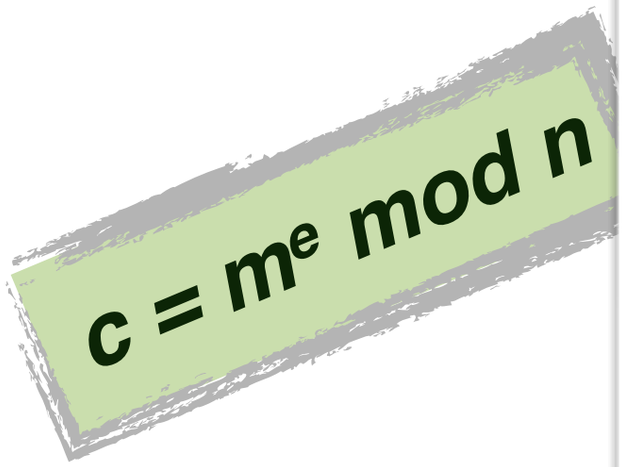
CS5438:
Security and Privacy:
Practice and Case Studies



Authentication Tokens

Instructors: Ari Juels and Vitaly Shmatikov
Spring 2016

Remember?

Matt Honan, *Wired* correspondent

How did it happen?

- Attackers started by compromising Honan's Amazon account
- Needed credit card number for Honan's Amazon account. How did they learn it?
- Attackers called Amazon and **added** a new credit card number to Honan's account. (Name, e-mail, and billing address sufficed.)
- Attackers called Amazon to reset Honan's password. For identity verification, Amazon asked for a credit card number…

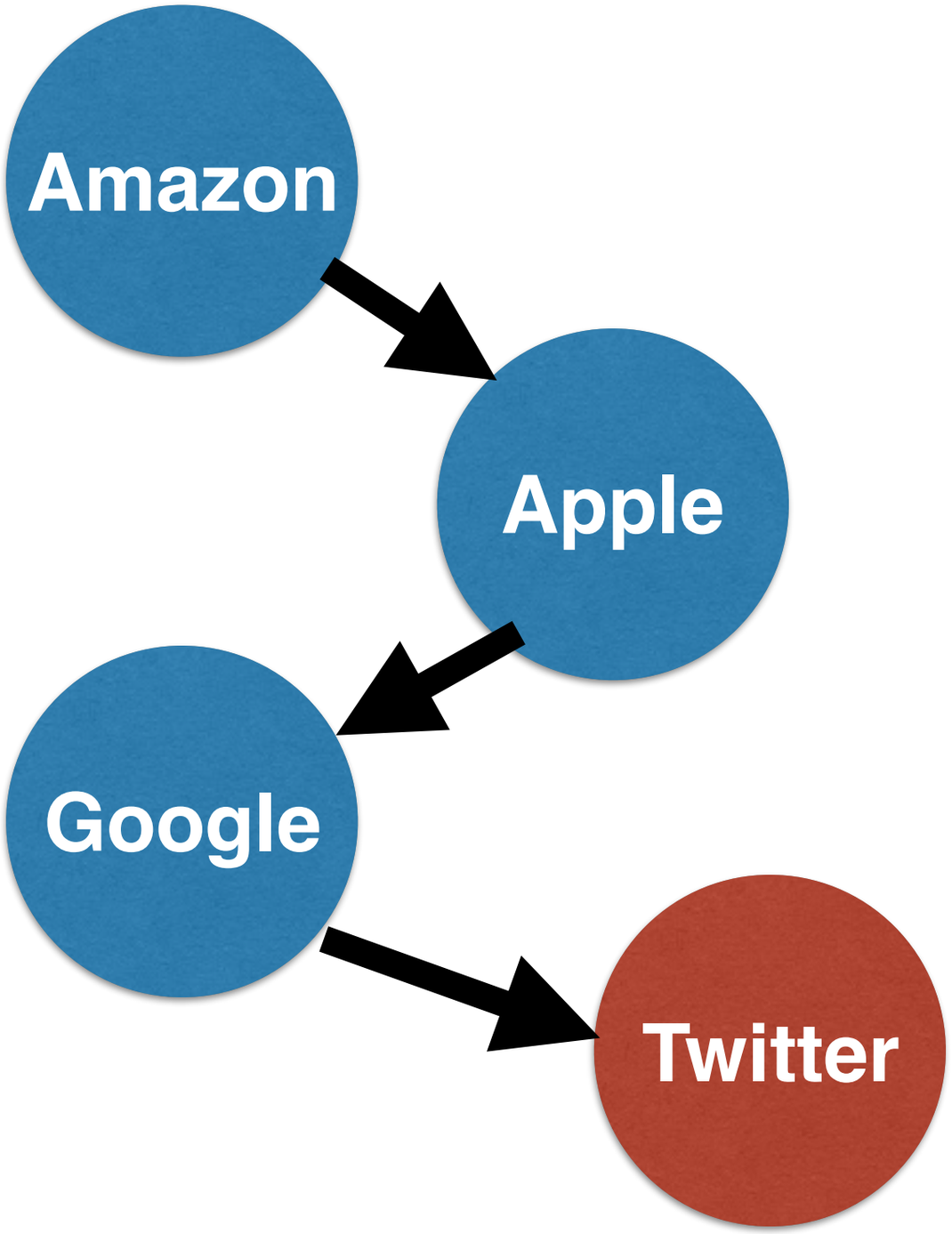
Matt Honan, *Wired* correspondent

How did it happen?

- Once logged in to Honan's Amazon account, attackers learned last four digits of real credit card numbers
- "*The very four digits that Amazon considers unimportant enough to display in the clear on the web are precisely the same ones that Apple considers security enough to perform identity verification*."

Matt Honan, *Wired* correspondent

Recap


Amazon
Apple
Google
Twitter


iCloud


Is this Mat Honan?
@mat
Follow
Clan Vv3 and Phobia hacked this twitter
Reply
Retweet
Favorite
5:12 PM - 3 Aug 12 via web · Embed this Tweet

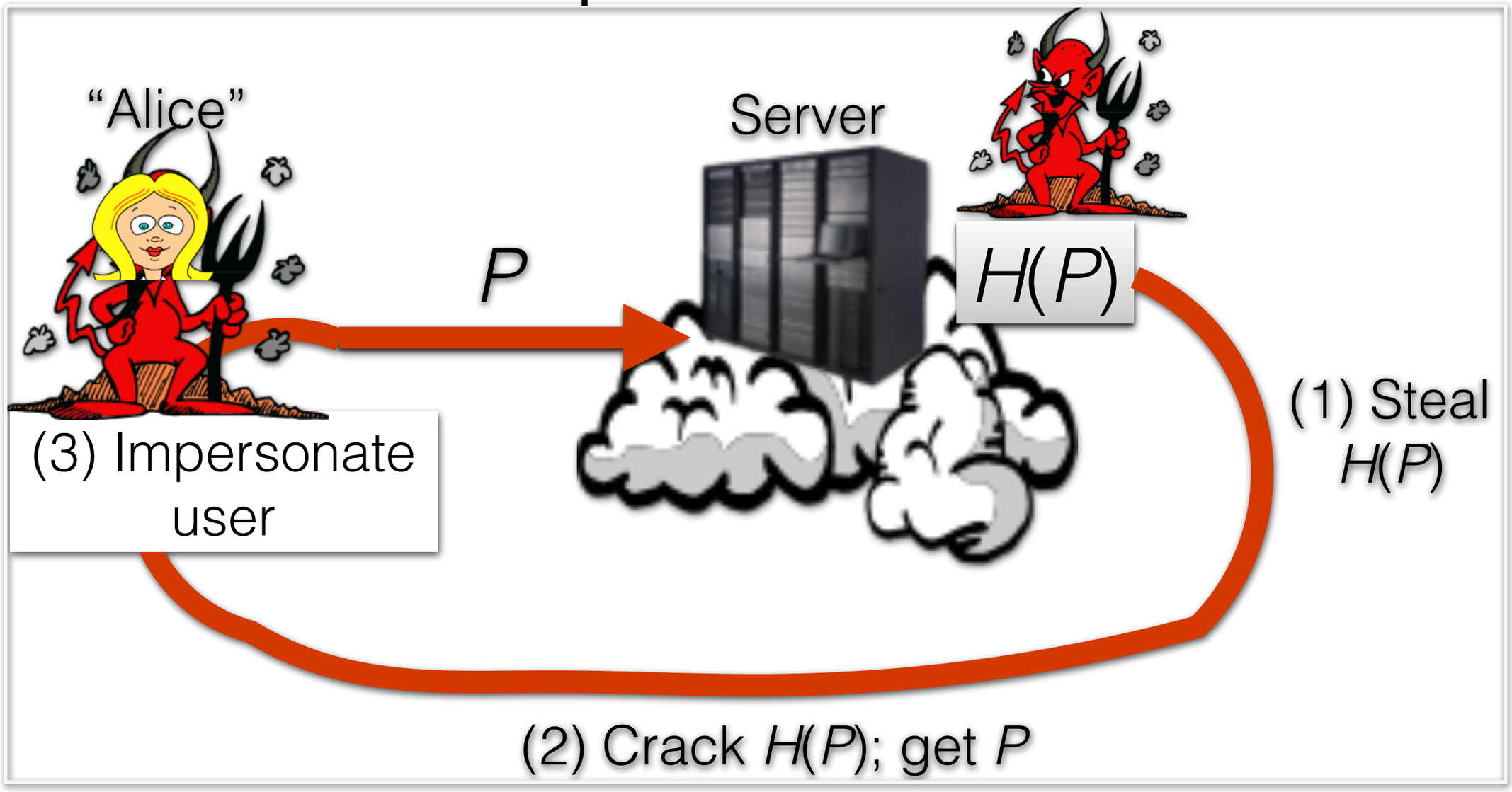
Mat Honan's recommended solution

Google Authenticator

In the beginning was the password (and it's still here)

"Something you know" authentication factor

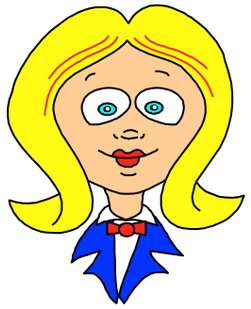
Remember the password cracking problem



But even if the server is well protected, passwords can still be stolen from *the user*.

Eavesdropping

Alice



P

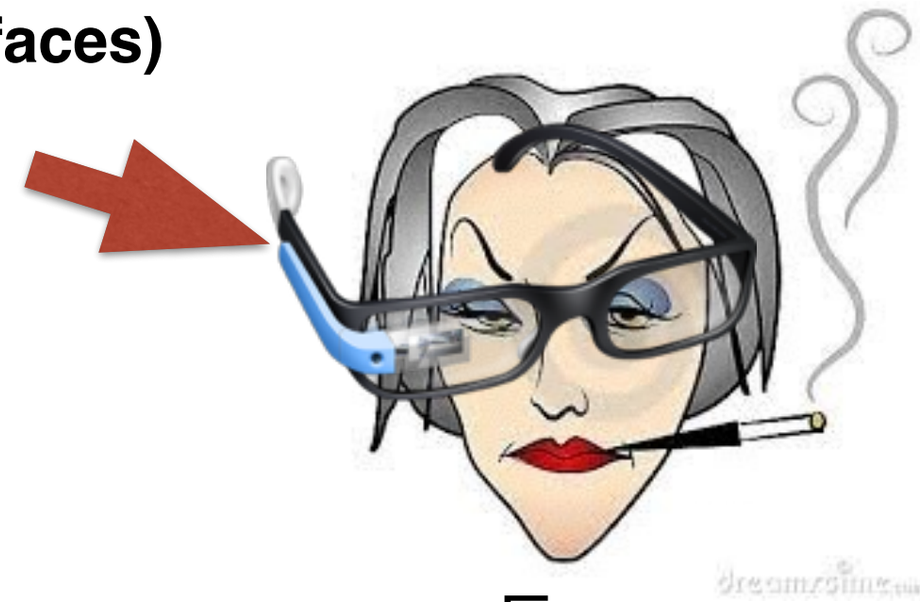
P

Eve

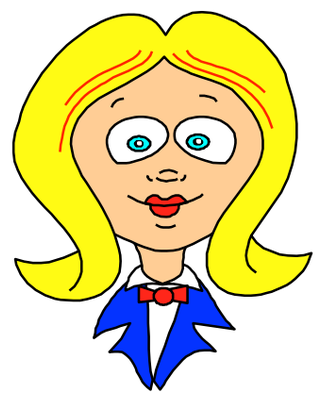
$h(P)$

Sticky notes

Google Glass
(if it resurfaces)



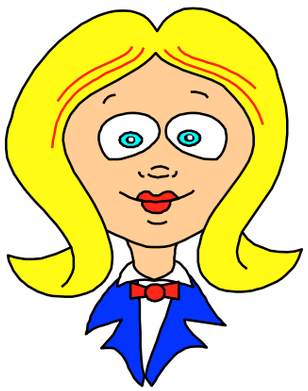
Eve



Alice

Visual capture

Malware



Alice

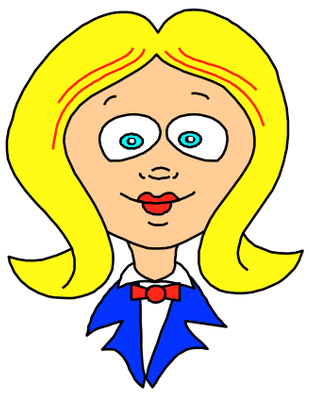
P

Eve

E.g., keystroke logger

Phishing

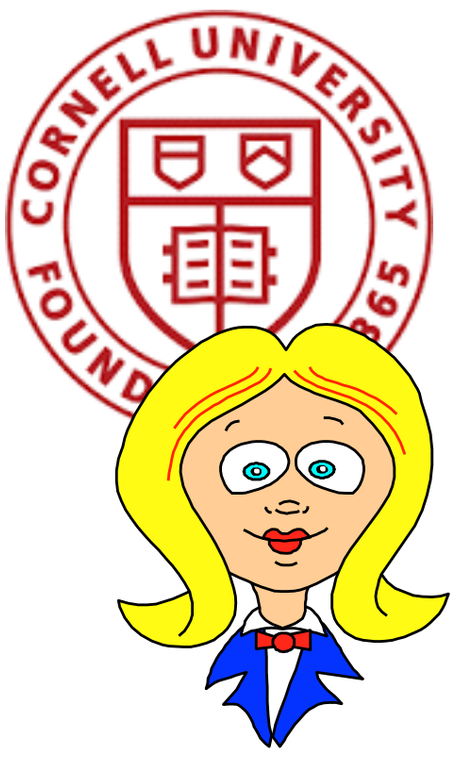
Eve



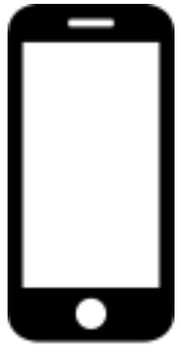
Alice

P

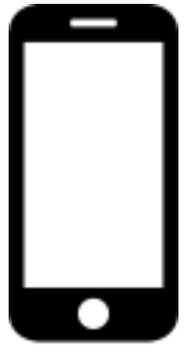
Social engineering



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
    - Password-reset questions, social engineering, etc., come into play…

Idea 1: User-driven password changes

- How do users change their passwords?

  `Password1`

  `Password2`

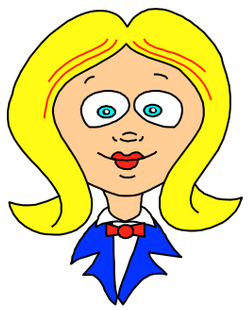
  `Password3`

  `Pa$sword1`

  - Y. Zhang, F. Monrose, M. K. Reiter: The security of modern password expiration: an algorithmic framework and empirical analysis. ACM CCS, pp. 176-186, 2010.

Idea 2: One-time passcodes

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

Another idea:

One-time passcode tokens

One-time passcode tokens

“Something you have” authentication factor

Many types

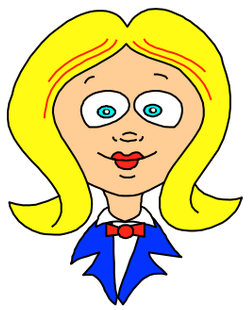
(Proof that security can be stylish)

How a time-based token works

secret key

K

Alice



P_T (e.g., 790062)

K

$P_T = F(K, T)$

T

$P_T = F(K, T)$

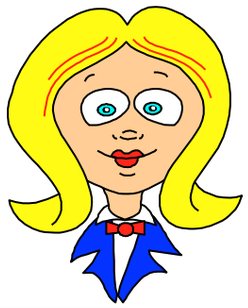
T

Similar for counter-based token

secret key

K

Alice



Pc (e.g., 878883)

K

✔

$P_C = F(K, C)$

$P_C = F(K, C)$

C←C+1

C←C+1

What's the function *F*?

- Should be hard to create passcodes without knowledge of *K*; some (simplified) variants used in practice:
  - $F(K,C) = \mathrm{AES}_K(C)$
  - $F(K,C) = H(C \,\|\, K)$
  - $F(K,C) = \mathrm{HMAC}(K,T)$ [OATH, RFC 6238 TOTP]
- Note: 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)

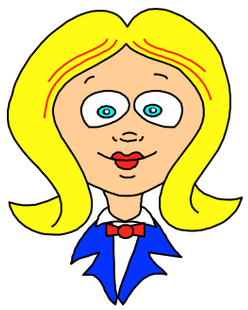
Adversarial model and security goal?

- Adversarial model:
  - Worst case assuming eavesdropping adversary?
  - Assume that the adversary learns a long sequence of passcodes P_1, P_2, … P_n.
- Security goal:
  - 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}.

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

K

Alice



Pc

K

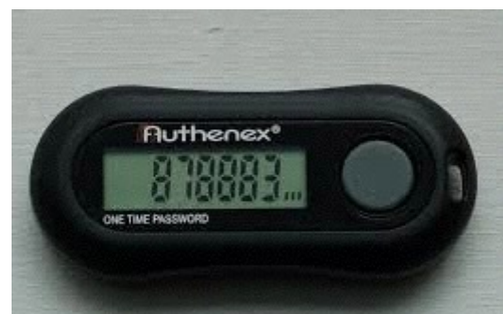
C

C-1

Pc = F(K, C)

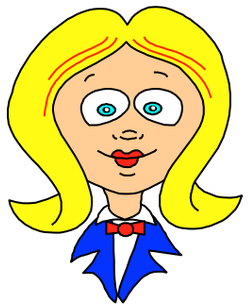
PC-1 = F(K, C-1)

C ← C+1



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

Alice

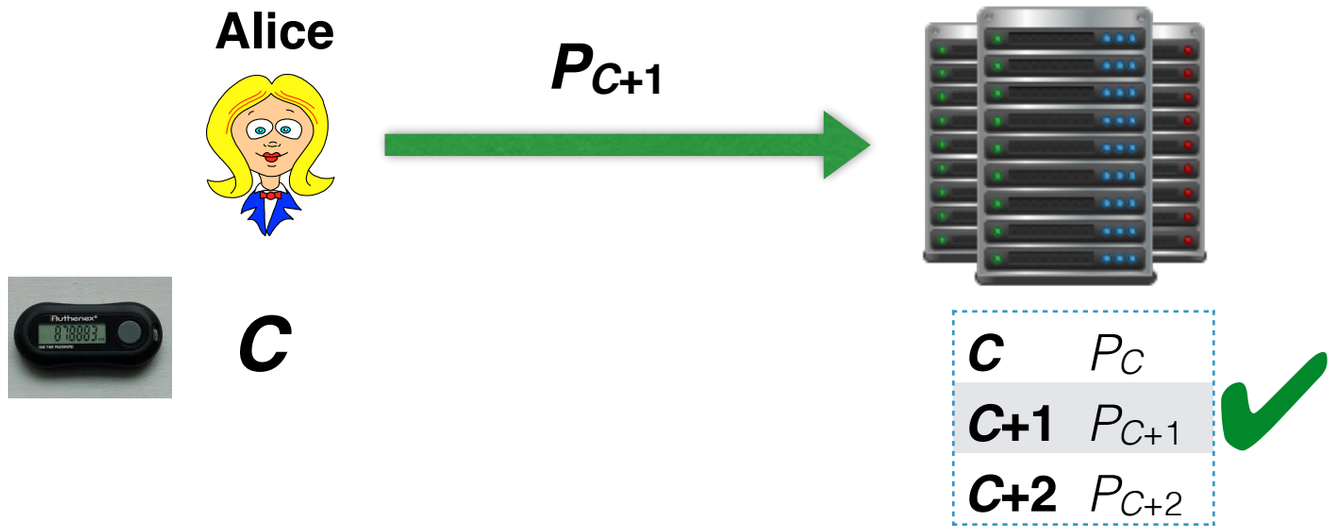


C+1

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

$P_C \stackrel{?}{=} F(K,C)$ ✔

- Desynchronization problems gone!
- Royal pain to use!

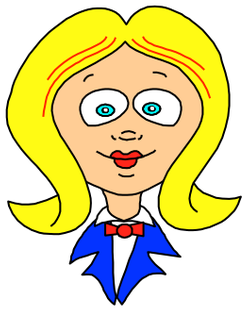
PIN entry

PIN entry

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

The PIN transmission problem

Alice



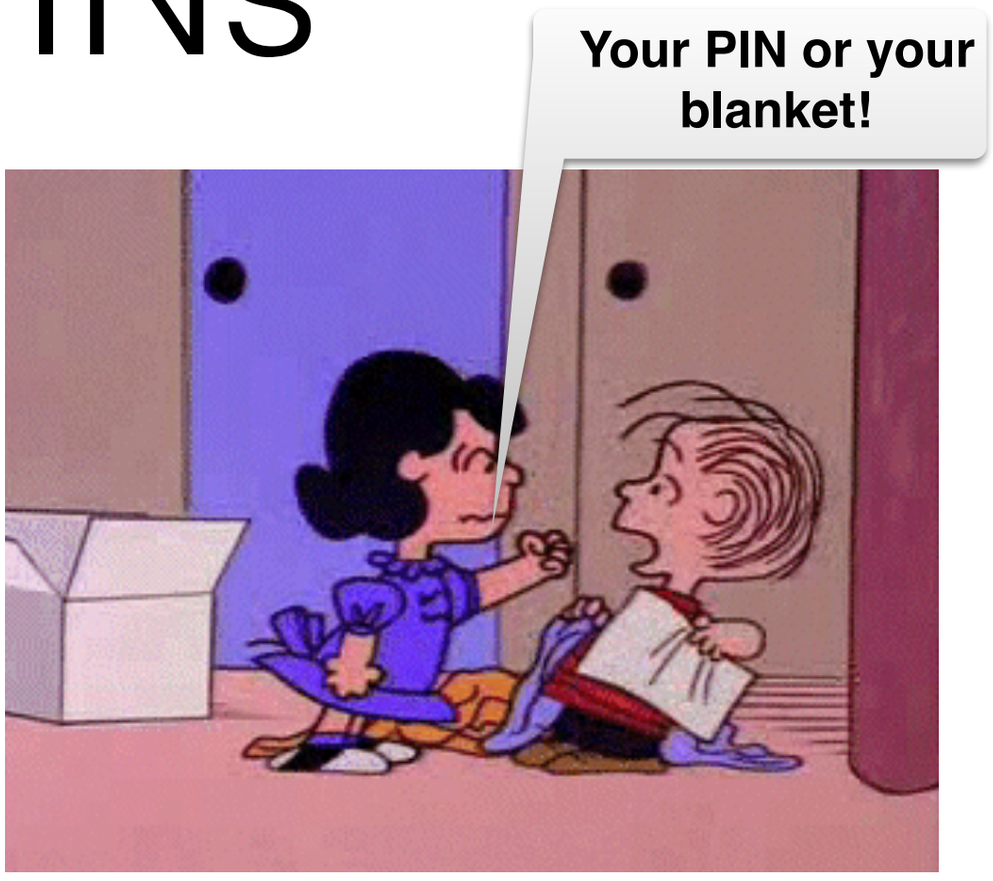
P_C (e.g., 878883)
+ **PIN** (e.g., 1234)

Eve

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

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.

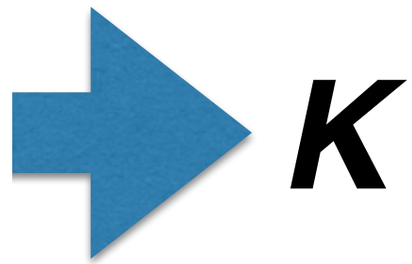


Protection against physical attacks

Mallory

What happens if there's a lunchtime attack on your token?

Mallory



K

- 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…

Funkspiel schemes

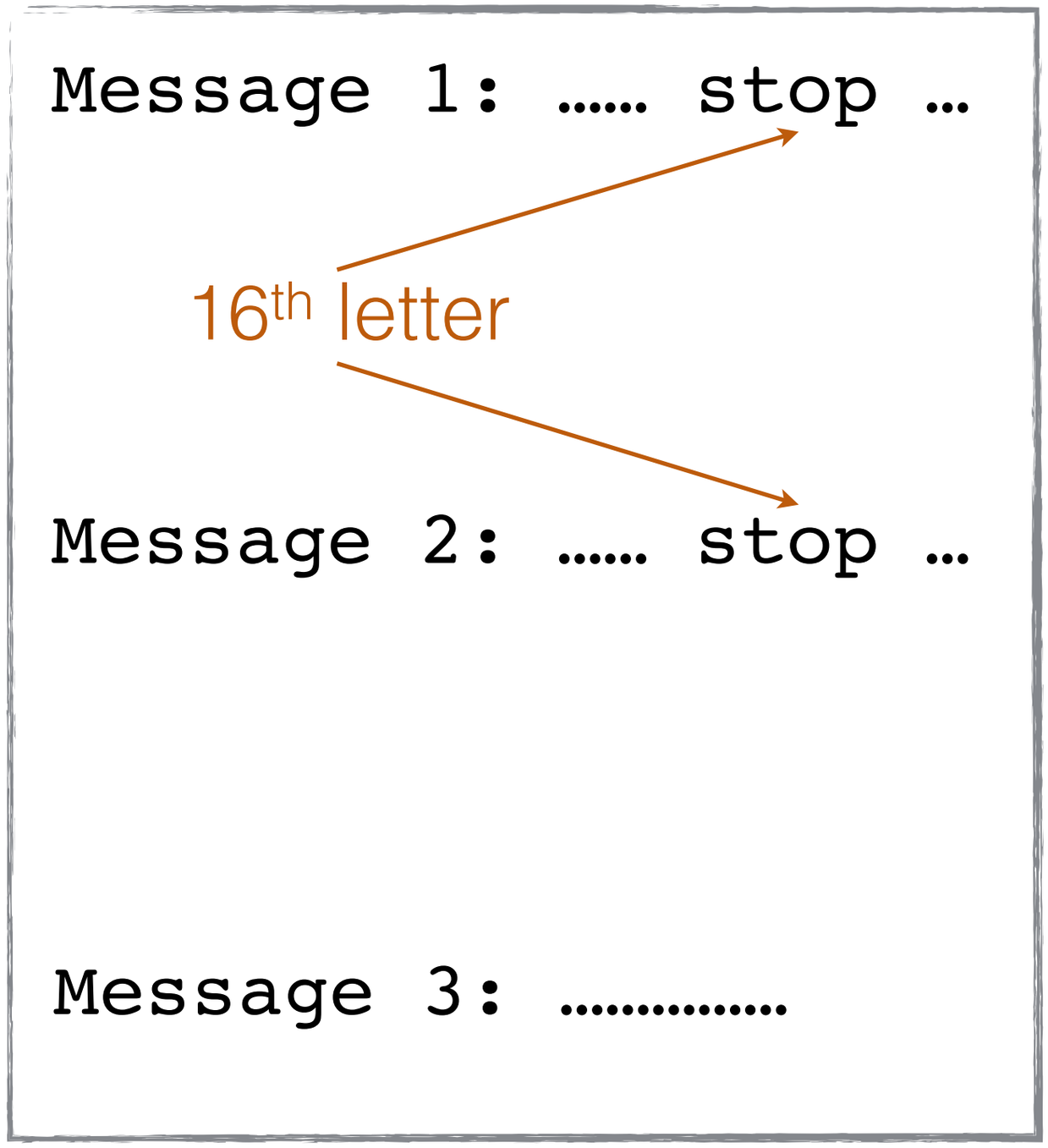
- Huub Lauwers was a Dutch agent with the Special Operations Executive (British intelligence) during WWII.
- He made radio transmissions to SOE.
- He was captured by the Germans in 1942, along with his radio.
- The Germans had also intercepted three messages.
- Germans sought to mount a “Funkspiel”, i.e., pass false messages to SOE by impersonating Lauwers.

Funkspiel schemes

- To detect the capture of agents, the SOE used a secret “message authentication code.”
  - Agents intentionally inserted special, pre-agreed errors into their messages
- The Germans knew this.
- They confronted Lauwers with his messages and demanded his code…

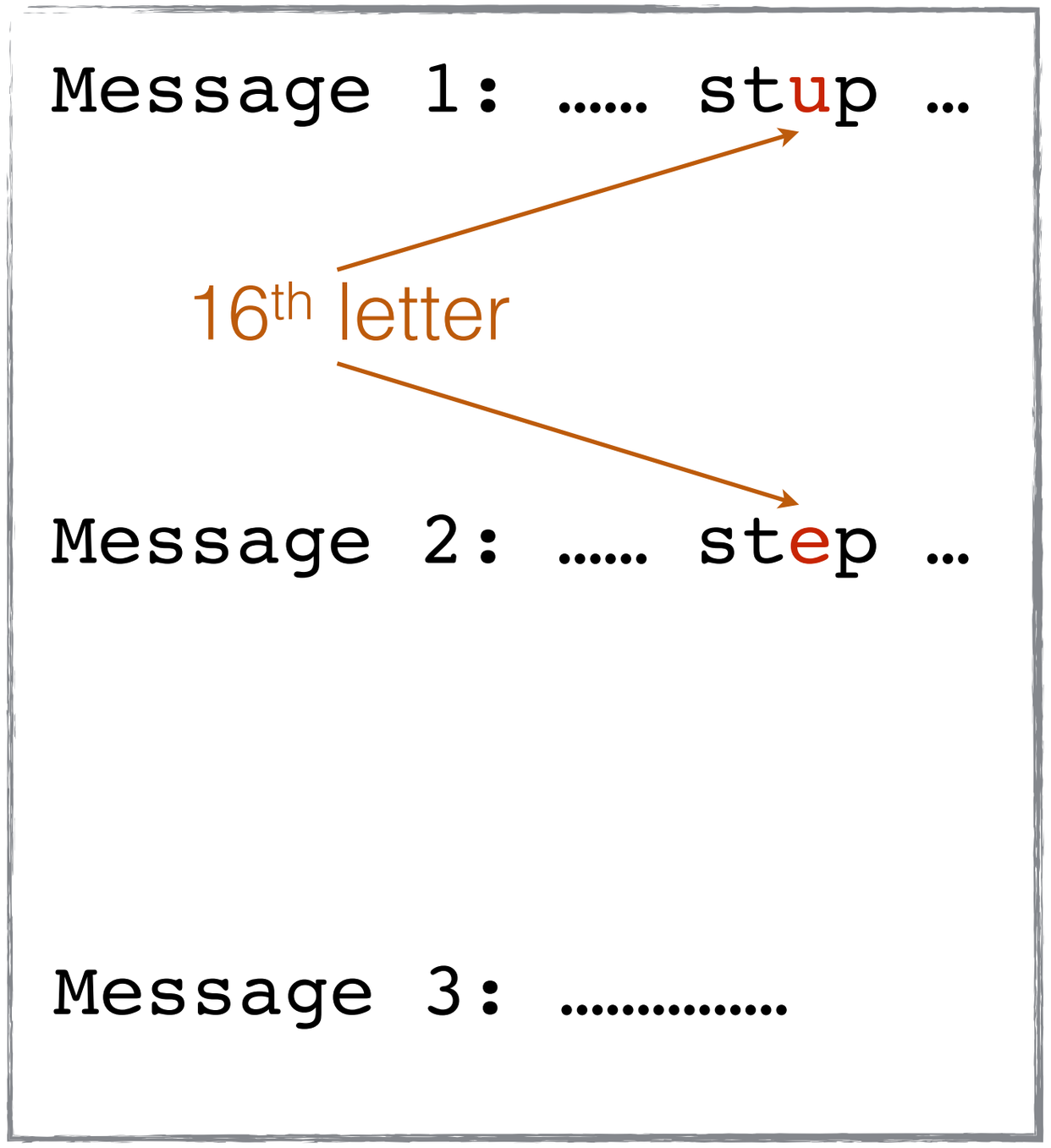
Authentication code

- Lauwers's "authentication code" was "corrupt the 16th letter of every message"



Authentication code

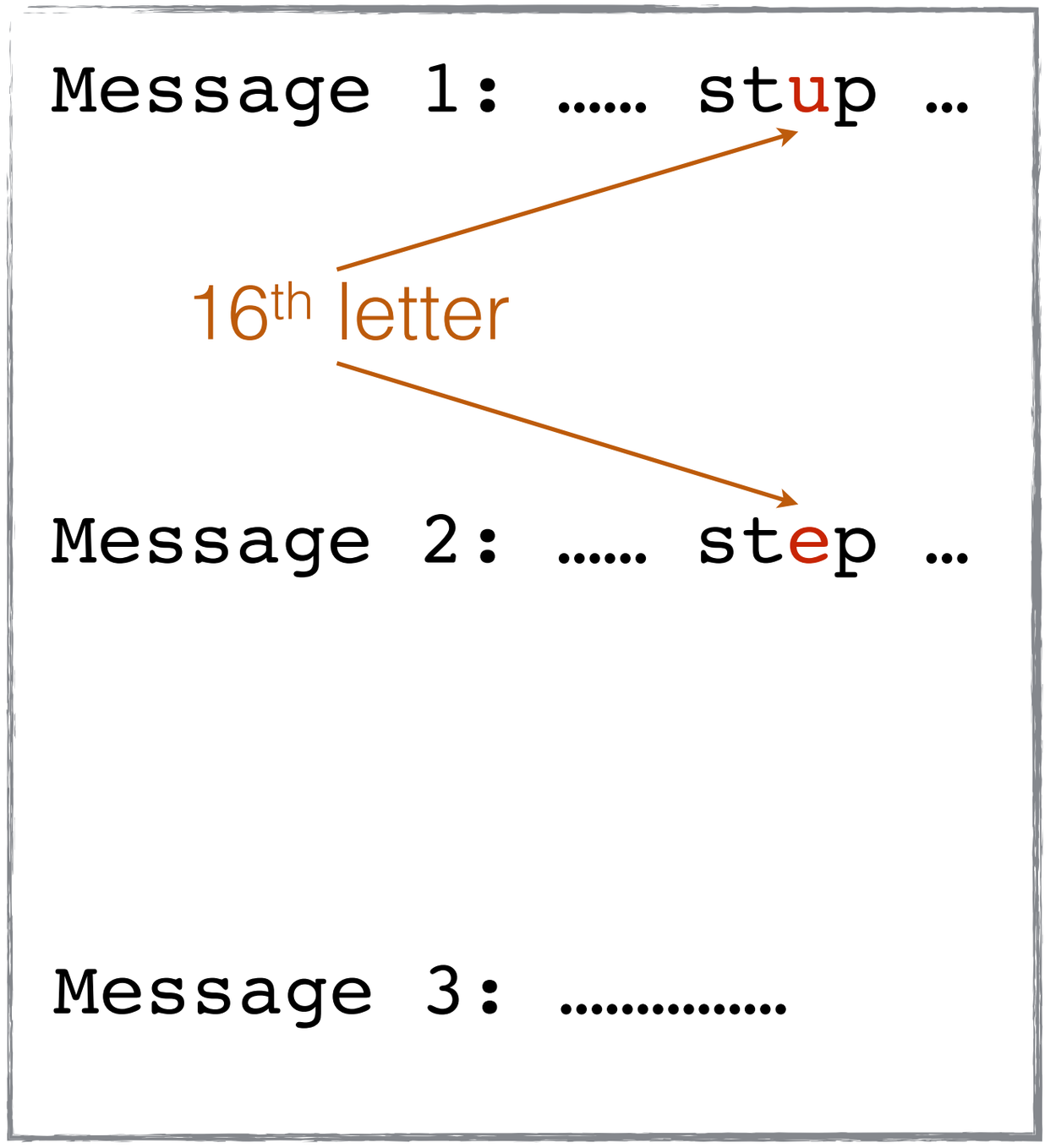
- Lauwers's "authentication code" was "corrupt the 16th letter of every message"
- Happily, Lauwers made a clever observation about his messages.
- He figured out how to fool Germans and alert SOE to his capture. How?
- He gave the Germans the wrong authentication code… "corrupt 'o' in the word 'stop'"



The result in WWII

What happened?

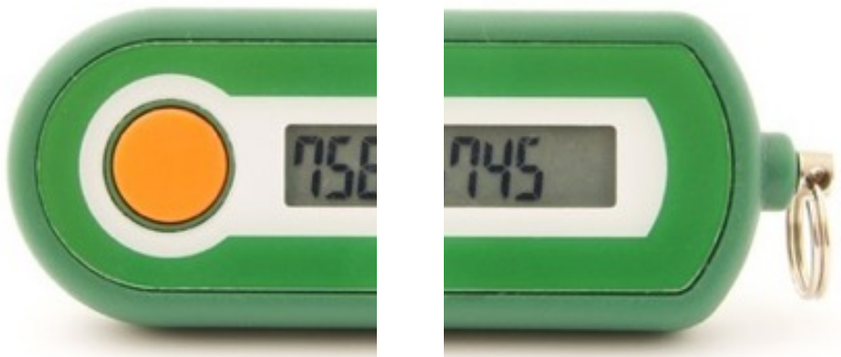
- The Germans were fooled!
- The British were fooled!
- The Germans captured many SOE agents…



Result in 21st century

Lauwers' cleverness became a product idea.

Idea: Funkspiel scheme for tamper detection



- Lunchtime attack involves tampering.
- Funkspiel idea: If tampering is detected, token embeds secret alert for server in passcodes.
- Even if adversary Mallory
  - Sees previous messages / passcodes (like Germans)
  - Breaks open token and gets future passcodes

…Mallory can’t tell if secret alert was activated!

Remember simplified token model

Alice

789128

~~789128~~
001025
330236
919511
668336
…

~~789128~~
001025
330236
919511
668336
…

(Simplified) secret alert: +1 (mod 10) for all digits in unused passcodes

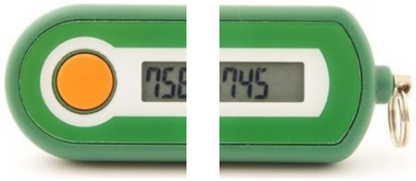
Alice

789128

~~789128~~
001025
330236
919511
668336
…

~~789128~~
001025
330236
919511
668336
…

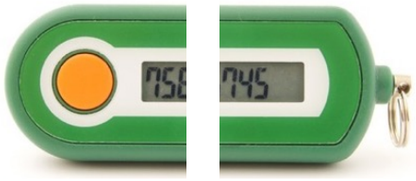
Secret alert: +1 (mod 10) for all digits in unused passcodes



~~789128~~
001025
330236
919511
668336
…

~~789128~~
001025
330236
919511
668336
…

Secret alert: +1 (mod 10) for all digits in unused passcodes



~~789128~~

112136

441347

020622

779447

…

~~789128~~

001025

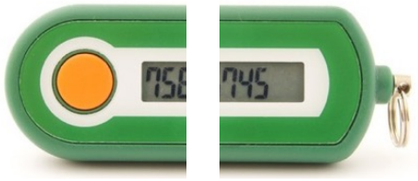
330236

919511

668336

…

Secret alert: +1 (mod 10) for all digits in unused passcodes



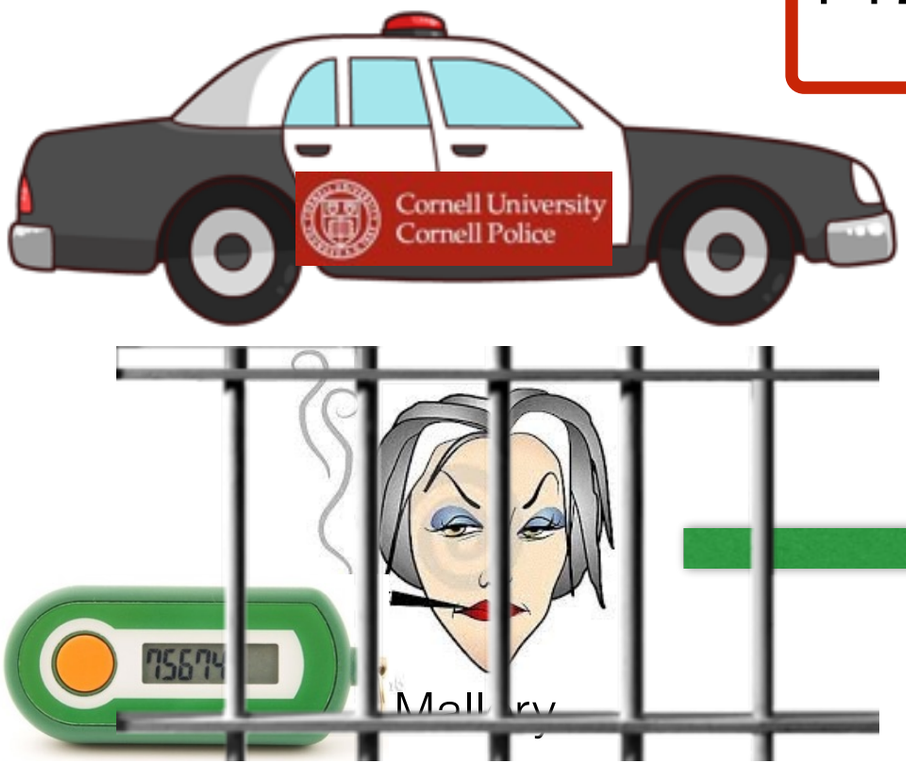
~~789128~~
112136
441347
020622
779447
…

Looks random to me…

Mallory

~~789128~~
001025
330236
919511
668336
…

112136 - 001025 = 111111 !!!



112136

~~789128~~

112136

441347

020622

779447

…

~~789128~~

001025

330236

919511

668336

…

In-class exercise

- What are the limitations of the +1 Funkspiel scheme? (Can you name two?)

- Bonus: Can you think of a fix?

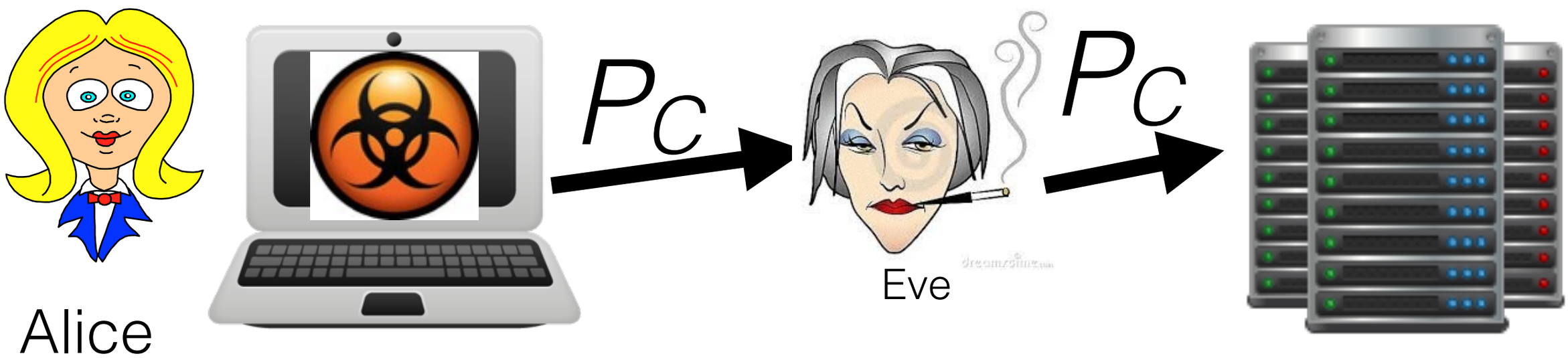
Good idea that doesn't work

- +1 is a *simplified* scheme with some problems, e.g.,
- If Mallory thinks silent alarm sounded, she can *subtract* 111111 to get valid passcode.
- Mallory can also simulate tampering by intercepting Alice's passcode and adding 111111 to it.

Building better authentication tokens

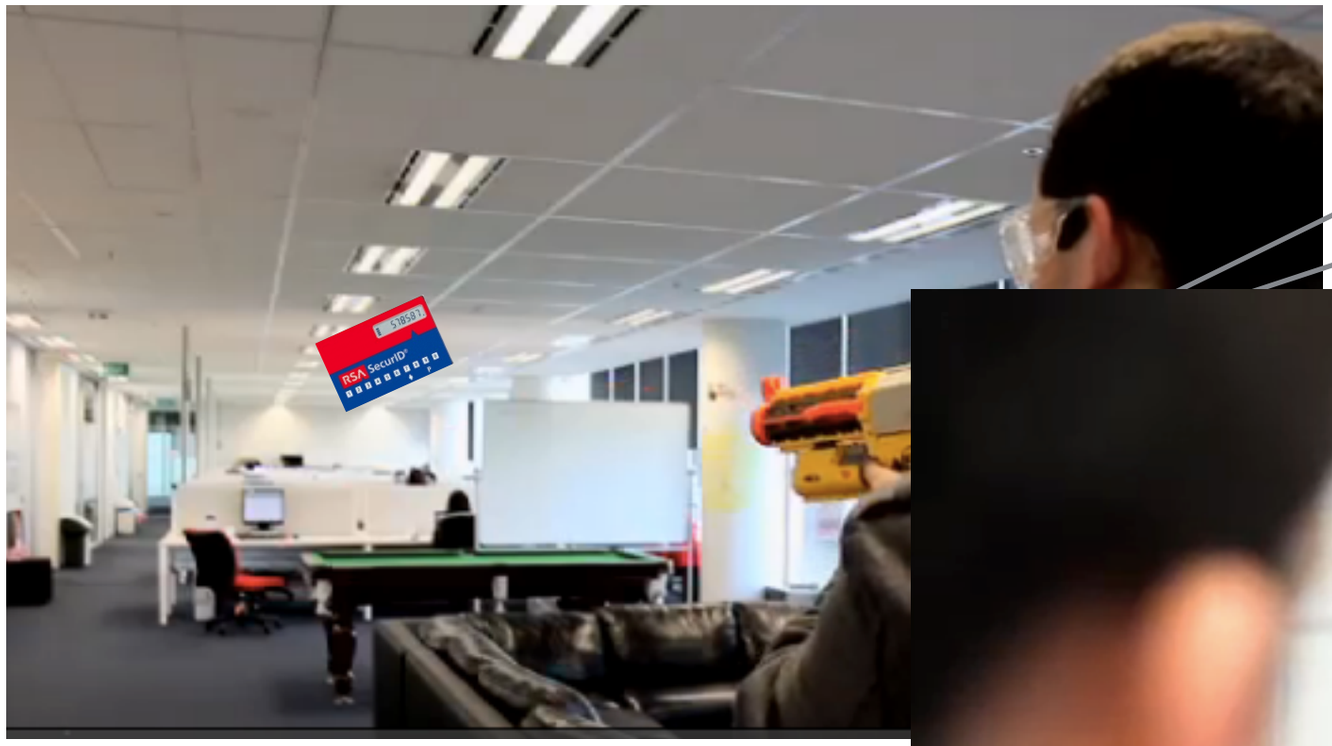
Authentication tokens are still problematic

- Man-in-the-middle attacks
  - Phishing, malware, social engineering can all capture at least one passcode
  - So Eve can impersonate Alice at least once



Authentication tokens are still problematic

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



Replacing clay pigeo

Testing friends' psychic abilities

Using for game of go-fish

Authentication tokens are *still* problematic

- Lost, forgotten, or broken tokens
  - Credential recovery problem
  - Back to the name of your favorite pet…

Authentication tokens still have problems

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

In-class exercise

- Suppose challenge-response mode with 5 challenges
- What's the probability of Mallory successfully impersonating Alice (on one try) before Alice ever authenticates?
- With what probability can Mallory impersonate Alice after eavesdropping on 7 authentication sessions? After 10? After 14? (*Assume that challenges are never repeated*.)

In-class exercise

- What's the probability of Mallory successfully impersonating Alice (on one try) before Alice ever authenticates?
  - $1/36^5$
- With what probability can Mallory impersonate Alice after eavesdropping on 7 authentication sessions? After 10? After 14? (Assume that challenges are never repeated.)
  - $1/36^5$
  - **But observe that after 14 sessions, the card is used up!**

In-class exercise

- Suppose the Mallory has learned half the values on the card via eavesdropping. With what probability can Mallory impersonate Alice assuming that challenges are now generated uniformly at random?

Authentication tokens still have problems

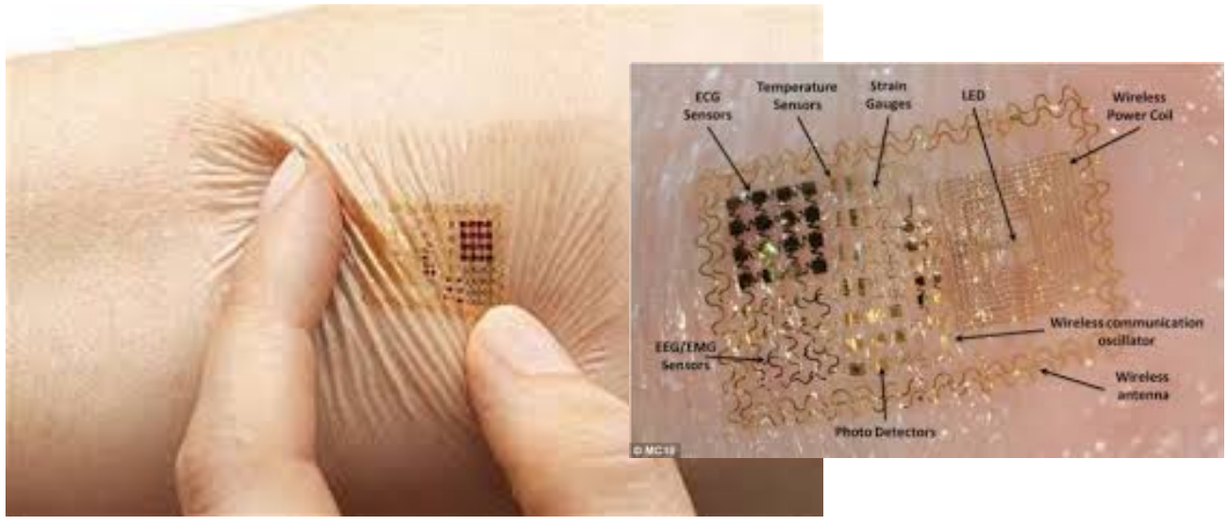
- Passcodes on mobile devices
  - Mobile devices are vulnerable to malware
  - SMS sometimes used; can be compromised in other ways
  - Consumers often don’t activate when it’s optional

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.

Yubikey

- Offered as a FIDO U2F token
- Pros:
  - No typing
    - Plugs into USB; touch activation
    - (Some models) activate via NFC with mobile devices
  - Public-key cryptography supported (some models)
    - Resists man-in-the middle attacks

Yubikey

- Cons:
  - Lost / broken token ➔ backup authentication problem
  - Bootstrapping: Who's going to distribute / pay for these things?
    - $18+
  - Who wants to carry yet another device?

Is authentication the killer app for smartwatches?

Remember from last lecture:

- Biometrics
- Wireless communication
  - (No passcode typing)
  - Can eliminate attacks such as man-in-the-middle
  - NFC interface for payments
- Always with you

Is authentication the killer app for smartwatches?

SALON

WEDNESDAY, SEP 10, 2014 01:43 PM EDT

A killer app for the Apple Watch: Gun control