
Digital Signatures, Key Exchange

4 May 2011
Lecture 9

Slide Credits: Steve Zdancewic (UPenn)

Topics for Today

- Digital Signatures
- Key Distribution
 - Needham-Schroeder Protocol

Physical Signatures

- Consider a paper check used to transfer money from one person to another
- Signature confirms authenticity
 - Only legitimate signer can produce signature
- In case of alleged forgery
 - 3rd part can verify authenticity
- Checks are cancelled
 - So they can't be reused
- Checks are not alterable
 - Or alterations are easily detectable

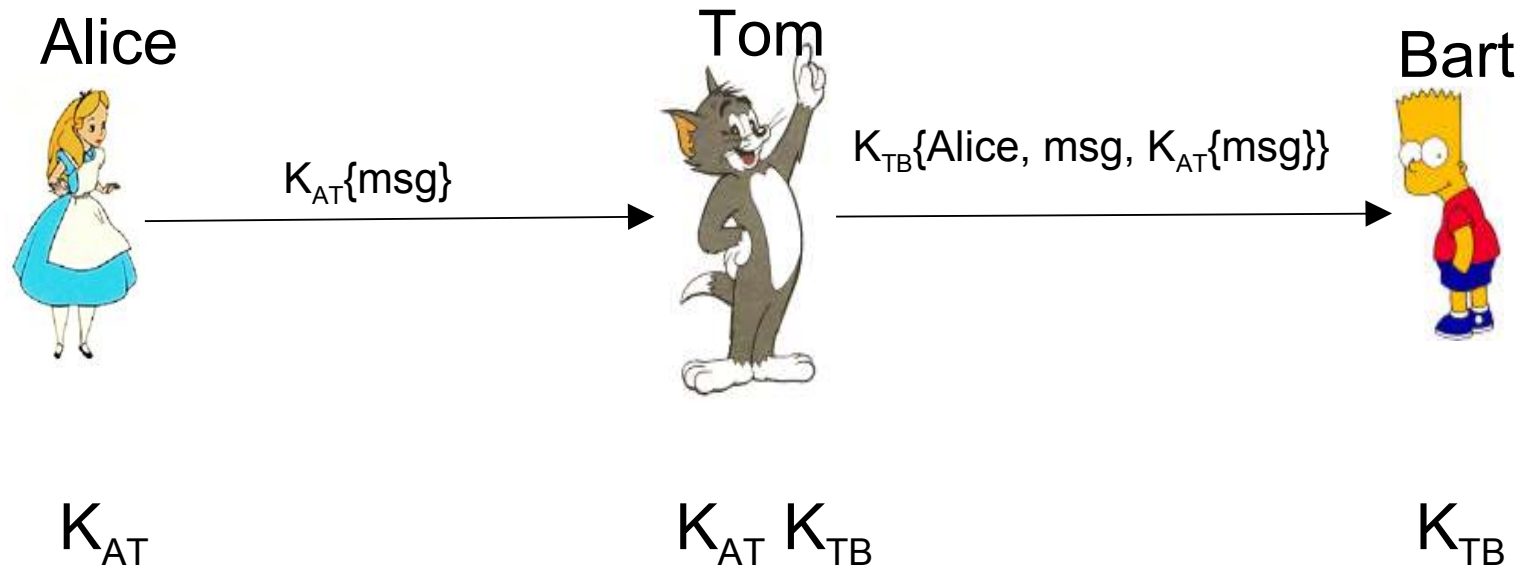
Digital Signatures: Requirements I

- A mark that only one principal can make, but others can easily recognize
- Unforgeable
 - If P signs a message M with signature $S_p\{M\}$ it is impossible for any other principal to produce the pair $(M, S_p\{M\})$.
- Authentic
 - If R receives the pair $(M, S_p\{M\})$ purportedly from P, R can check that the signature is really from P.

Digital Signatures: Requirements II

- Not alterable
 - After being transmitted, $(M, S_P\{M\})$ cannot be changed by P, R, or an interceptor
- Not reusable
 - A duplicate message will be detected by the recipient.
- Nonrepudiation
 - P should not be able to claim they didn't sign something when in fact they did
 - (Related to unforgeability: If P can show that someone else could have forged P's signature, they can repudiate ("refuse to acknowledge") the validity of the signature.)

Digital Signatures with Shared Keys



- Tom is a trusted 3rd part (or aribter)
- **Authenticity:** Tom verifies Alice's message, Bart trusts Tom
- **No Forgery:** Bart can keep $msg, K_{AT}\{msg\}$ which only Alice (or Tom, but he's trusted not to) could produce

Preventing Reuse and Alteration

- To prevent reuse of the signature
 - Incorporate a *timestamp* (or sequence number)
- Alteration
 - If a block cipher is used, recipient could splice-together new messages from individual blocks
- To prevent alteration
 - Timestamp must be part of each block
 - Or ... use *cipher block chaining*

Digital Signatures with Public Keys

- Assumes the algorithm is commutative
 - $D(E(M,K),k) = E(D(M,k),K)$
- Let K_A be Alice's public key
- Let k_A be her private key
- To sign msg, Alice sends $D(msg, k_A)$
- Bart can verify the message with Alice's public key

- Works! RSA: $(m^e)^d = m^{ed} = (m^d)^e$

Digital Signatures with Public Keys

Alice



$k_A\{msg\}$

Bart



k_A, K_A, K_B

k_B, K_B, K_A

- No trusted 3rd party
- Simpler algorithm
- More expensive
- No confidentiality

Variations on Public Key Signatures

- Timestamps again (to prevent replay)
 - Signed certificates valid for only some time
- Add an extra layer of encryption to guarantee confidentiality
 - Alice sends $K_B\{k_A\{msg\}\}$ to Bart
- Combined with hashes
 - Send $(msg, k_A\{MD5(msg)\})$

Thinking about Digital Signatures

- We have seen two uses of encryption so far:
 - Secrecy (encrypt/decrypt)
 - Authentication (digital signatures)
- The two have very different requirements
 - Strength of cipher
 - Lifetime
 - Storage
- It's normal to have separate encryption and signing key pairs
 - Why?
- What risks are associated with digital signatures that are not present in secret communication?
 - The other way around?

So Far

- Digital Signatures
- Key Distribution
 - Needham-Schroeder Protocol

Key Establishment

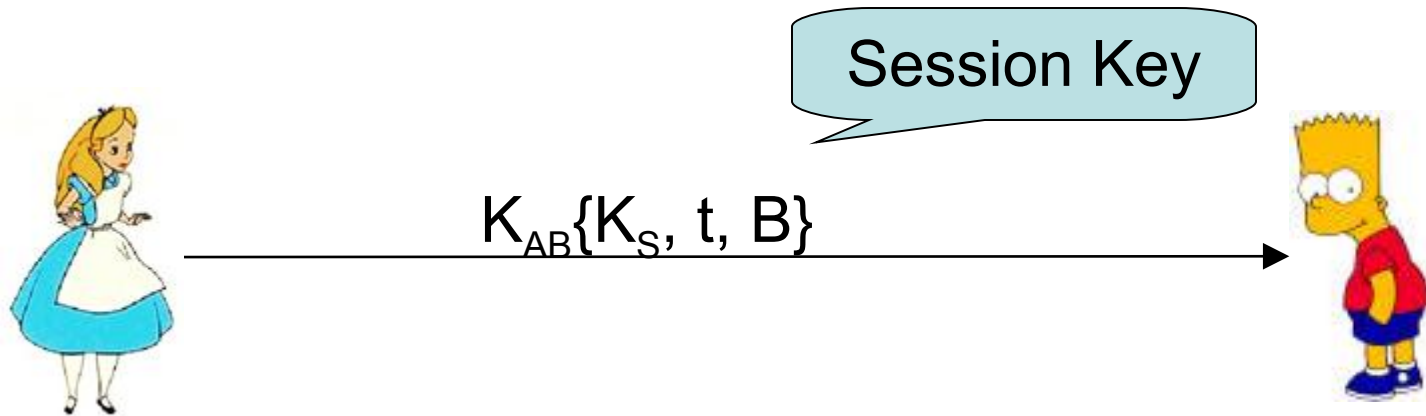
- Establishing a “session key”
 - A shared key used for encrypting communications for a short duration – a session
 - Need to authenticate first

- Symmetric key
 - Point-to-Point
 - Needham-Schroeder
 - Kerberos

Symmetric Keys

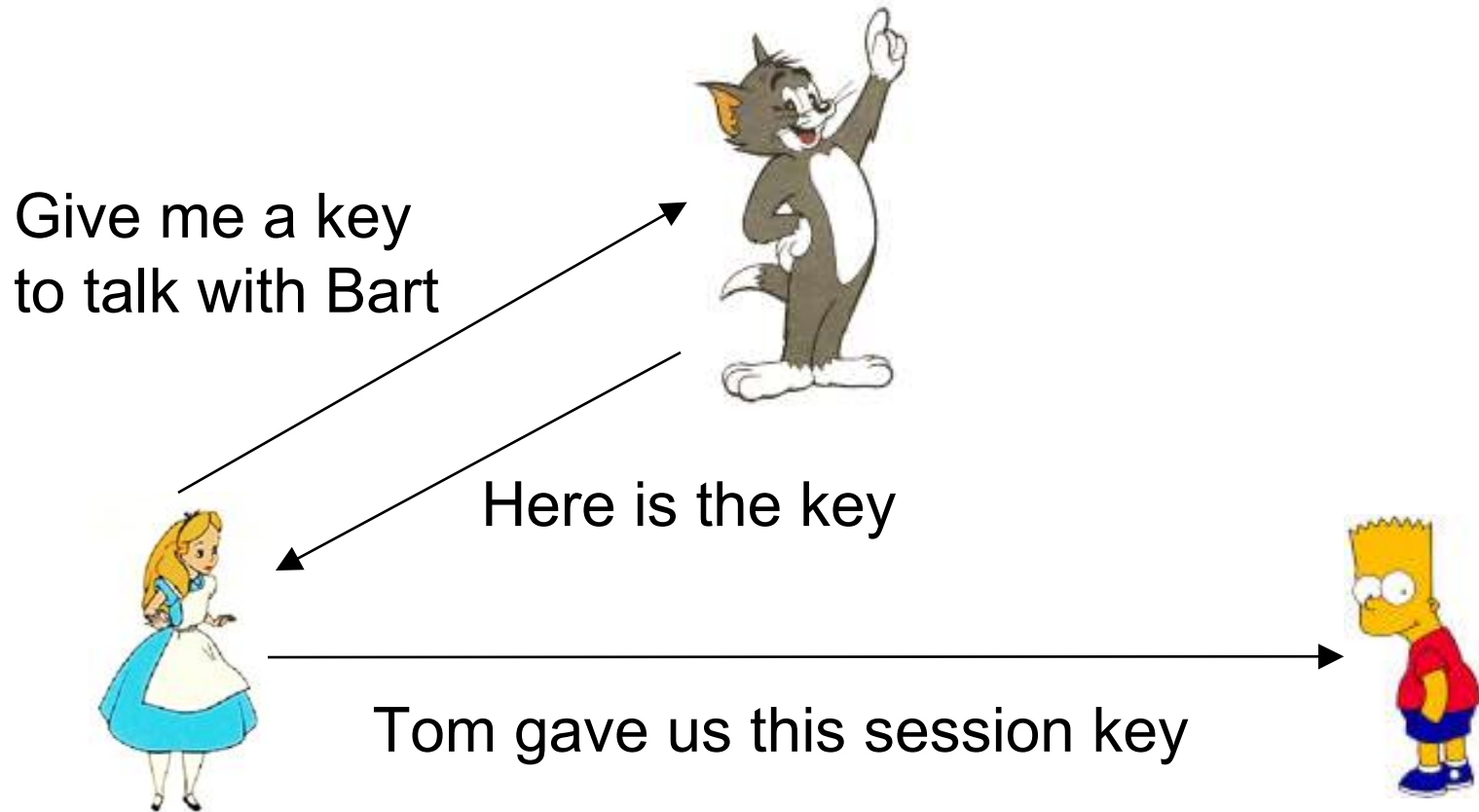
- Key establishment using only symmetric keys requires use of pre-distribution keys to get things going
- Then protocol can be based on
 - Point to point distribution, or
 - Key Distribution Center (KDC)

Point-to-Point



- Should also use timestamps and nonce.
- Session key should include a validity period
- Could also use public key cryptography to
 - Authenticate
 - Exchange symmetric shared key

Key Distribution Centers



Distribution Center Setup

- A wishes to communicate with B
- T (trusted 3rd party) provides session keys
- T has a key K_{AT} in common with A and a key K_{BT} in common with B
- A authenticates T using a nonce n_A and obtains a session key from T
- A authenticates to B and transports the session key securely

Needham-Schroeder Protocol

1. $A \rightarrow T$: A, B, n_A

2. $T \rightarrow A$: $K_{AT}\{K_S, n_A, B, K_{BT}\{K_S, A\}\}$

A decrypts K_{AT} and checks n_A and B. Holds K_S for future correspondence with B.

1. $A \rightarrow B$: $K_{BT}\{K_S, A\}$

B decrypts with K_{BT}

1. $B \rightarrow A$: $K_S\{n_B\}$

A decrypts with K_S

1. $A \rightarrow B$: $K_S\{n_B-1\}$

B checks n_B-1

Conclusion

- Digital Signatures
- Key Distribution
 - Needham-Schroeder Protocol