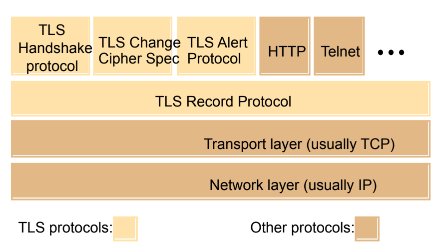
X.509 Certificates

```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       Validity
           Not Before: Jul  9 16:04:02 1998 GMT
           Not After : Jul  9 16:04:02 1999 GMT
       Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
                OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
                   33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
                   66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
                   70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
                   16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
                   c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
                   8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
                   d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
                   e8:35:1c:9e:27:52:7e:41:8f
               Exponent: 65537 (0x10001)
   Signature Algorithm: md5WithRSAEncryption
       93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
       92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
       ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
       d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
       0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
       5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
       8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
       68:9f
```


Transport Layer Security (TLS)

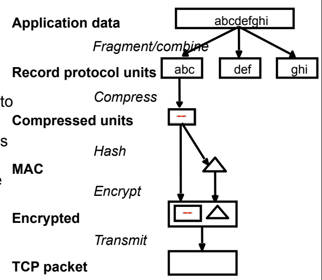
- SSL (Secure Socket Layer) was developed by Netscape for electronic transaction security.
- SSL was adopted as TLS as an Internet standard.
- A protocol layer is added below the application layer for:
  - Negotiating encryption and authentication methods.
  - Bootstrapping secure communication
- It consists of two layers:
  - The Record Protocol Layer implements a secure channel by encrypting and authenticating messages
  - The Handshake Layer establishes and maintains a secure session between two nodes.

TLS Protocol Stack

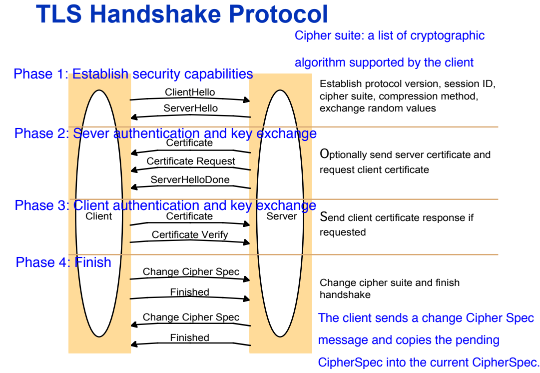


TLS Record Protocol

- The record protocol takes an application message to be transmitted,
  - fragments the data into manageable blocks,
  - optionally compresses the data,
  - computes a message authentication code (MAC),
  - encrypts and
  - adds a header.



TLS Handshake Protocol



CSE 486/586 Administrivia

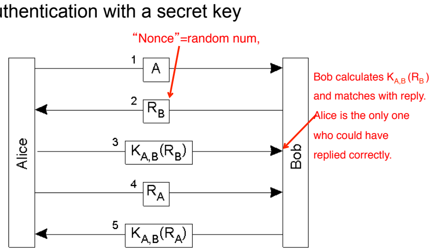
- CSE 622 Advanced Computer Systems
  - Probably on Android platform
  - Is open for registration
- PA4 due this Friday @ 2:59pm.
  - Will target finishing up grading next week.
- PA3 grading underway
  - Will be done this week.
- Final: 5/6, Monday, 3:30pm – 6:30pm
  - Davis 101
  - Everything up to this Friday
- Anonymous feedback form still available.
- Please come talk to me!

Authentication

- Use of cryptography to have two principals verify each others' identities.
  - Direct authentication: the server uses a shared secret key to authenticate the client.
  - Indirect authentication: a trusted authentication server (third party) authenticates the client.
  - The authentication server knows keys of principals and generates temporary shared key (ticket) to an authenticated client. The ticket is used for messages in this session.
    - E.g., Verisign servers

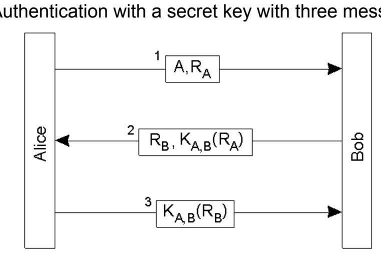
Direct Authentication

- Authentication with a secret key



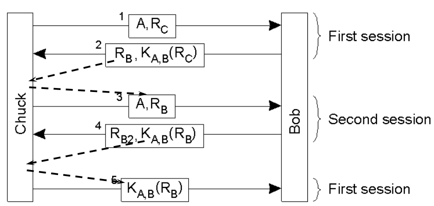
"Optimized" Direct Authentication

- Authentication with a secret key with three messages



- Anything wrong with this?

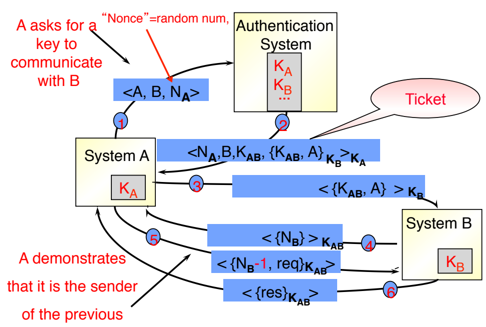
Reflection Attack



Needham-Schroeder Authentication

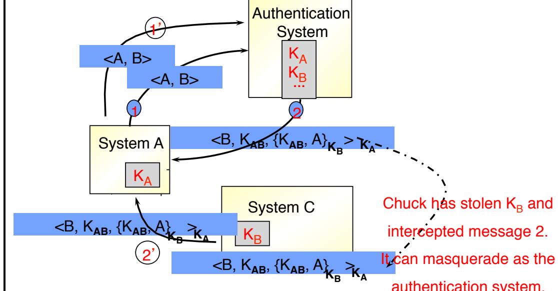
- An authentication server provides secret keys.
  - Every client shares a secret key with the server to encrypt their channels.
- If a client A wants to communicate with another client B,
  - The server sends a key to the client A in two forms.
  - First, in a plain form, so that the client A can use it to encrypt its channel to the client B.
  - Second, in an encrypted form (with the client B's secret key), so that the client B can know that the key is valid.
  - The client A sends this encrypted key to the client B as well.
- Basis for Kerberos

Needham-Schroeder Authentication



Nonce N_A in Message 1

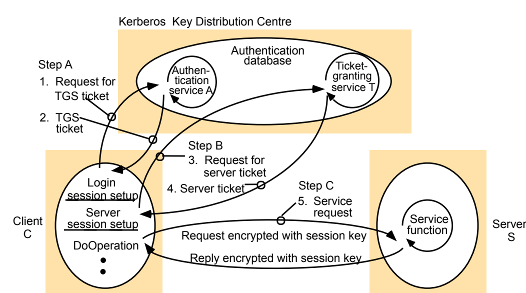
Because we need to relate message 2 to message 1



Kerberos

- Follows Needham-Schroeder closely
- Time values used for nonces
  - To prevent replay attacks
  - To enforce a lifetime for each ticket
- Very popular
  - An Internet standard
  - Default in MS Windows

Kerberos



Summary

- Digital certificates
  - Binds a public key to its owner
  - Establishes a chain of trust
- TLS
  - Provides an application-transparent way of secure communication
  - Uses digital certificates to verify the origin identity
- Authentication
  - Needham-Schroeder & Kerberos

Acknowledgements

- These slides contain material developed and copyrighted by Indranil Gupta (UIUC), Jennifer Rexford (Princeton) and Michael Freedman (Princeton).