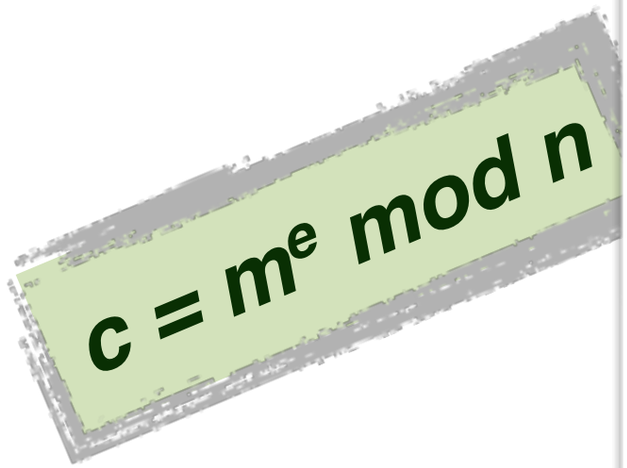
CS6431: Security and Privacy Technologies



Authentication Tokens (A Survey)

Instructors: Ari Juels, Tom Ristenpart, Vitaly Shmatikov
Fall 2016
8 September 2016, Lecture by Ari Juels

CORNELL UNIVERSITY FOUNDED A.D. 1865
CORNELL NYC TECH

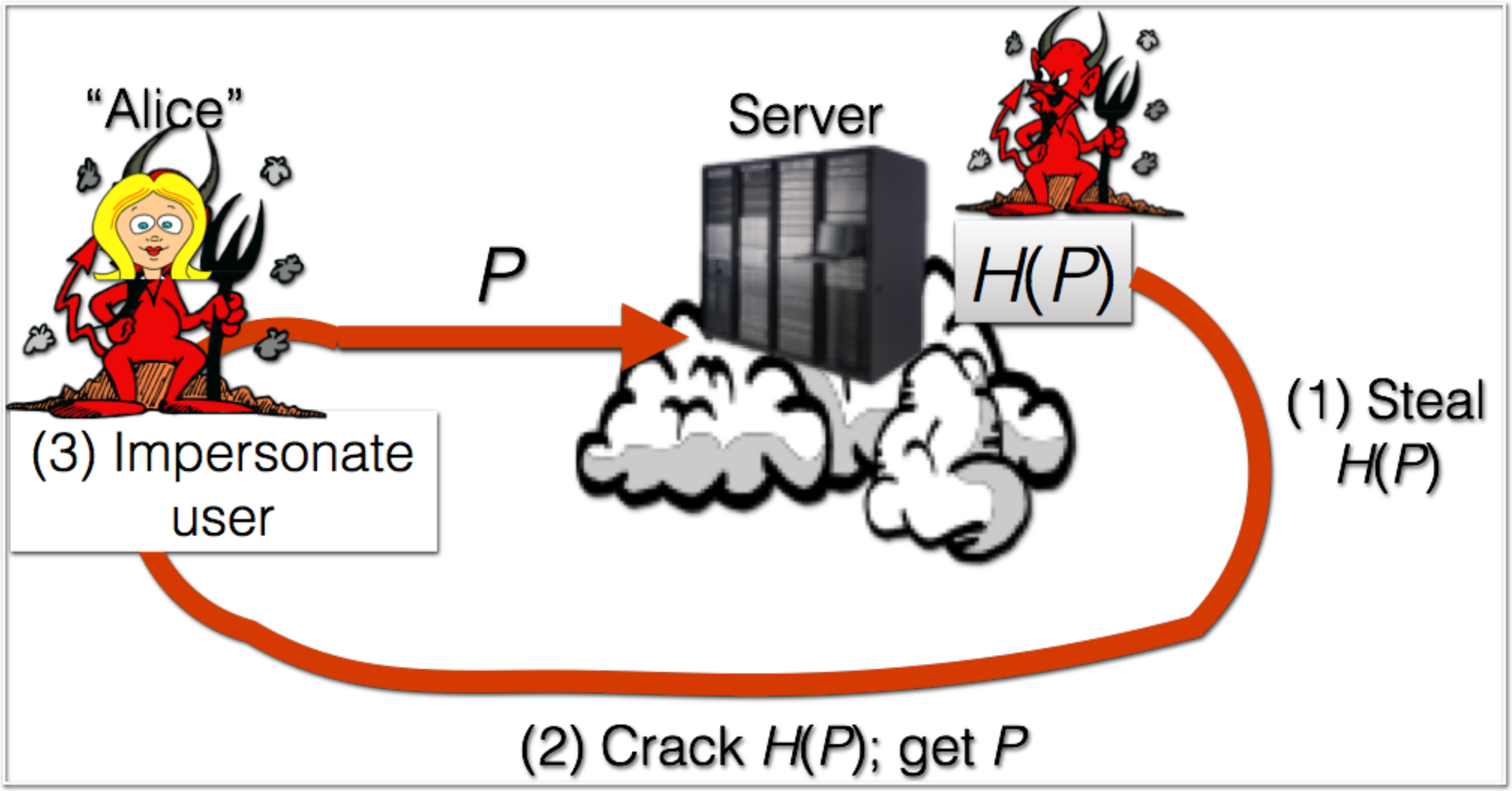
The three classical authentication factors

1. Something you know
   - E.g., password, password-recovery question
2. Something you are
   - Biometric
3. **Something *you have***
   - **Authentication token**

- But there are also…
  - Someone you know
    - "Social authentication"
    - E.g., Facebook Trusted Contacts
  - Somewhere you are
    - Location as an authentication factor
  - Etc., etc.

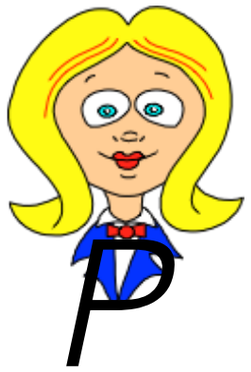
What can we do about the password problem?

Password cracking



Eavesdropping

Alice



P

(secret)

(secret)

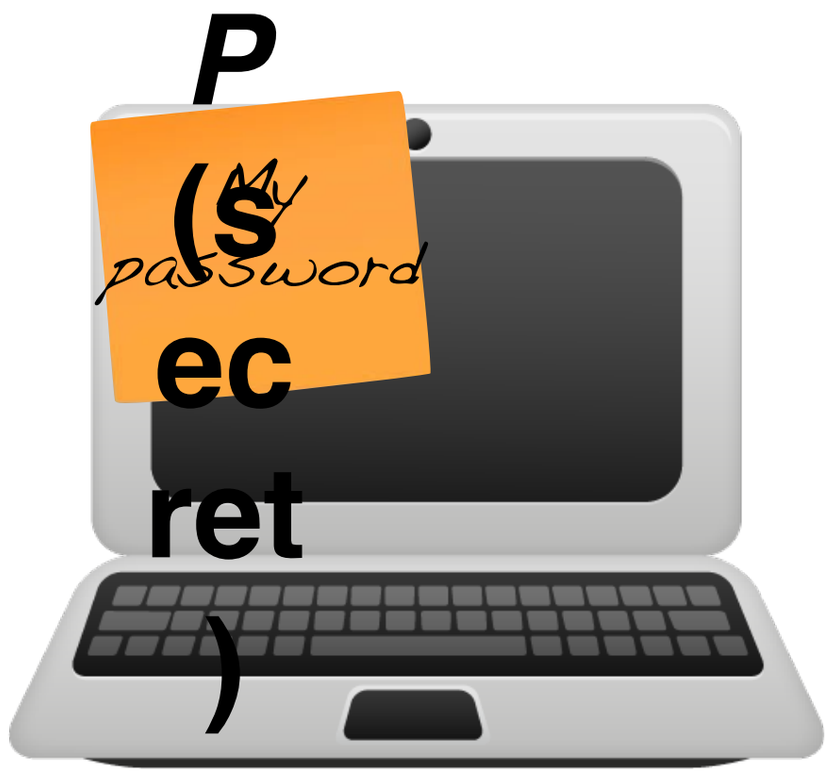
Eve

h(*P*)

Sticky notes

Google Glass

Eve



Alice

Visual capture

Malware

Alice

P

(secret)

Eve

E.g., keystroke logger

Phishing

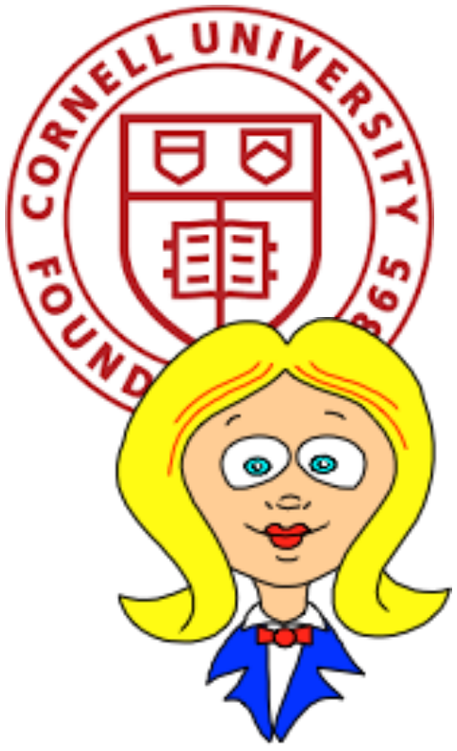
Alice

P

(secr
et)

Eve

Social engineering



Alice

"Hi, Eve. This is Cornell IT. (Go Big Red!!) A hacker has broken into your account, and we need to change your password..."

Eve

Idea 1: User-driven password changes

- Common interval: 90 days
- May help sometimes, but…
  - 90 days is a long time!
  - Helps users forget passwords
    - Estimated $150 cost per user per year
      - META group estimate: 1.75 help desk calls a month; Gartner group: 30% of calls are for password resets; Forester research: $25 / call
    - Makes social engineering *worse*

Idea 1: User-driven password changes

- How do users change their passwords?

```
Password1

Password2

Password3

Pa$sword1
```

Yinqian Zhang, Fabian Monrose, Michael K. Reiter: The security of modern password expiration: an algorithmic fra

Idea 2: One-time passcode token

Alice

789128

~~789128~~
001025
330236
919511
668336
…

~~789128~~
001025
330236
919511
668336
…

A scratch-off variant

- **Pros:**
  - Fits in wallet
  - Recyclable
  - You feel as though you have a chance of winning the lottery
- **Cons:**
  - Winning the lottery just means you can log into your bank account
  - Messy, inconvenient
  - Limited-use

One-time passcode tokens

“Something you have” authentication factor

Many types

(Proof that security can be stylish)

How a time-based token works

secret key

K

Alice

P_T (e.g., 790062)

K

$P_T = F(K, T)$

T

$P_T = F(K, T)$

T

Similar for counter-based token

secret key

K

Alice

Pc (e.g., 878883)

K

✔

Pc = F(K, C)

Pc = F(K, C)

C←C+1

C←C+1

Adversarial model and security goal?

- Adversarial model:
  - One-time passcode tokens assume an *eavesdropping / passive* adversary
- Security goal:
  - Assume that the adversary learns an arbitrarily (polynomially) long sequence of passcodes P_1, P_2, … P_n.
  - We want adversary not to be able to guess P_{n+1}.
  - What does this mean?
    - Ideally, adversary can do no better than random guess at P_{n+1}.
    - Looks like what crypto primitive?

What's the function *F*?

Some (simplified) variants used in practice:

- $F(K,C) = H(C \,||\, K)$
  - E.g., Old RSA SecurID used proprietary hash (Alleged SecurID Hash Function (ASHF))
    - 64-bit key of 10% of tokens recoverable with two months of passcodes
      - Biryukov, A., Lano, J., & Preneel, B. (2004, January). Cryptanalysis of the alleged SecurID hash function. In Selected areas in cryptography (pp. 130-144). Springer Berlin Heidelberg.
    - Why did they roll their own?
      - Invented in 1985…
- $F(K,C) = \text{AES}_K(C)$
  - E.g., RSA SecurID today (simplified)
- $F(K,C) = \text{HMAC}(K,T)$
  - In OATH standard, RFC 6238 Time-Based One-Time Password Algorithm (TOTP)
  - E.g., Google authenticator

What's the function *F*?

Output needs to be truncated for passcode display

- E.g., $P_C = F(K,C) = H(C \| K) \bmod 1{,}000{,}000$ (for 6 digits)
  - OATH approach
- Let P'_C denote *untruncated* passcode
  - We'll come back to this…

What happens if Alice pushes the button but doesn't authenticate?

K ***Pc*** ***K***

C ***C-1***

$P_C = F(K, C)$ $P_{C-1} = F(K, C-1)$

C←C+1

The fix: accept a *window* of *W* passcodes

Alice

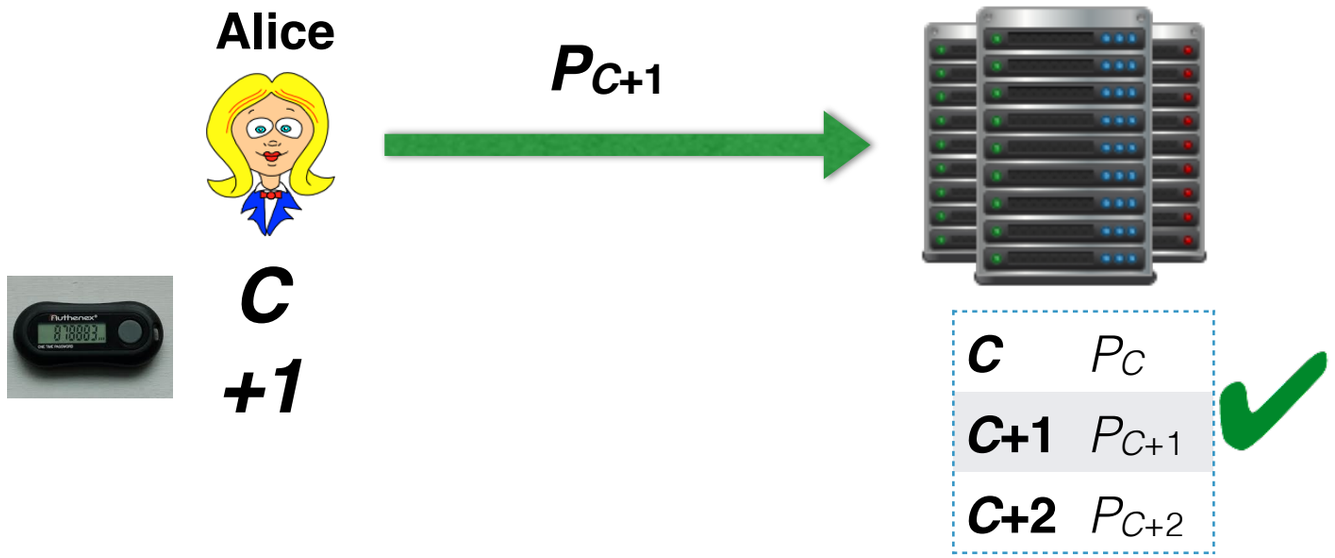
C+1

P_{C+1}

C	P_C
C+1	P_{C+1}
C+2	P_{C+2}

✔

The fix: accept a *window* of *W* passcodes



Drawback?

- Now adversary can guess any of *W* passcodes to impersonate Alice
- I.e., window size *W* gives increases adversary's success probability by factor of *W* !
- And you'll still get desynchronized if your six-year-old daughter discovers how fun it is to press the button…

How about challenge-response?

random *C*

Pc

***Pc* ≟ *F*(*K*,*C*)** ✔

- Desynchronization problems gone!
- Royal pain to use!

PIN entry

PIN entry

- User also typically enters a PIN
  - Token: “something-you-have”
  - PIN: “something-you-know”
  - Together, they are “two-factor” authentication
- But how do you protect the PIN?

The PIN transmission problem

Alice

Pc (e.g., `878883`)
+ **PIN** (e.g., 1234)

Eve

- A PIN is just like a password
- So Eve can steal it as she stole passwords
- We're struck with our original problem!

A puzzle

- Consider a 6-digit token with keyed-in 4-digit PIN

- Is there some way to build a token that displays a 6-digit one-time passcode Q_C such that:
  1. Q_C can be checked by the server;
  2. PIN can be verified by the server; and
  3. PIN isn't revealed to eavesdropper Eve?
- Remember *without* PIN:
  - $P'_C = H(C \| K)$ (untruncated)
  - $P_C = H(C \| K)$ (truncated)
  - You should use hash function for simplicity (assume random oracle model)

Good idea that doesn't work

- How about $\boldsymbol{Q_C = H(P_C \| \text{PIN})}$ (truncated to 6 digits)?
- The hashed value $P_C \| \text{PIN}$ is 6 + 4 = 10 digits long
- 10,000,000,000 possible values for $P_C \| \text{PIN}$
- Small search space—equivalent to ~33-bit key
  - E.g., Bitcoin mining rig typically achieves over 4,000,000 hashes per second
  - So Q_C can be cracked and PIN extracted in less than an hour!

What's done in practice?

- Idea 1: Check the PIN on the token.
  - If it's wrong, don't emit passcode or else emit incorrect passcode.
- Idea 2: $Q_C = H(K \,||\, C \,||\, \text{PIN})$ (truncated) (or $Q_C = H(P'_C \,||\, \text{PIN})$)
  - Why?
- Idea 3: $Q_C = P_C + \text{PIN}$ (digitwise addition)
  - This turns out to be a form of encryption of the PIN under the passcode…
  - E.g.:

```
  878883      P_C
+  1234       PIN
--------
  879017      Q_C
```

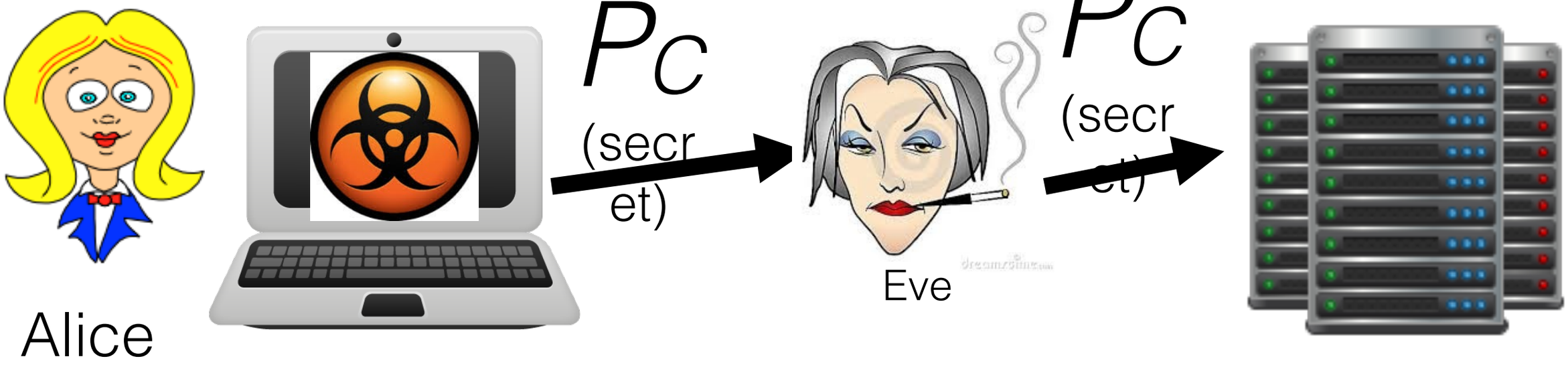
Duress PINs

- If user is physically threatened…
- Can enter a second, special "duress" PIN
- E.g., 1234 ⇒ 4321
- Server still authenticates user.
  - But it sounds silent alarm, calls police, calls in U.S. Marines, etc.
- Rumored use in ATMs
- Nice idea, but not actually in common use.

Problems with authentication tokens

Man-in-the-middle attacks

- Phishing, malware, social engineering can all capture at least one passcode
- So Eve can impersonate Alice at least once



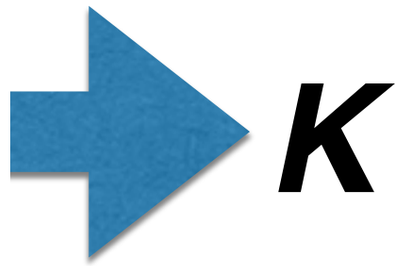
Lunchtime attacks

Mallory

- You leave your token on your desk during lunch.
- Mallory steals into your office, breaks open your token and extracts secret.
- Mallory replaces token so you don’t know about attack.
- Mallory uses your passcodes and impersonates you…

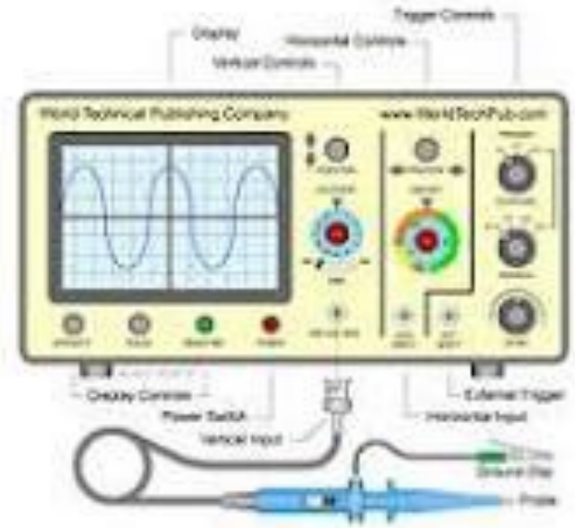
Lunchtime attacks

Mallory



- Countermeasures:
  - Tamper resistance
    - E.g., delete key if tampering detected (standard method)
  - Tamper detection
    - Covert channel in passcodes to alert server

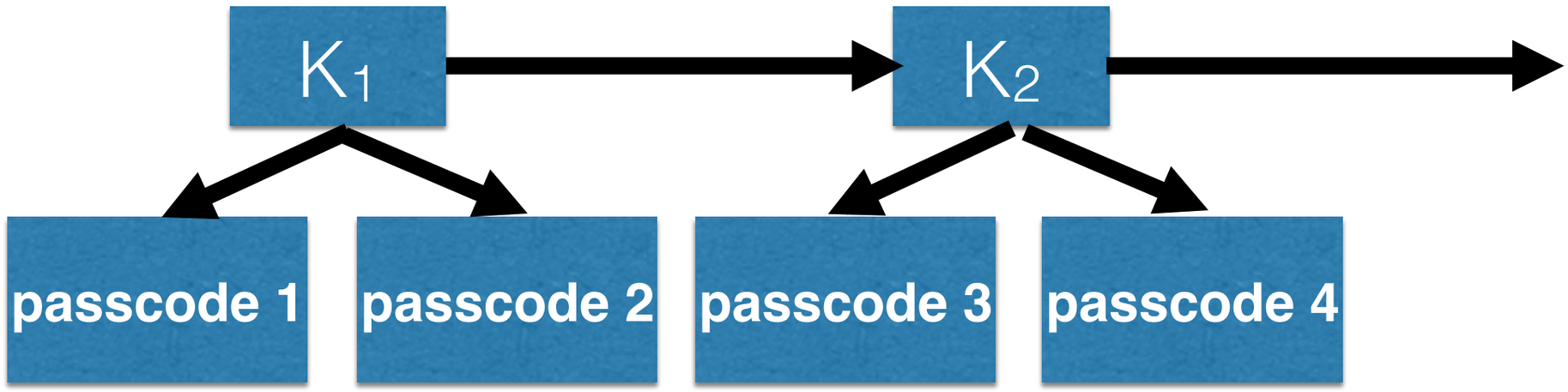
Power-analysis attacks



Mallory

- Counter-countermeasure to tamper protections
- Idea:
  - Collect power traces with minimal device tampering
  - Power traces reflect key-dependent operations
- Differential power analysis (DPA)
  - Idea: Statistically analyze power traces over many computations and a range of inputs
  - Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Differential power analysis." In Advances in Cryptology—CRYPTO'99, pp. 388-397. Springer Berlin Heidelberg, 1999.

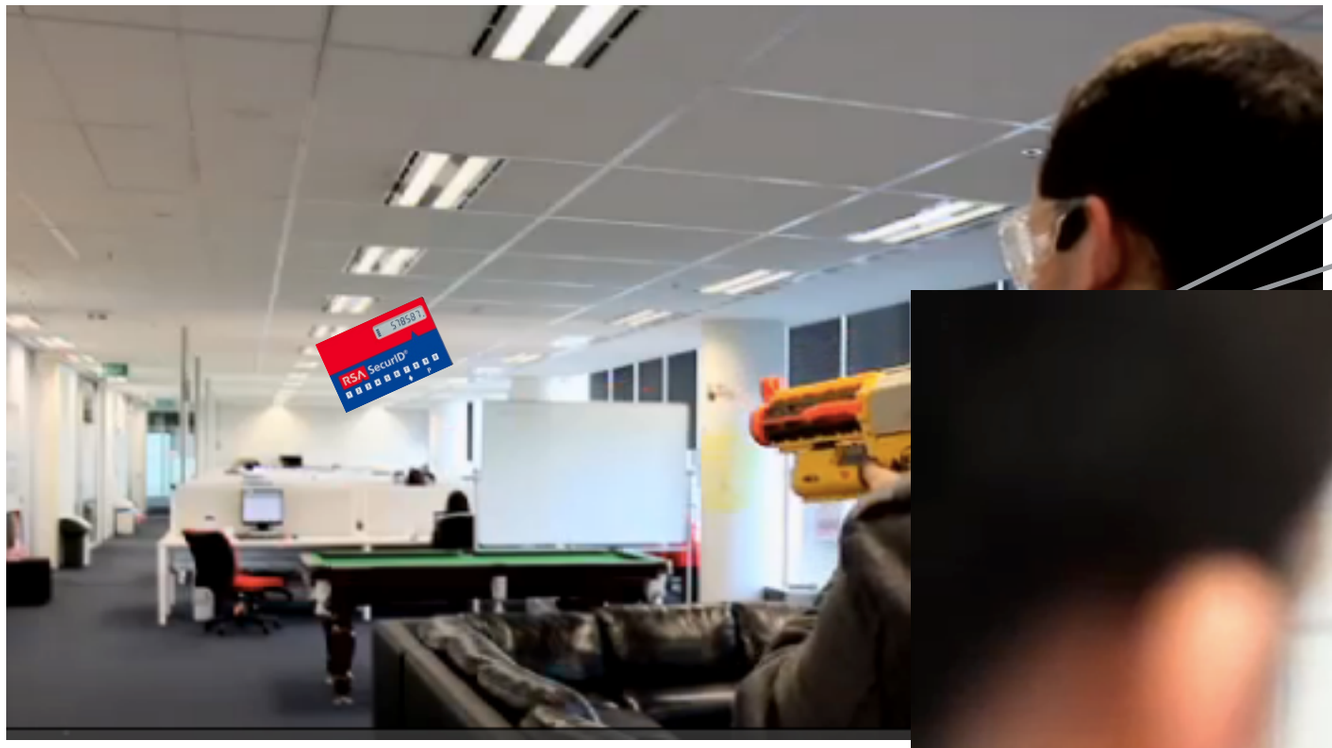
And the counter-counter-countermeasure?



- Counter-counter-countermeasure for DPA attacks
- Idea: Synchronous token-server key update
  - P. Kocher. Leak-resistant cryptographic indexed key update. U.S. Patent 6,539,092. March 25, 2003.
- Why does this help?

Poor usability

- Things people don't like:
  - Wearing authentication tokens as necklaces, carrying them everywhere, etc.
  - Transcribing passcodes + PINs
- Users dislike using tokens for authentication…



Replacing clay pigeo

Testing friends' psychic abilities

Using for game of go-fish

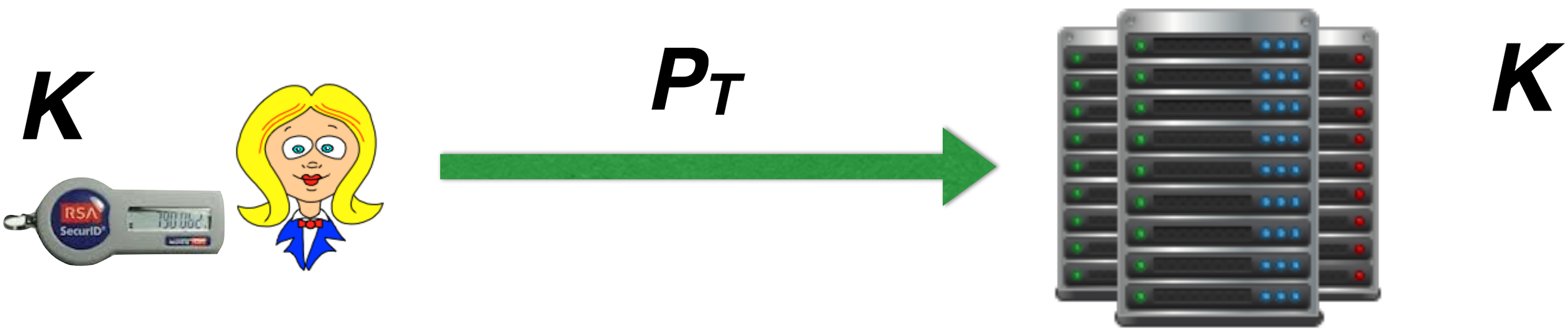
Lost, forgotten, and broken tokens

- **The Achilles heel of "something you have"**
- How to recover?
  - Help desk call for temporary password
  - Life questions…

Cost

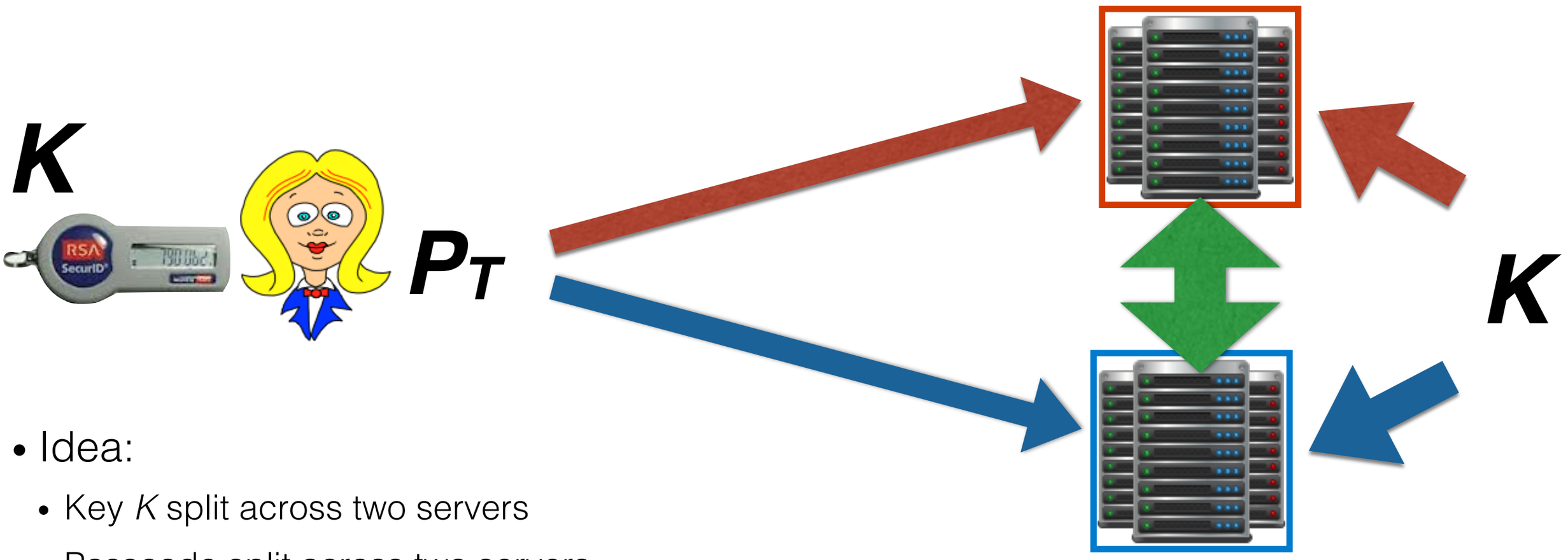
- Tokens can cost $50-60 apiece
- Some lower-cost options available…
  - E.g., Deepnet GridID

Server compromise



- Note: We can't even protect server-side secrets via hashing!

Countermeasure: Distributed Authentication



- Idea:
  - Key *K* split across two servers
  - Passcode split across two servers
  - Servers verify correctness of passcode through distributed cryptographic operation
  - Compromise of one server does not reveal *K* or passcodes
  - Extensible to *k*-out-of-*n*
- E.g., Dyadic Security offers as commercial product
- Note: public-key crypto isn't practical because of short passcode lengths!

Other problems

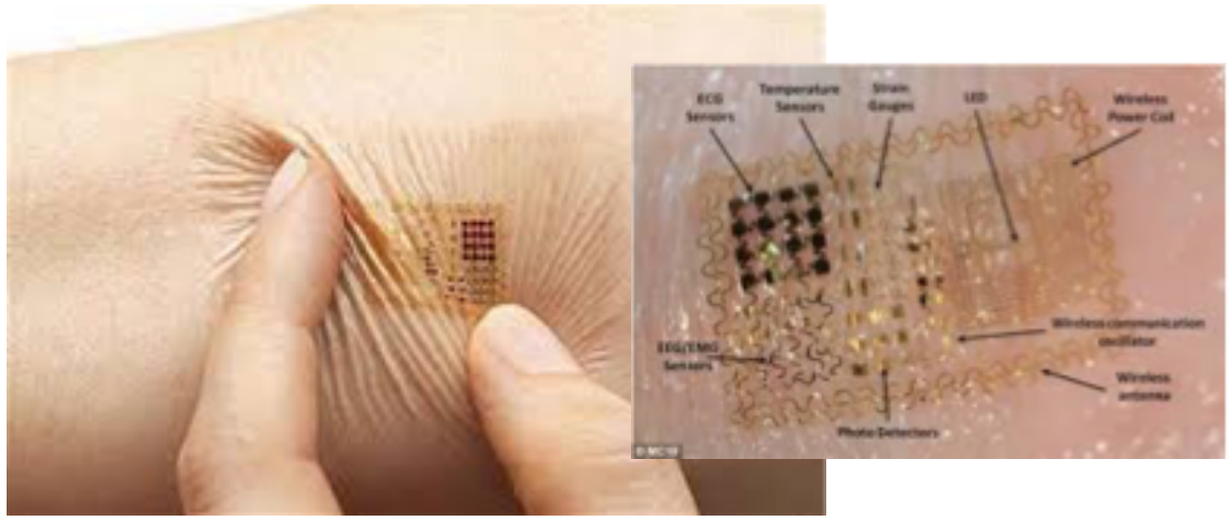
- Mobile devices are vulnerable to malware
- SMS sometimes used; can be compromised in other ways
- Consumers often don’t activate when it’s optional

The future of authentication tokens

The authentication situation is desperate.

But Motorola has an answer (two, actually).

Good for teenagers: "… you can be sure that they'll be far more interested in wearing an electronic tattoo, if only to piss off their parents…"

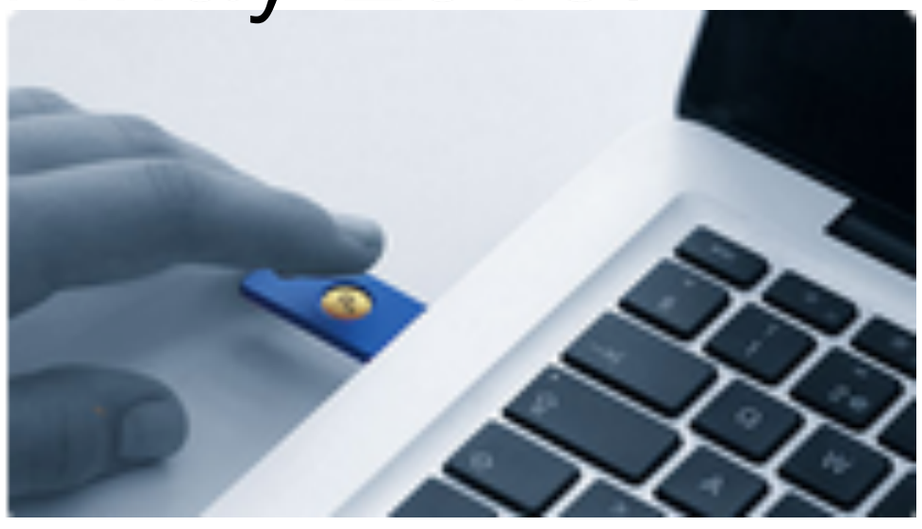


"The pill features a small chip with one switch that uses your stomach acids to activate an 18-bit ECG-like signal inside your body."

Already FDA approved.

FIDO U2F tokens

- FIDO (Fast IDentity Online) Alliance
  - Heavily supported by Google
- Universal Second Factor (U2F)
- FIDO Alliance. Universal 2nd Factor (U2F) Overview, FIDO Alliance Proposed Standard. 14 May 2015.



Web authentication with U2F

Registration

- Independent credentials for each *origin* S
  - "Origin" is (protocol, hostname, port)-triple
- To register with S, Device generates origin-specific key pair (SK_S, PK_s) and random *key handle* H_S
  - Sends (PK_S, H_S) to S
  - Stores SK_S locally with h(S)

Web authentication with U2F

Authentication

- Origin S sends H_S and challenge c to browser
- Browser sends H_S, hash h(S), and random challenge c to Device*
- Device checks that H_S is valid and was issued for h(S)
- Device digitally signs c
- Signature sig is sent to origin, which verifies it against PK_S

What problems in SecurID does U2F address?

- *Usability* (partly): Eliminates transcription(partly)
  - Device may communicate via USB, Bluetooth LE, NFC, etc.
- *Server-compromise*: Using public-key crypto
- *Man-in-the-middle attacks*: Origin checking

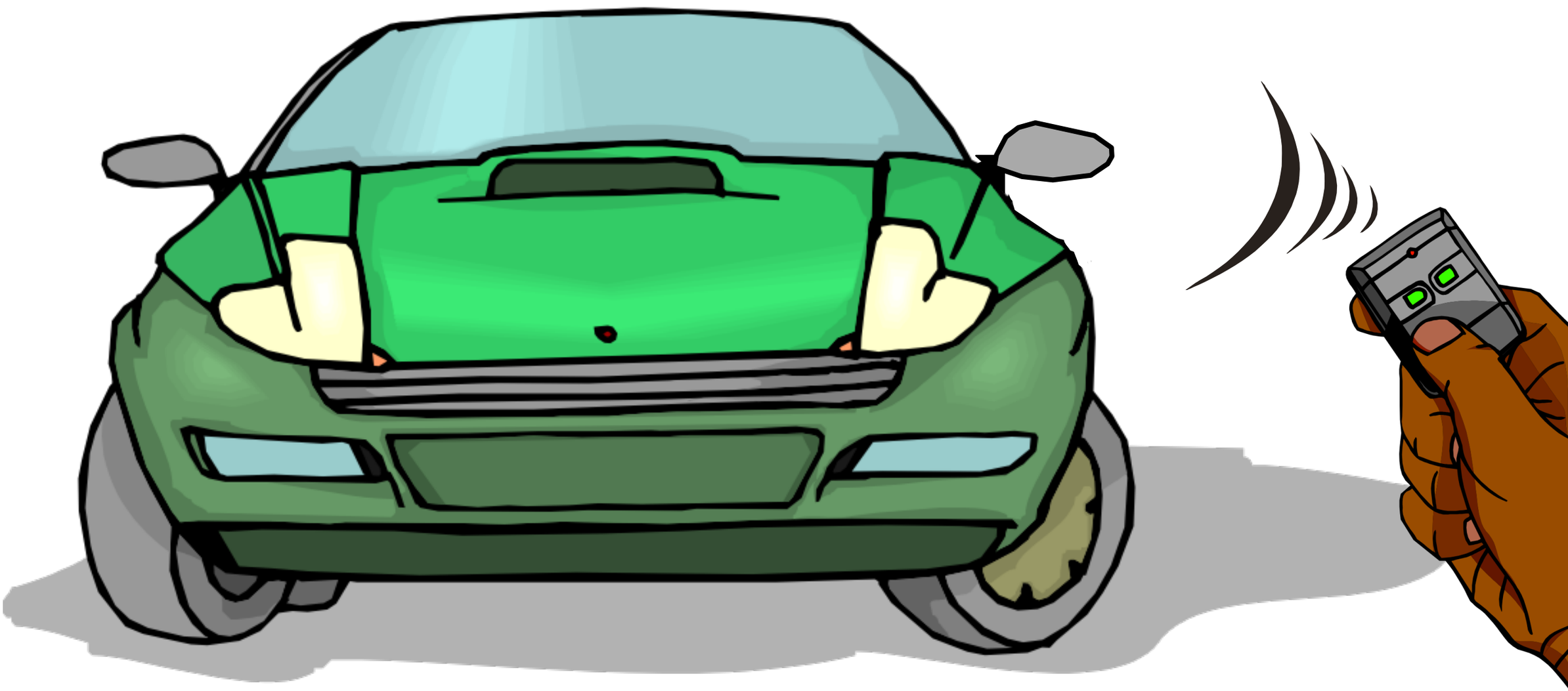
Hardware protection

- Tamper-resistant secure element (SE) may be used to protect SK_S
  - For cost effectiveness, SE has *very little memory*
  - SK_S is "wrapped" (encrypted) under symmetric key stored in SE
- SE also protects private attestation key that proves device is genuine

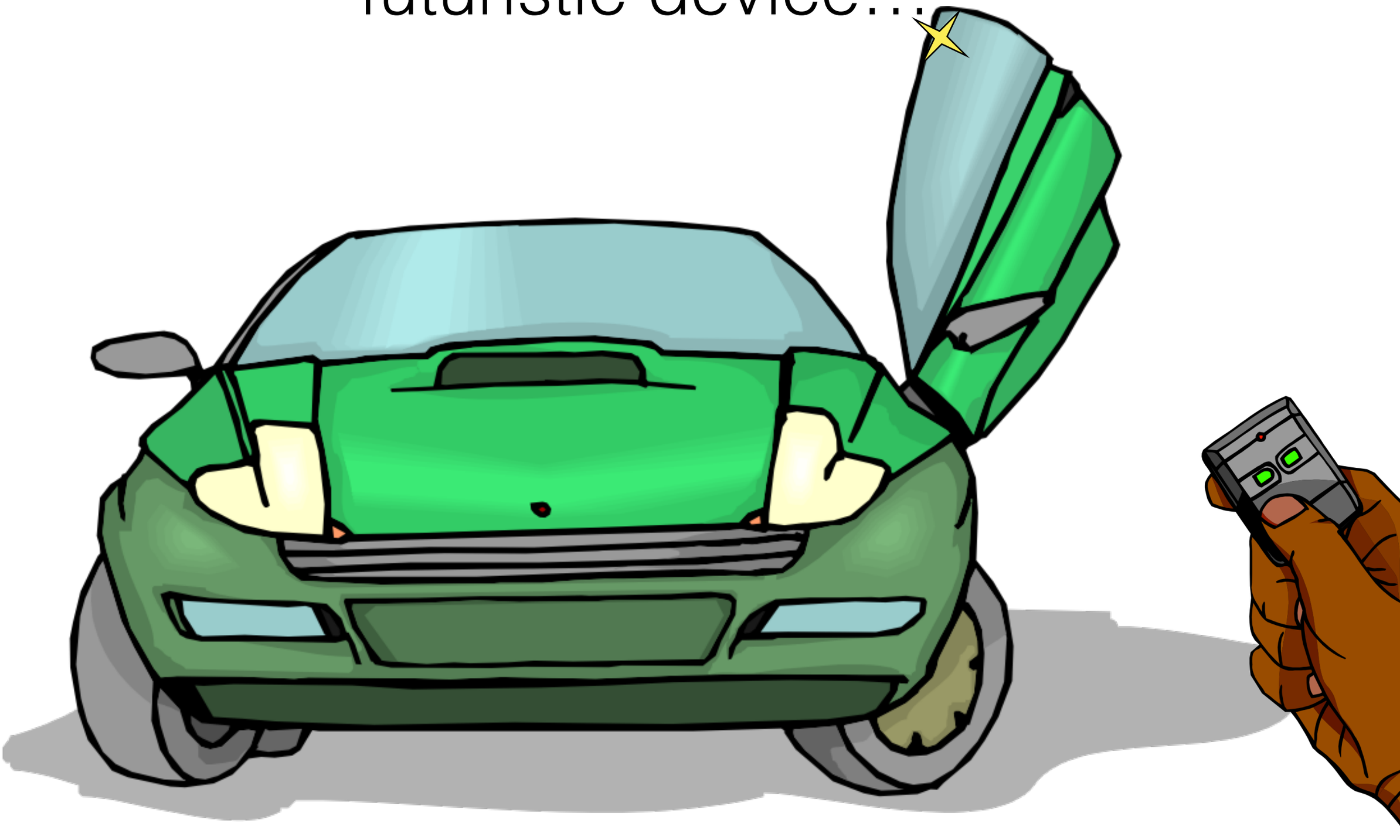
What problems does U2F not address?

- *Lost, forgotten, and broken tokens*
- *Cost*: Who's going to pay for these things
  - $18-$79 (for limited edition colors)
- *Man-in-the-browser attacks*
  - E.g., what if Trojan presents incorrect bank transaction details?
- *Usability*: User still needs to insert token, unlock token, press button, etc., etc.

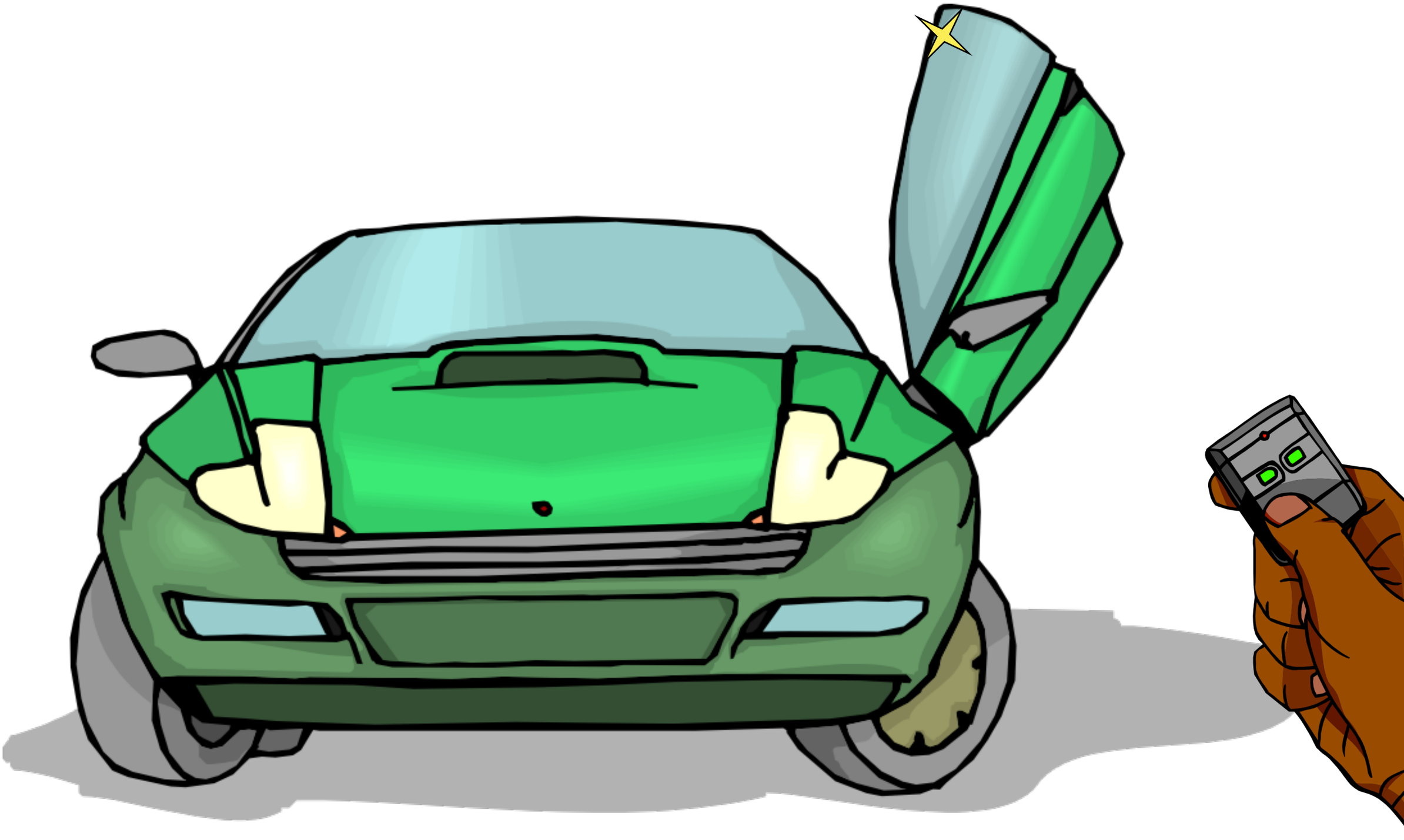
Real usability would require a futuristic device…



Real usability would require a futuristic device…



Like something invented in the 1980s…



It’s now 2016.
Why can’t I unlock my computer like my car?

It’s now 2016.
Why can’t I unlock my computer like my car?

Several wireless-authenticator research proposals

- Locking workstation when user walks away
  - Roy Want and Andy Hopper. “Active Badges and Personal Interactive Computing Objects”. IEEE Transactions on Consumer Electronics, 38(1):10–20, Feb 1992
- Proximity-based file-system decryption
  - Mark D. Corner, Brian Noble: Zero-interaction authentication. MOBICOM, pp. 1-11. 2002.
- Uber-device for web passwords, screen saver passwords, file system, PINs for consumer devices, etc.
  - Stajano, Frank. "Pico: No more passwords!." Security Protocols XIX, pp. 49-81. Springer Berlin Heidelberg. 2011.

…and several complications

- How to achieve legacy compatibility?
  - E.g., retooling of websites?
- How to balance automation with user confirmation?
- **Bluetooth, 802.11, etc. historically unreliable, especially device-to-laptop connection**
- But this is changing…

Nymi

- User authenticates to wristband when putting it on
  - Biometric authentication via heartbeat (EKG)
- Band remains attached -> user remains authenticated
- Motion sensors + NFC, Bluetooth, etc. offer futuristic authentication options
  - E.g., gesture to open car trunk
- "Prosthetic biometric"

"Prosthetic biometrics"?

nymi™

Last year…

MasterCard and Nymi say they've completed the first heartbeat-authenticated mobile payment in the wild

MARK SULLIVAN AUGUST 12, 2015 9:58 AM

VB

Will smartwatches be the killer authenticator?

- They already have
  - Physiological sensors
  - Motion sensors
  - Detachment detection
  - NFC + Bluetooth
  - etc., etc.

SALON

WEDNESDAY, SEP 10, 2014 01:43 PM EDT

A killer app for the Apple Watch: Gun control

Interesting related technology: Eddystone-EID

- Eddystone: Google BLE beacon format standard
- EID: Ephemeral ID
- Rotating ID just like that in user authentication token
  - $\mathrm{PRF}_K(T)$, where T is current "epoch" (length at least 512 secs.)
- Addresses both privacy and authentication
  - E.g., Samsonite Track&Go
    - Tracks luggage from mobile app within 70 meters
    - Crowdsourced location of missing luggage

Many interesting research questions

- Formal analysis of U2F protocol
  - (Attestation key looks problematic)
- Physical attacks on Nymi, smartwatches?
  - Detachment
  - Biometric impersonation? Photoplesythmography?
- New ways to detect man-in-the-browser attacks?
  - E.g., can smartwatch video laptop or mobile screen?
- Homework: Can device secretly alert servers when tampered with?