Transport Cryptography with TLS

Andrew Lytle
Washington State University

Outline

- Introduction, problem statement
- Background
  - Authentication
    - PKI
    - Certificates
    - Signatures
  - Encryption
    - Ciphers
- TLS
  - Cipher Suites
  - Handshake
  - Key exchange

Problem

We need a way to protect data being sent over public networks

We need to be sure that the data is authentic, and that no one else can read the data

We may want to authenticate one or both parties

Solution

Transport Layer Security

- Protocol that solves all our problems
- Provides a standardized way to securely connect two parties
- Facilitates the selection of authentication and encryption methods
- Provides both parties with a shared secret for encryption
- Layer 5 and 6 in OSI model

Authentication

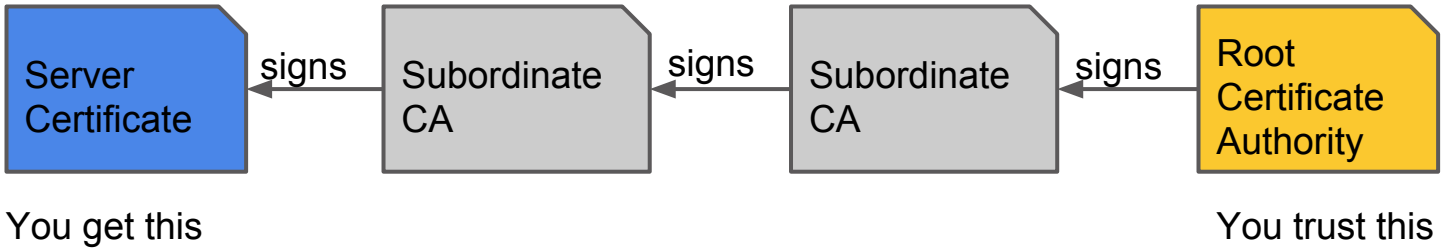
We need to ensure entities are who they say they are

We do this using a public key infrastructure (PKI) and X.509 certificates

Public Key Infrastructure

Each certificate is signed by a notary, whose certificate may be signed by another notary, and eventually we get to a trusted authority

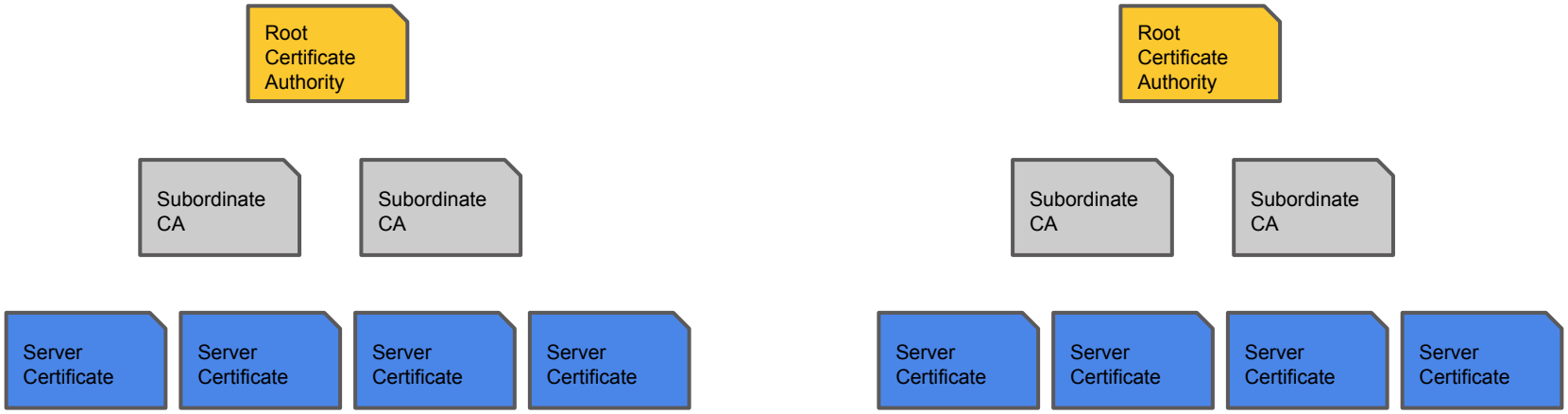
Trusted Certificate Authorities (CA) certificates are installed on web browsers and computers by default, or may be installed manually



PKI Cont.

This leads us to a hierarchical model with multiple CAs

CAs are responsible for verifying the identity of a certificate holder



X.509 Certificates

- Standard method for storing certificates
- Contains information about the subject
  - Name, Org Unit, Country, City, Domain names
  - Public key
  - Certificate validity period
- Contains information about the issuer
  - Name, Org Unit, Country, City
- Signature algorithms, use constraints

X.509 Certificate

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=US, O=Andrew Lytle, OU=Andrew Lytle Certificate Authority, CN=Andrew Lytle Root

Validity

Not Before: Jul 18 00:00:00 2013 GMT

Not After : Jul 17 23:59:59 2017 GMT

Subject: C=US, O=Andrew Lytle, OU=Andrew Lytle Certificate Authority, CN=*.andrewlytle.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:d9:35:21:e8:01:b2:d2:d9:cf:cc:c6:db:d1:07:

40:67:7a:b7:92:3d:44:0f:d7:a4:64:32:4c:49:aa:

…

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

X509v3 Extended Key Usage:

TLS Web Server Authentication

X509v3 Subject Alternative Name:

DNS:wsu.andrewlytle.com, andrewlytle.com

Signature Algorithm: sha1WithRSAEncryption

52:11:98:bc:14:32:9d:63:fa:ea:0c:fb:55:32:56:e3:9a:52:

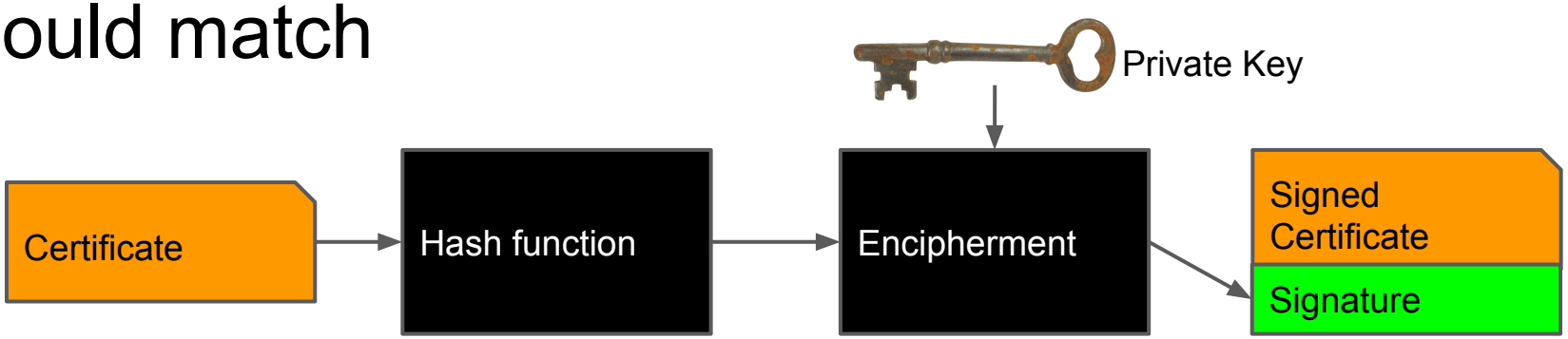
65:93:04:cd:13:36:d5:a0:40:85:90:95:4c:88:79:dd:41:61:

…

Digital Signatures

A signature is performed by **hashing** a certificate, then enciphering the result with a **private key**

To verify a signature, the signature is decrypted with the corresponding **public key**. The certificate is hashed locally and the results should match



Encryption

Data needs to be encrypted to protect from eavesdroppers

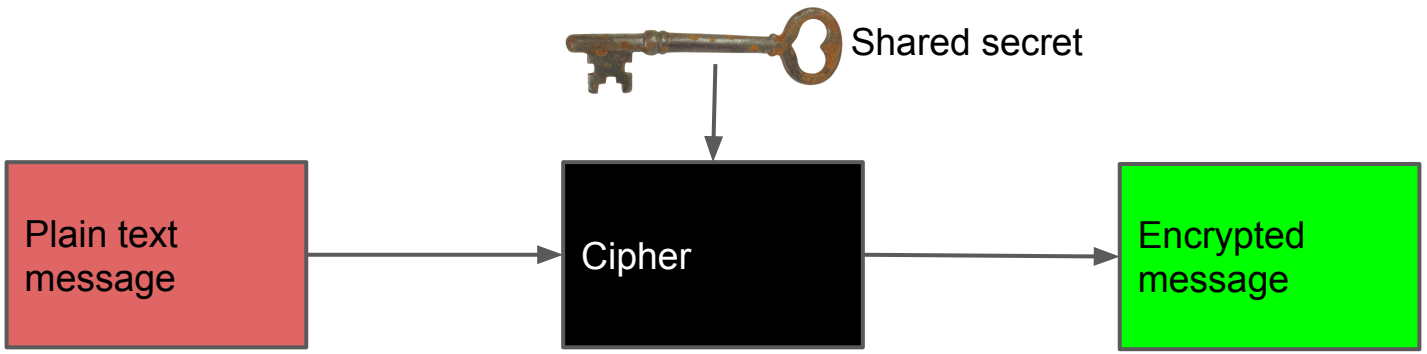
This is accomplished with encryption algorithms such as AES or RC4

These algorithms use a secret shared between the two parties to encrypt and decrypt data

Encryption cont.

Different types of encryption algorithms

- Block
  - Cipher block chaining
  - Galois/Counter mode
- Stream

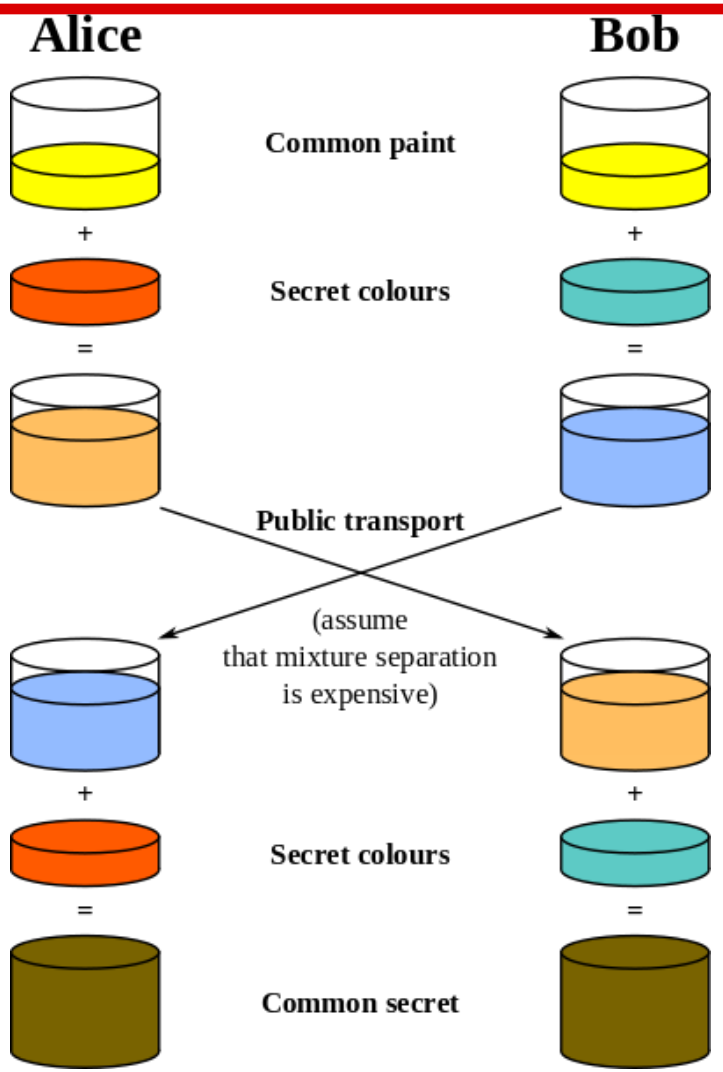


Key Exchange Methods

Symmetric key needs to be shared between both parties

Key exchange methods facilitate generating a shared secret

- Diffie-Hellman (shown)
- DHEC (similar)
- RSA



By Original schema: A.J. Han Vinck, University of Duisburg-Essen SVG version: Flugaal [Public domain], via Wikimedia Commons

TLS - Bringing it all together

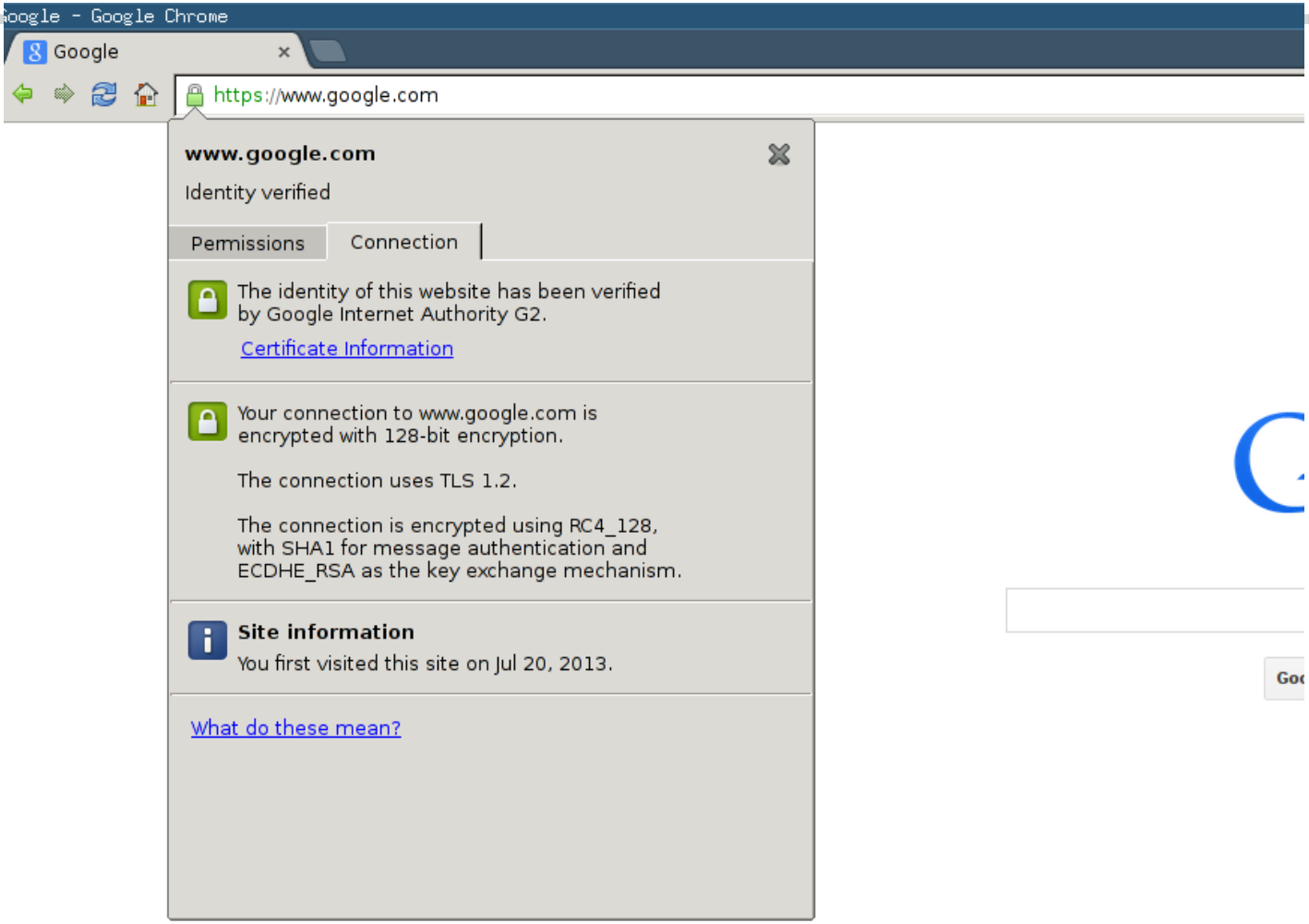
TLS combines all of these components into a single protocol

- PKI certificate based authentication
- Cipher suite selection
- Secure key exchange
- Message authentication

Cipher Suite

Protocol, Key Exchange, Message Auth, Encryption, Message Hash

TLS_ECDH_RSA_WITH_RC4_128_SHA



10.10.1.109	173.194.33.103	TLSv1.2	284	Client Hello
173.194.33.103	10.10.1.109	TLSv1.2	1434	Server Hello
173.194.33.103	10.10.1.109	TLSv1.2	1165	Certificate
10.10.1.109	173.194.33.103	TLSv1.2	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
173.194.33.103	10.10.1.109	TLSv1.2	308	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
173.194.33.103	10.10.1.109	TLSv1.2	119	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	119	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	107	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	127	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	462	Application Data

TLS Handshake in Detail

Client Hello

Client sends its information, including protocol version and enabled cipher suites

Server Hello

Server selects protocol version and cipher suite. Random numbers are generated based on key exchange protocol selected

Server Certificate

Server sends its certificate to the client. Client confirms certificate

Server Certificate Request (Optional)

Server demands client's certificate. Used to authenticate the client, if necissary

Client Certificate

Client Certificate Verification

Client verifies that it is in possession of the associated private key

(Client) Change Cipher Spec

Generation of shared secret is complete and both sides are authenticated. Client requests to finish handshake and begin session

(Server) Change Cipher Spec

Server confirms and begins session

Example

Client Hello

Server Hello

Server Certificate

Server Certificate Request (Optional)

Client Certificate

Client Certificate Verification

(Client) Change Cipher Spec

(Server) Change Cipher Spec

10.10.1.109	173.194.33.103	TLSv1.2	284	Client Hello
173.194.33.103	10.10.1.109	TLSv1.2	1434	Server Hello
173.194.33.103	10.10.1.109	TLSv1.2	1165	Certificate
10.10.1.109	173.194.33.103	TLSv1.2	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
173.194.33.103	10.10.1.109	TLSv1.2	308	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
173.194.33.103	10.10.1.109	TLSv1.2	119	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	119	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	107	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	127	Application Data
10.10.1.109	173.194.33.103	TLSv1.2	462	Application Data

Other Modes

- Client Authentication
- No encryption
- Anonymous
- Pre-shared key

4	0.009833000	127.0.0.1	127.0.0.1	TLSv1.2	223	Client Hello
6	0.019196000	127.0.0.1	127.0.0.1	TLSv1.2	2949	Server Hello, Certificate, Certificate Request, Server Hello Done
8	0.079339000	127.0.0.1	127.0.0.1	TLSv1.2	2952	Certificate, Client Key Exchange
10	0.143462000	127.0.0.1	127.0.0.1	TLSv1.2	335	Certificate Verify
12	0.183094000	127.0.0.1	127.0.0.1	TLSv1.2	125	Change Cipher Spec, Finished
14	0.202455000	127.0.0.1	127.0.0.1	TLSv1.2	72	Change Cipher Spec
16	0.203194000	127.0.0.1	127.0.0.1	TLSv1.2	119	Finished
18	235.3463510(	127.0.0.1	127.0.0.1	TLSv1.2	115	Application Data
20	235.3831290(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
22	235.3841410(	127.0.0.1	127.0.0.1	TLSv1.2	115	Application Data
24	235.3842110(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
26	295.7041840(	127.0.0.1	127.0.0.1	TLSv1.2	197	Application Data
28	295.7432740(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
30	295.7441910(	127.0.0.1	127.0.0.1	TLSv1.2	197	Application Data
32	295.7443850(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
34	371.4621970(	127.0.0.1	127.0.0.1	TLSv1.2	221	Application Data
36	371.4992000(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
38	371.5003340(	127.0.0.1	127.0.0.1	TLSv1.2	221	Application Data
40	371.5005290(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
42	415.8293440(	127.0.0.1	127.0.0.1	TLSv1.2	214	Application Data
44	415.8672070(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
46	415.8681460(	127.0.0.1	127.0.0.1	TLSv1.2	214	Application Data
48	415.8683410(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
50	493.2836730(	127.0.0.1	127.0.0.1	TLSv1.2	273	Application Data
52	493.3230940(	127.0.0.1	127.0.0.1	TLSv1.2	104	Application Data
54	493.3242530(	127.0.0.1	127.0.0.1	TLSv1.2	273	Application Data

```
  [Window size scaling factor: 1024]
▷ Checksum: 0xfeab [validation disabled]
  Urgent pointer: 0
▷ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
▷ [SEQ/ACK analysis]
▽ Secure Sockets Layer
▽ TLSv1.2 Record Layer: Application Data Protocol: Application Data
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 126
    Encrypted Application Data: 5468697320544c532073657373696f6e2069732072756e6e...
```

```
0040  49 84 17 03 03 00 7e 54  68 69 73 20 54 4c 53 20   I.....~T his TLS
0050  73 65 73 73 69 6f 6e 20  69 73 20 72 75 6e 6e 69   session  is runni
0060  6e 67 20 69 6e 20 75 6e  65 6e 63 72 79 70 74 65   ng in un encrypte
0070  64 20 6d 6f 64 65 2e 20  54 68 69 73 20 6d 65 61   d mode.  This mea
0080  6e 73 20 74 68 61 74 20  61 6c 6c 20 74 65 78 74   ns that  all text
0090  20 69 73 20 73 65 6e 74  20 69 6e 20 74 68 65 20    is sent  in the
00a0  63 6c 65 61 72 55 c8 df  3d e5 76 39 e0 4d b5 73   clearU.. =.v9.M.s
00b0  df a0 8c ca 16 de fb 67  56 77 15 10 18 57 85 45   .......g Vw...W.E
00c0  86 36 5c 69 43                                     .6\iC
```

Security

Known Attacks

- BEAST (Block padding)
- CRIME, BREACH (compression)
- RC4 statistical biases

Other

- Trust in CA
- Recent NSA leaks - Dual_EC_DRBG
- Implementation flaws

Questions?