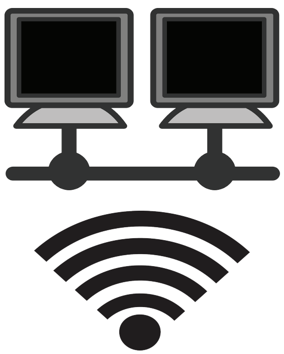
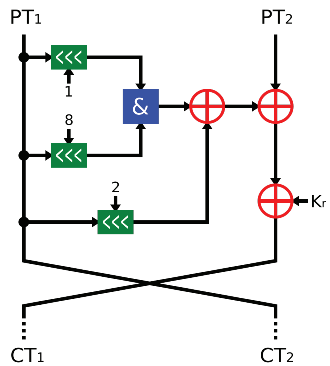
CSE 40567 / 60567: Computer Security





Cryptography 2

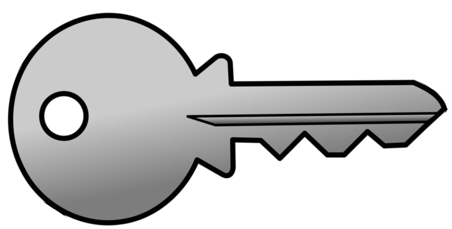
Homework #1 is due **tonight** at 11:59PM

See **Assignments Page** on the course website for details

Key management strategies

Thus far, we've discussed authenticating actors, but have assumed keys were already in-place for the protocols

How can we use authentication protocols to help us exchange keys securely?



Basic Key Management

- Let's assume Carol is a trusted third party
- Carol distributes certificates upon request

- A certificate is an electronic document that conveys a key and related meta-data
- Guaranteed by Carol

Basic Key Exchange Protocol

Timestamp

1. $A \longrightarrow C: A, B$
2. $C \longrightarrow A: \{A, B, K_{AB}, T\}_{K_{AC}}, \{A, B, K_{AB}, T\}_{K_{BC}}$
3. $A \longrightarrow B: \{A, B, K_{AB}, T\}_{K_{BC}}, \{X\}_{K_{AB}}$

Needham-Schroeder Protocol

Like the basic key exchange protocol, but with nonces instead of timestamps:

1. $A \longrightarrow C\text{: } A, B, N_A$
2. $C \longrightarrow A\text{: } \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BC}}\}_{K_{AC}}$
3. $A \longrightarrow B\text{: } \{K_{AB}, A\}_{K_{BC}}$
4. $B \longrightarrow A\text{: } \{N_B\}_{K_{AB}}$
5. $A \longrightarrow B\text{: } \{N_B - 1\}_{K_{AB}}$

Bob checks if Alice is alert

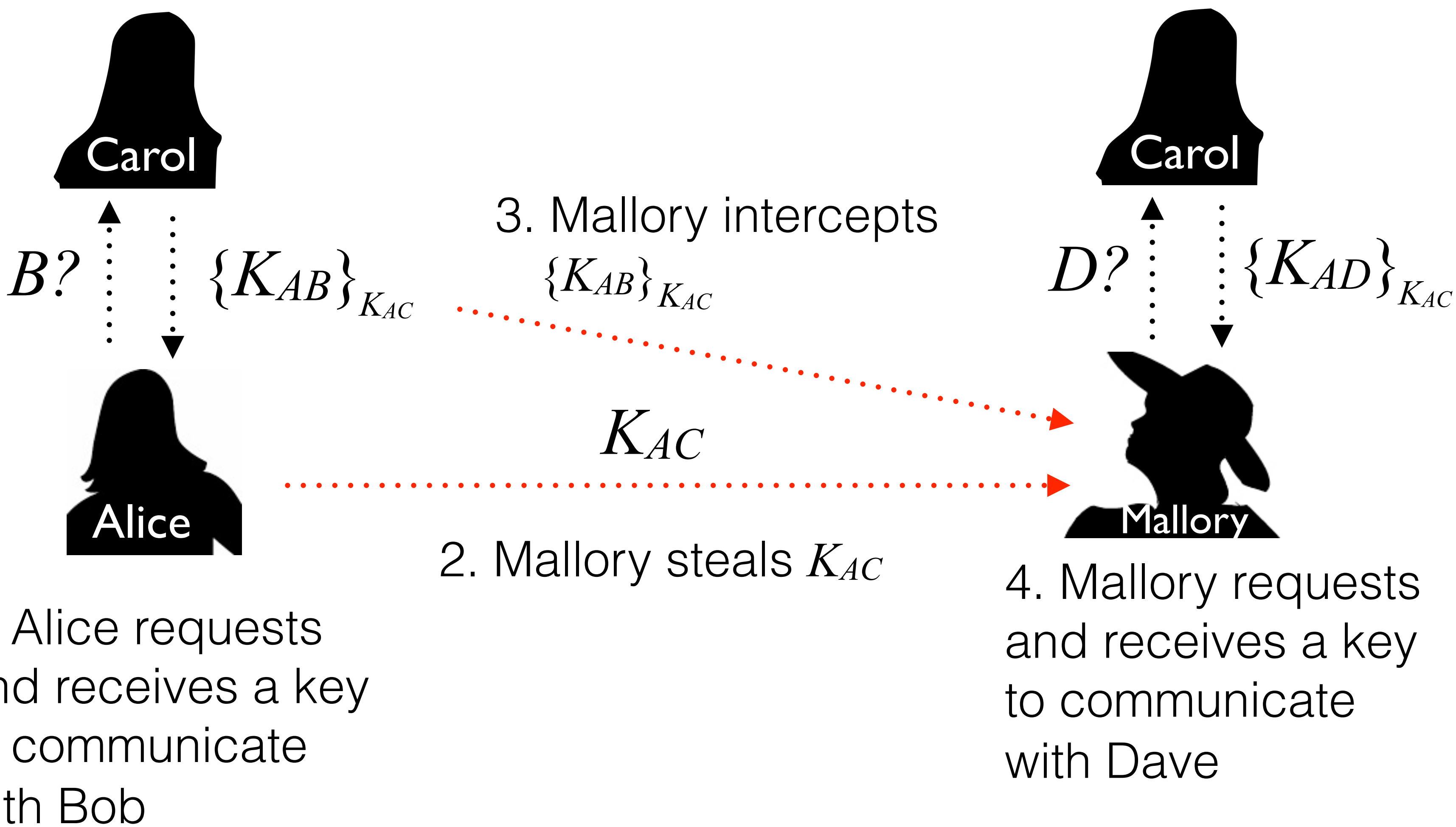
E. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers," Communications of the ACM 21 (12): 993–999, 1978

Flaw in Needham-Schroeder Protocol

Bob has to assume K_{AB} from Carol is fresh

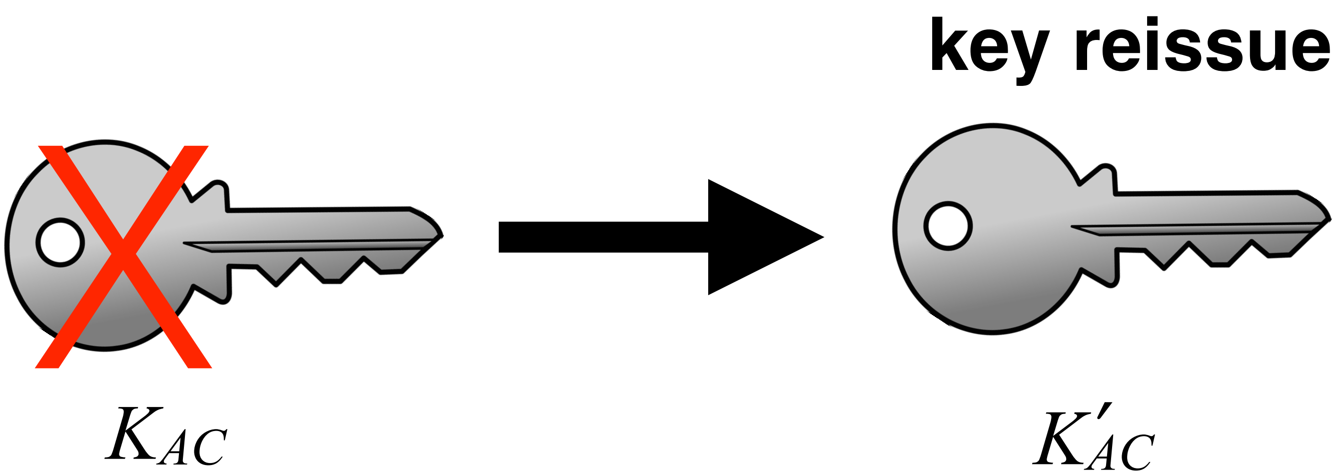
- K_{AB} is always conveyed by Alice
- What if Alice waited a year between steps 2 and 3?
  - Mallory can use K_{AB} to establish a session with Bob
  - If K_{AB} is compromised, Bob can't easily detect a change made by Carol

Flaw in Needham-Schroeder Protocol



Alice's response to compromise

- Assume Alice finds out about the stolen key by comparing message logs with Bob
- Alice initiates **key revocation**



Trouble with key revocation in Needham-Schroeder protocol

- Alice can't handle the key revocation by herself
  - She has no idea that Mallory has her key for communication with Dave
- Carol must handle key revocation and reissue
  - She needs to keep an exhaustive log for *every* key request

Alice requested Bob's key, Alice requested Dave's key, Bob requested Alice's key, Bob requested Dave's key, Dave requested Bob's key…

Fundamental Problem: Assumptions

- Anderson: “1978 was a a kinder, gentler world”
    - Computer security in that era focused on keeping “bad guys” out
    - Now we expect users to be adversaries
- Needham-Schroeder works if all of the actors behave themselves, and attacks only come from the outside

Kerberos

Two trusted third-parties:

1. Authentication server, which users log into
2. Ticket granting server, which gives users tickets needed to access resources

"And before them a dreaded hound, on watch, who has no pity, but a vile stratagem."

J. Steiner, C. Neuman, and J. Schiller, "Kerberos: An authentication service for open network systems," USENIX Winter Conference, 1988

Kerberos protocol

Authentication server: Carol

Ticket granting server: Dave

Alice's password: P

Alice needs access to a resource provided by Bob:

session key

lifetime

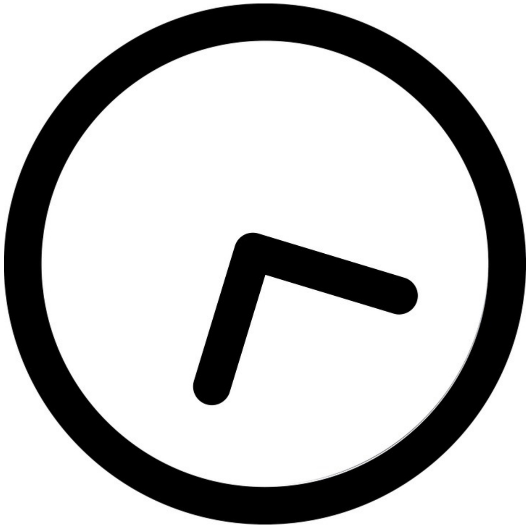
transaction key

1. $A \longrightarrow C$: P
2. $C \longrightarrow A$: $\{K_{AS}\}_P$
3. $A \longrightarrow D$: A, B
4. $D \longrightarrow A$: $\{T_D, L, K_{AB}, B\{T_D, L, K_{AB}, A\}_{K_B}\}_{K_{AS}}$
5. $A \longrightarrow B$: $\{T_D, L, K_{AB}, A\}_{K_B}$, $\{A, T_A\}_{K_{AB}}$
6. $B \longrightarrow A$: $\{T_A + 1\}_{K_{AB}}$

Bob's key (known by Bob and Dave)

What does this fix in Needham-Schroeder?

- Timestamps are used in place of nonces
  - Revoked / expired keys are easily detected
  - New source of trouble: out of synch clocks



Race conditions

Where is Kerberos used?

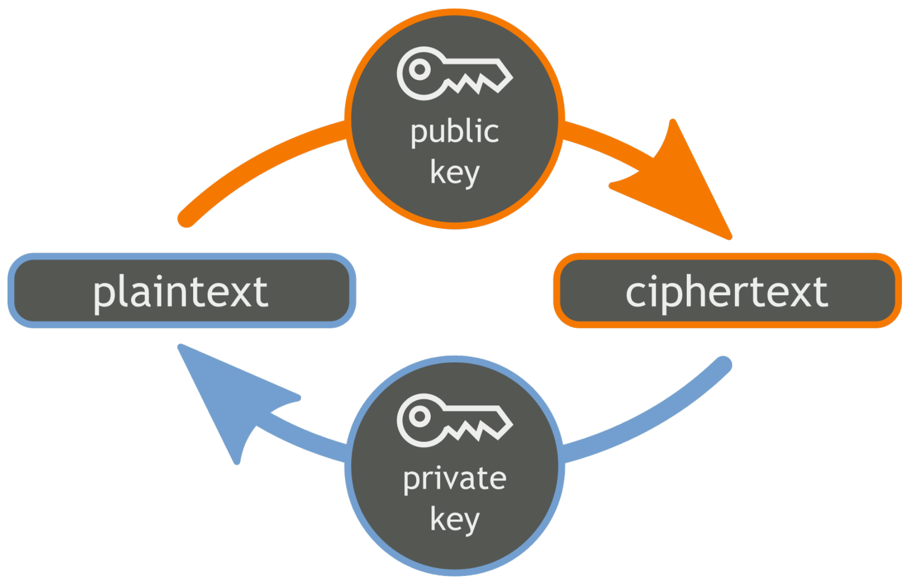
Kerberos is the default authentication mechanism in Microsoft Windows

1. Account is created on the Domain Controller and given password
2. Kerberos client creates shared secret
3. User enters username and password, Kerberos client generates secret key *on the client*
4. User and Authentication Service running on the Domain Controller communicate using shared secret

https://redmondmag.com/articles/2012/02/01/understanding-the-essentials-of-the-kerberos-protocol.aspx

Practical considerations for key management

- Passing around symmetric keys is messy
- **Public-key Cryptography** helps us somewhat
  - Public Key Infrastructure
- We'll have a lot more to say about this…



BAN (Burrows–Abadi–Needham) Logic

$A \mid\equiv X$	Alice believes X
$A \mid\sim X$	Alice once said X
$A \mid\Rightarrow X$	Alice has jurisdiction over X
$A \mid\triangleleft X$	Alice sees X
$\#X$	X is fresh
$\{X\}_k$	X is encrypted under the key k
$A \overset{k}{\leftrightarrow} B$	A and B share the key k

Message meaning rule

If Alice sees a message encrypted under k, and k is a good key for communicating with Bob, then she will believe that the message was once said by Bob.

$$\frac{A \mid\equiv A \stackrel{k}{\leftrightarrow} B,\ A \mid\!\lhd \{X\}_k}{A \mid\equiv B \mid\!\sim X}$$

Nonce-verification rule

If Bob once said a message, and the message is fresh, then Alice believes it.

$$\frac{A \mid\equiv \#X,\ A \mid\equiv B \mid\sim X}{A \mid\equiv B \mid\equiv X}$$

Jurisdiction rule

If Bob believes something, and is an authority on the matter, then Alice should believe him.

$$\frac{A \mid\equiv B \mid\Rightarrow X,\ A \mid\equiv B \mid\equiv X}{A \mid\equiv X}$$

Smartcard banking protocol

Transaction takes place between Alice's smart card and a vending machine owned by Bob, which contains his smart card

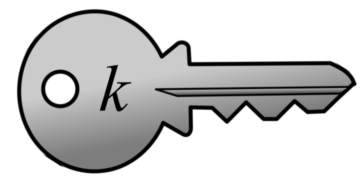
1. $A \longrightarrow B$: $\{A, N_A\}_k$
2. $B \longrightarrow A$: $\{B, N_B, A, N_A\}_k$
3. $A \longrightarrow B$: $\{A, N_A, B, N_B, X\}_k$

electronic check

Chip-enabled Bank of America BankAmericard Visa Signature Credit Card
BY-SA 2.0 tales of a wandering youkai

Verification of smartcard banking protocol

Assumption: k is only available to actors who can be trusted to execute the protocol faithfully



Goal: Prove that Bob should trust the check

$$\boxed{B \mid\equiv X}$$

*Reasoning proceeds backwards

Verification of smartcard banking protocol

1. $B \mid\equiv X$ follows from $B \mid\equiv A \mid\Rightarrow X$ $\longleftarrow$ hardware constraint

and

$B \mid\equiv A \mid\equiv X$

2. $B \mid\equiv A \mid\equiv X$ follows from $\#X$ and

and

$B \mid\equiv A \mid\sim X$

Verification of smartcard banking protocol

3. $\#X$ follows from its occurrence in $\{A, N_A, B, N_B, X\}_k$

Guaranteed by this sequence number

4. $B \mid\equiv A \mid\sim X$ follows from the hardware constraint

Q.E.D.

Limits of formal verification

- Bad Assumption: What if Mallory stole k?

- Smartcard hardware is not sufficient to guarantee security

- Implementation flaw: what if k is actually two keys — a transaction key and an undiversified bank key?

These aren't flaws of the formal method, but practical constraints