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

Background: Digital Signatures

- Asymmetric crypto
- 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

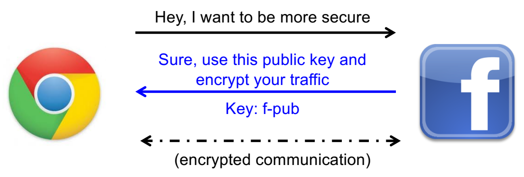
Heard of Firesheep?

- 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

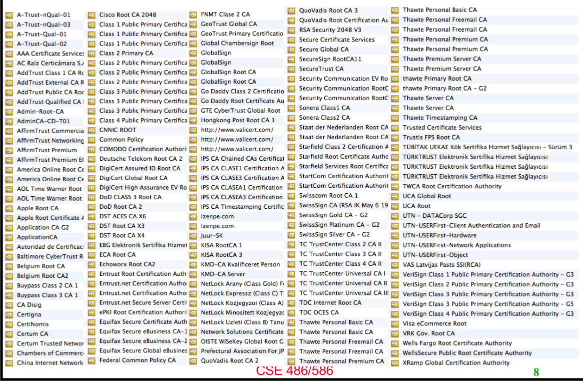

Hey, I want to be more secure
Sure, use this public key and encrypt your traffic
Key: f-pub
(encrypted communication)

- 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…

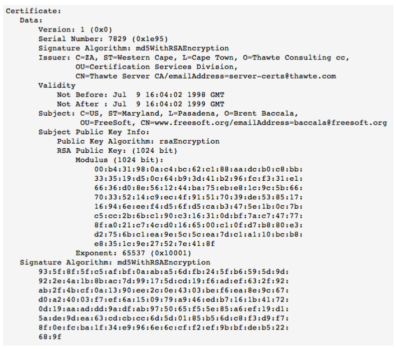


A-Trust-nQual-01
A-Trust-nQual-03
A-Trust-Qual-01
A-Trust-Qual-02
AAA Certificate Services
AC Raíz Certicámara S.A
AddTrust Class 1 CA Ro
AddTrust External CA R
AddTrust Public CA Roo
AddTrust Qualified CA
Admin-Root-CA
AdminCA-CD-T01
AffirmTrust Commercia
AffirmTrust Networking
AffirmTrust Premium
AffirmTrust Premium EC
America Online Root Ce
America Online Root Ce
AOL Time Warner Root
AOL Time Warner Root
Apple Root CA
Apple Root Certificate A
Application CA G2
ApplicationCA
Autoridad de Certificaci
Baltimore CyberTrust R
Belgium Root CA
Belgium Root CA2
Buypass Class 2 CA 1
Buypass Class 3 CA 1
CA Disig
Certigna
CertiNomis
Certum CA
Certum Trusted Networ
Chambers of Commerc
China Internet Network
Cisco Root CA 2048
Class 1 Public Primary Certifica
Class 1 Public Primary Certifica
Class 1 Public Primary Certifica
Class 2 Primary CA
Class 2 Public Primary Certifica
Class 2 Public Primary Certifica
Class 2 Public Primary Certifica
Class 3 Public Primary Certifica
Class 3 Public Primary Certifica
Class 3 Public Primary Certifica
Class 4 Public Primary Certifica
CNNIC ROOT
Common Policy
COMODO Certification Authori
Deutsche Telekom Root CA 2
DigiCert Assured ID Root CA
DigiCert Global Root CA
DigiCert High Assurance EV Ro
DoD CLASS 3 Root CA
DoD Root CA 2
DST ACES CA X6
DST Root CA X3
DST Root CA X4
EBG Elektronik Sertifika Hizmet
ECA Root CA
Echoworx Root CA2
Entrust Root Certification Auth
Entrust.net Certification Autho
Entrust.net Certification Autho
Entrust.net Secure Server Certi
ePKI Root Certification Authori
Equifax Secure Certificate Auth
Equifax Secure eBusiness CA-1
Equifax Secure eBusiness CA-2
Equifax Secure Global eBusines
Federal Common Policy CA
FNMT Clase 2 CA
GeoTrust Global CA
GeoTrust Primary Certificatio
Global Chambersign Root
GlobalSign
GlobalSign
GlobalSign Root CA
GlobalSign Root CA
Go Daddy Class 2 Certificatio
Go Daddy Root Certificate Au
GTE CyberTrust Global Root
Hongkong Post Root CA 1
http://www.valicert.com/
http://www.valicert.com/
http://www.valicert.com/
IPS CA Chained CAs Certificat
IPS CA CLASE1 Certification A
IPS CA CLASE3 Certification A
IPS CA CLASEA1 Certification
IPS CA CLASEA3 Certification
IPS CA Timestamping Certific
Izenpe.com
Izenpe.com
Juur-SK
KISA RootCA 1
KISA RootCA 3
KMD-CA Kvalificeret Person
KMD-CA Server
NetLock Arany (Class Gold) F
NetLock Expressz (Class C) T
NetLock Kozjegyzoi (Class A)
NetLock Minositett Kozjegyz
NetLock Uzleti (Class B) Tanu
Network Solutions Certificate
OISTE WISeKey Global Root G
Prefectural Association For JP
QuoVadis Root CA 2
QuoVadis Root CA 3
QuoVadis Root Certification Au
RSA Security 2048 V3
Secure Certificate Services
Secure Global CA
SecureSign RootCA11
SecureTrust CA
Security Communication EV Ro
Security Communication RootC
Security Communication RootC
Sonera Class1 CA
Sonera Class2 CA
Staat der Nederlanden Root CA
Staat der Nederlanden Root CA
Starfield Class 2 Certification A
Starfield Root Certificate Autho
Starfield Services Root Certifica
StartCom Certification Authorit
StartCom Certification Authorit
Swisscom Root CA 1
SwissSign CA (RSA IK May 6 19
SwissSign Gold CA – G2
SwissSign Platinum CA – G2
SwissSign Silver CA – G2
TC TrustCenter Class 2 CA II
TC TrustCenter Class 3 CA II
TC TrustCenter Class 4 CA II
TC TrustCenter Universal CA I
TC TrustCenter Universal CA II
TC TrustCenter Universal CA III
TDC Internet Root CA
TDC OCES CA
Thawte Personal Basic CA
Thawte Personal Basic CA
Thawte Personal Freemail CA
Thawte Personal Freemail CA
Thawte Personal Premium CA
Thawte Personal Basic CA
Thawte Personal Freemail CA
Thawte Personal Freemail CA
Thawte Personal Premium CA
Thawte Personal Premium CA
Thawte Premium Server CA
Thawte Premium Server CA
thawte Primary Root CA
thawte Primary Root CA – G2
Thawte Server CA
Thawte Server CA
Thawte Timestamping CA
Trusted Certificate Services
Trustis FPS Root CA
TÜBİTAK UEKAE Kök Sertifika Hizmet Sağlayıcısı – Sürüm 3
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı
TWCA Root Certification Authority
UCA Global Root
UCA Root
UTN – DATACorp SGC
UTN-USERFirst-Client Authentication and Email
UTN-USERFirst-Hardware
UTN-USERFirst-Network Applications
UTN-USERFirst-Object
VAS Latvijas Pasts SSI(RCA)
VeriSign Class 1 Public Primary Certification Authority – G3
VeriSign Class 2 Public Primary Certification Authority – G3
VeriSign Class 3 Public Primary Certification Authority – G3
VeriSign Class 3 Public Primary Certification Authority – G5
VeriSign Class 4 Public Primary Certification Authority – G3
Visa eCommerce Root
VRK Gov. Root CA
Wells Fargo Root Certificate Authority
WellsSecure Public Root Certificate Authority
XRamp Global Certification Authority

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

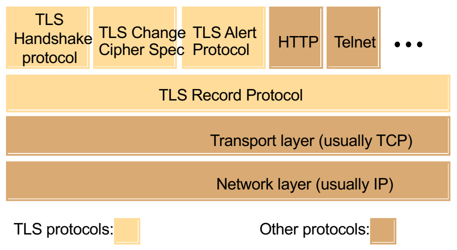


```
Certificate:
   Data:
       Version: 1 (0x0)
       Serial Number: 7829 (0x1e95)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       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)

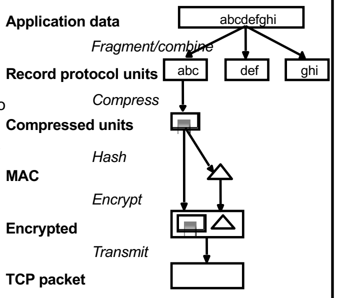
- SSL (Secure Socket Layer) was developed by Netscape for electronic transaction security.
- SSL was adopted as TLS as an Internet standard.
- A protocol layer is added below the application layer for:
  - Negotiating encryption and authentication methods.
  - Bootstrapping secure communication
- It 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

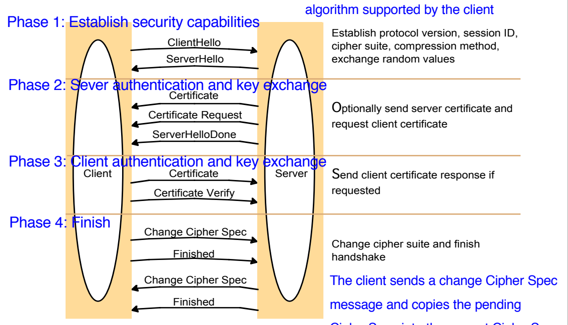


TLS Record Protocol

- 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


Cipher suite: a list of cryptographic algorithm supported by the client
Phase 1: Establish security capabilities
ClientHello
ServerHello
Establish protocol version, session ID, cipher suite, compression method, exchange random values
Phase 2: Sever authentication and key exchange
Certificate
Certificate Request
ServerHelloDone
Optionally send server certificate and request client certificate
Phase 3: Client authentication and key exchange
Client
Server
Certificate
Certificate Verify
Send client certificate response if requested
Phase 4: Finish
Change Cipher Spec
Finished
Change Cipher Spec
Finished
Change cipher suite and finish handshake
The client sends a change Cipher Spec message and copies the pending CipherSpec into the current CipherSpec.

CSE 486/586 Administrivia

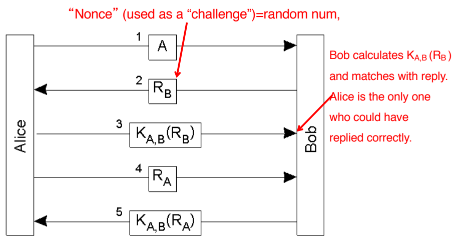
- Final: 5/18/2017, Thursday, 6 pm – 8 pm, Knox 110
- PA4 due on 5/12/2017 at 12 pm.

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

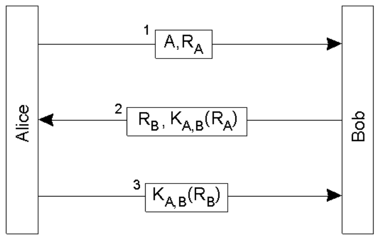
Direct Authentication

- Authentication with a secret key


"Nonce" (used as a "challenge")=random num,
Alice
Bob
1 A
2 R_B
3 $K_{A,B}(R_B)$
4 R_A
5 $K_{A,B}(R_A)$
Bob calculates $K_{A,B}(R_B)$ and matches with reply. Alice is the only one who could have replied correctly.

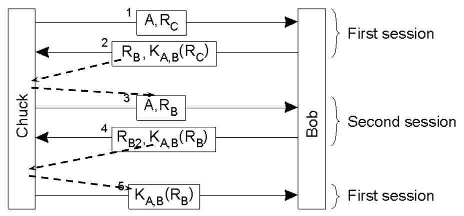
"Optimized" Direct Authentication

- Authentication with a secret key with three messages



- Anything wrong with this?

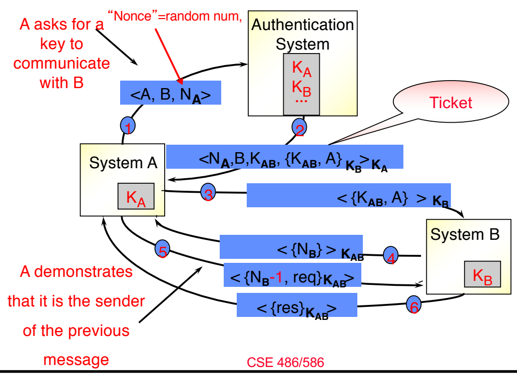
Reflection Attack



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

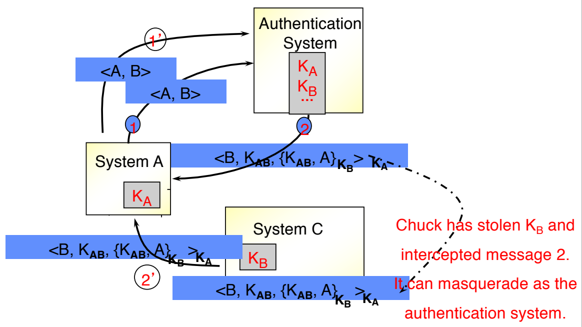
Needham-Schroeder Authentication



A asks for a key to communicate with B
"Nonce"=random num,
Authentication System
K_A
K_B
...
Ticket
<A, B, N_A>
1
2
System A
K_A
<N_A,B,K_{AB}, {K_{AB}, A} $_{K_B}$> $_{K_A}$
3
< {K_{AB}, A} > $_{K_B}$
System B
K_B
< {N_B} > $_{K_{AB}}$
4
5
< {N_B-1, req}$_{K_{AB}}$>
< {res}$_{K_{AB}}$>
6
A demonstrates that it is the sender of the previous message

Nonce N_A in Message 1

Because we need to relate message 2 to message 1



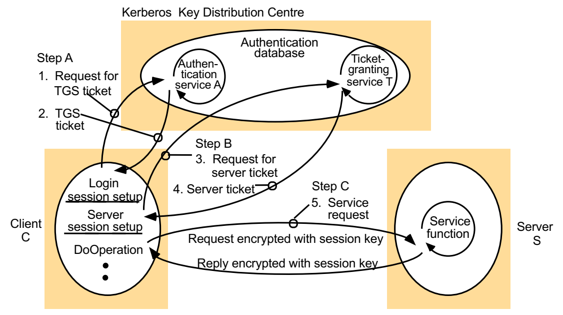
1'
<A, B>
<A, B>
Authentication System
K_A
K_B
...
1
2
System A
K_A
<B, K_{AB}, {K_{AB}, A}$_{K_B}$ > $_{K_A}$
System C
K_B
<B, K_{AB}, {K_{AB}, A}$_{K_B}$ > $_{K_A}$
2'
<B, K_{AB}, {K_{AB}, A}$_{K_B}$ > $_{K_A}$

Chuck has stolen K_B and intercepted message 2. It can masquerade as the authentication system.

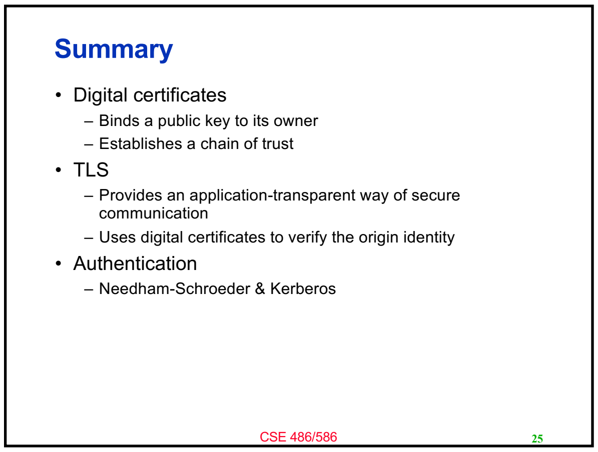
Kerberos

- 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

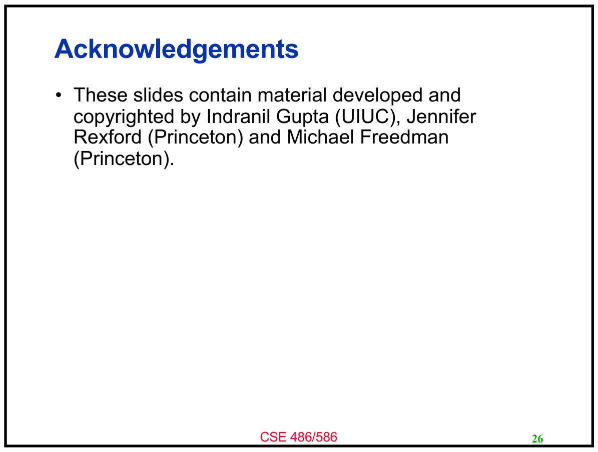


Kerberos Key Distribution Centre
Authentication database
Authen-tication service A
Ticket-granting service T
Step A
1. Request for TGS ticket
2. TGS ticket
Step B
3. Request for server ticket
4. Server ticket
Step C
5. Service request
Login session setup
Server session setup
DoOperation
Client C
Request encrypted with session key
Reply encrypted with session key
Service function
Server S



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).