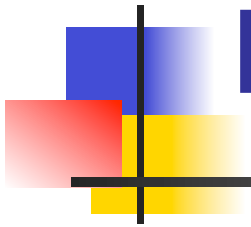
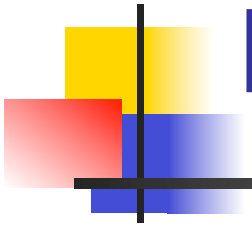


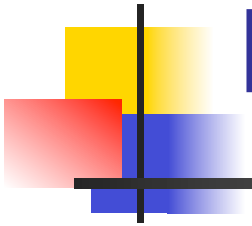
Protocols

Prof. Sloan’s Slides



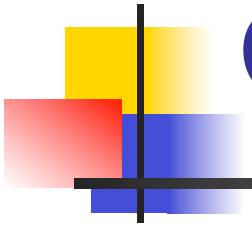
Protocols

- Passwords can be considered as part of a (simple) protocol.
- But fancier things, or both principals devices, definitely require protocol
  - E.g., Key fob–car; IFF system



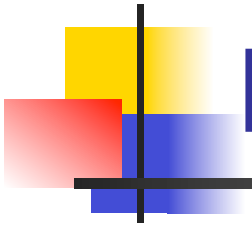
Protocols

- A set of rules for how ≥2 principals do something, typically over public communication channel
  - E.g., authenticate one to another; mutually authenticate; vote so all agree on outcome but votes are secret; commit to a value
- Must of course be specified precisely
- Often very delicate; can break if explicit/ implicit assumptions don't hold, or protocol is flat-out breakable.



Common Protocol Ingredients

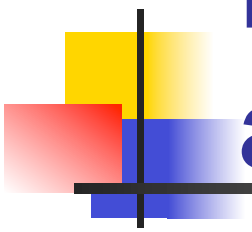
- Two parties can have secure communication by using cryptography with shared key
  - But must have pre-established key, key distribution, or public-key crypto
- ***Nonce*** "number used once"—can generate arbitrary random number
- Can generate very crudely synched timestamps



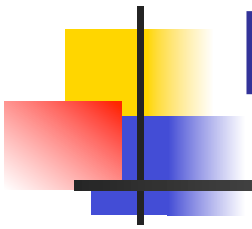
Password Protocol

- General notation: Alice (A) and Bob (B) share a secret K_{ab}.
- **Password protocol:**
  - **B→A: K_{ab}**
- Notation: Lines have two parts (split by colon): 1st specifies principals sending and receiving; second part gives the message sent.

Example: Simple Challenge and response



- Car engine *E* authenticating smart key fob transponder *T* once key is inserted into ignition
- Two steps:
  1. *E* sends *T* a nonce *N*
  2. *T* sends back *(T, N)* encrypted with their shared key



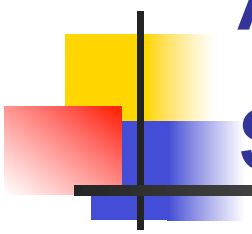
Protocol Notation

- Putting things in brackets with a key subscript means encrypted with that key:
  - E.g., $T \rightarrow E : \{T, N\}_{K_{ET}}$ *means* "T sends to E T & N encrypted with E and T's shared key".
- Simple Challenge-response becomes:

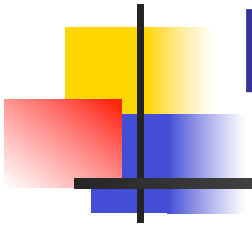
$E \rightarrow T : N$

$T \rightarrow E : \{T, N\}_{K_{ET}}$

Assumption needed for security



- Nonce must be *unpredictable* pseudorandom number; not just fresh number never used before, such as the date, or next in sequence 1,2,3,....
- Otherwise, car thief can figure out what next challenge to key fob will be, and ask the key fob himself as owner walks away from the car.
  - This would work even if fob was checking the newness of the nonce! (Unlikely)



Man-in-the middle attacks

- Say *E* allowed fob transponder *T* to transmit request *without* being inserted by sending *"Please"*
  - Crook sends *"Please"* to *E*, gets back challenge *N*, sends *N* to *T; T* sends proper response to crook thinking crook is *E;* crook gives this response to *E.*
  - Perhaps unreasonable for ignition key, but how about garage-door remote?
- Many protocols can be broken this way.

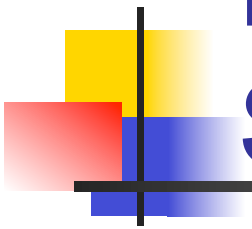
Mutual Challenge Response

$$A \rightarrow B : N_a$$

$$B \rightarrow A : \{N_a, N_b\}_{K_{ab}}$$

$$A \rightarrow B : N_b$$

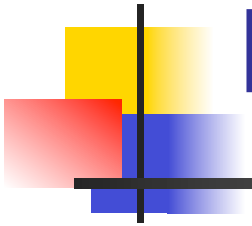
Famous Protocol: Needham-Schroeder



- Key distribution protocol from the late 1970s.
- Parties are arbitrary pool of principals and trusted key server S. Allows any one principal A to request S to give a new session key for use by A and B.
- I.e., starts by A telling S that she wants a new session key to communicate with B.
- Each principal has unique shared key with S; denote shared key of A and S by K_{AS}

Needham-Schroeder Protocol

$$
\begin{aligned}
A \rightarrow S: &\quad A, B, N_A \\
S \rightarrow A: &\quad \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \\
A \rightarrow B: &\quad \{K_{AB}, A\}_{K_{BS}} \\
B \rightarrow A: &\quad \{N_B\}_{K_{AB}} \\
A \rightarrow B: &\quad \{N_B - 1\}_{K_{AB}}
\end{aligned}
$$



Problem with N–S

- Anybody who steals Alice's key with Sam (K_{AS}) can impersonate Alice to 3rd parties!
- Is this okay?
- Probably not today, but really it's all about what assumptions you make.
- (Using timestamp for nonce would fix this problem.)