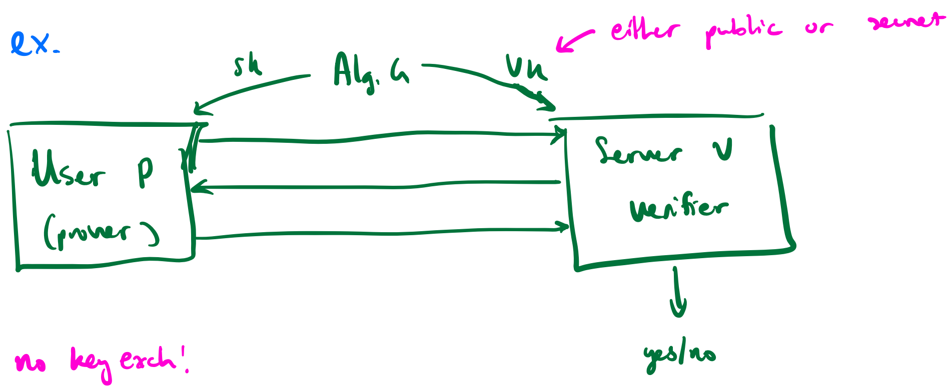
ID protocols

ex.



no key exch!

Applications

Physical locks: Friend-or-foe
- → Wireless car entry
- → Opening office door

Login at bank ATM or desktop computer

Internet: login to remote site after key exch w/ one-sided auth (e.g. HTTPS)

Security models

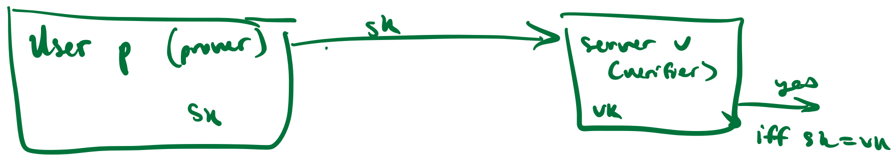
1. Direct attacker: impersonates prover w/ no info other than vk (e.g. door lock)
2. Eavesdropping attacker: impersonates prover after eavesdropping on a few convos between prover and verifier (e.g. wireless car entry)
3. Active attacker: interrogates prover and attempts to impersonate prover (e.g. fake ATM)

Password Protocols

PWD: finite set of passwords

Alg. G (keygen):

choose $pw \leftarrow PWD$, output $sk = vk = pw$



Problem: vk must be kept secret.

Soln: store Hash of pwd's

(only works against strong pwd's, dictionary attack can break weak passwords)

Dict of 360M words covers 25% of all pwd's

Offline / batched offline dict attacks can easily break LOTS of passwords (e.g. 2012 LinkedIn hack)

Soln: different salt (public) for each user, salt pw + attack, prevents fast batch attacks

and use slow hash fns

(PBKDF2, bcrypt)

Problem: custom ASIC hardware can evaluate hash fns 50,000x faster than CPU

Soln: Make it need a lot of memory (e.g. scrypt)

Eavesdropping Security Model

Adv has:

- → Server vk
- → transcript of interactions btwn honest prover, verifier

Goal: impersonate prover to verifier

Protocol is secure if no eff. adv can win this game.

Pwd protocol is insecure.

One-time passwords (e.g. 2-factor auth)

Setup (alg G)

- → choose rand. key k
- → output $sk = (k, 0)$, $vk = (k, 0)$

Ident:

prover $sk = (k, 0)$

$r_0 \leftarrow F(k, 0)$

$r_1 \leftarrow F(k, 1)$

server $vk = (k, 0)$

yes iff. $r = F(k, 0)$

inc ctr per auth or after time period