Recap:

**Diffie-Hellman (El Gamal):** exponentiation is easy, discrete log is hard

Encryption – Bob picks a a large prime, p, and generator, g, and a private key <b>

Bob's public key < g^b mod p, g, p>

Bob's private key <b>

Alice wants to send a message to bob.

She picks <a> (Could be Alice's private key or a new number she picked randomly for sending message m)

Alice sends <g^a mod p, m * g^(ab) mod p>

Decryption – Bob takes Alices m* g^(ab) and divides it by g^(ab)

**El Gamal signature:**

Bob wants to sign m, generate Sm randomly

Sends m, g^(Sm) mod p, Sm + m*b mod (p-1)

y = Sm + m*b mod (p-1)

Verification: g^y = (g^sm)(g^b)^m

<m, signature>

Take message, attach signature, encrypt the whole thing

**RSA:**

Bob picks two large primes p & q and computes n = p*q

He also computes d,e such that d*e = 1 mod Ø(n)

Bob's public key <e, n>

Bob's private key <d>

Encryption: Alice wants to send a message to Bob.

She sends m^e mod n

Decryption: Bob decrypts by raising it to the d  (m^e)^d = M^(ed) = m mod n

(Theorem: For all a such that (a,n) = 1, a^Ø(n) = 1 mod n.)

Signature:

Bob signs by sending m, m^d mod n
Alice verifies by \((m^d)^e = m\)

**Public Key Cryptography & Infrastructure**

Certificate – Signed (public key + common name)

A certificate contains: <common name, public key, signature>

They are signed by a CA (Certificate Authority)

**Certificate Authorities:**
- “Trust anchors”
- entities whose public keys are embedded in browsers.
- VeriSign is trusted by Microsoft, and pays them money to have its key in browser

**Registration Authorities**
- Physically Verify Identity
- Once RA has verified identity, person presents public key, CA signs (public key + common name) and it and hands it back

**Weaknesses:**
- Certificates embedded in browser can be changed, so be wary of public computers
- You have to pay for certs from VeriSign
- Certs expire
- Single point of failure. If Verisign's private keys gets stolen, sites could be easily “signed” be Verisign.
- Physical identification process needs strengthening

**Certificate Chains:**
- You can obtain certificate issuing certificates from CA for extra $$
- Such certs have a bit set that allow the owner of the cert to sign the public key of other users in their domain.
- During verification the verifier starts at the public key of the trusted anchor and goes down the chain verifying signatures.

**Certificate Revocation:**
- Cancellation of a certificate
- No revocation today - no way for browser to check currently
- Online certificate status protocol (OCSP)
  - Requires bowser to check in real time with VeriSign when authorizing a certificate
  - Basically a black list check.
SSL/TLS (Secure Socket Layer/Transport Layer Security)
- SSL v2 – released by Netscape
- SSL v3 – released by Microsoft
- TLS – IETF committee (Internet Engineering Taskforce)
- SSL/TLS – https

<table>
<thead>
<tr>
<th>HTTP</th>
<th>SSL/TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP</td>
</tr>
<tr>
<td></td>
<td>IP</td>
</tr>
<tr>
<td></td>
<td>LINK</td>
</tr>
<tr>
<td></td>
<td>PHY</td>
</tr>
</tbody>
</table>

Application level security

Browser:
Picks session and cipher key

Website:
Has certificate and private key

--------------> hi <--------
<----- Cert, Suit of ciphers <----
>------Epub(session key) cipher ---->  
<------ talk using session key ------->