Communication over Internet

1. Communicating Parties browser and server
2. Browser establishes a connection with a server (like icici bank) through url/ip address
3. Authentication: Server will send you (browser) its public key.
4. You can verify this public key is indeed the authentic public key through digital certificates (root CA, dig. Sign, verification process of digital certificates, public key of root CA)
5. Browser will generate a symmetric key and communicate it to server after encrypting it with server’s public key.
6. Server will decrypt it with its private key.
7. Secure connection is established
8. Communication happens through symmetric key encryption like AES/DES

PKI acts as the outer envelope, while core of cryptography is symmetric in all communications because of it being faster and thus feasible to use. Public key cryptography (with large key size of min. 1024 bits) is relatively much slower.

A(m)🡪 EB+ 🡪 EB+ (m)🡪 DB- 🡪 DB-(EB+ (m)) 🡪 m conf..

C Trudy

m: IDA, PwA, M, T, nonce, challenge-response

A(m)🡪 EB+ 🡪 EB+ (m)🡪 DB- 🡪 DB-(EB+ (m)) 🡪 m conf..

C Trudy

How can you avoid **replay** attack?

A: id(A), id (B), nonce, timestamp

1. Parties need to verify that they are authenticated in the real time.

To this end, we use:

Nonce, time-stamp, identity of A, identity of B, challenge-response

One way Authentication

A: dig\_signA (m, id(A), id (B), nonce RA, timestamp)

Two Authentication

A: dig\_signA (id(A), id (B), nonce RA, timestampA)

B: dig\_signB (id(B), id (A), RA, timestampB, RB)

Three way authentication

A: dig\_signA (id(A), id (B), nonce RA, timestampA)

B: dig\_signB (id(B), id (A), RA, timestampB, RB)

A: dig\_signA (id(A), id (B), nonce RA, timestampA)

B: dig\_signB (id(B), id (A), RA, timestampB, RB)

A: dig\_signA (id(A), id (B), nonce RB,

timestampA)