Honey

Thanks to Ari Juels for most of this deck!

In 1943, German intelligence made a major discovery

A Spanish fisherman discovered a body washed ashore.

- *Body was that of British Royal Marine Capt. (Maj.) William 'Bill' H.N. Martin*
- Spain was neutral, but…
- A German agent in a town nearby got wind of the discovery.
- Martin was hand carrying a letter…

In 1943, German intelligence made a major discovery

A Spanish fisherman discovered a body washed ashore.

- The Germans knew the Allies' planned a major invasion, but not *where*.
- Martin's letter referred to a plan for General 'Jumbo' Wilson to invade *Greece*.
- On Hitler's order, the Germans deployed three Panzer divisions in Greece to meet the attack.

What happened?

- The Allies invaded ***Sicily***.
- Captain Martin never existed. He was a plant.
- The British went to extraordinary lengths to fabricate Martin, e.g.,
  - Found corpse of homeless man with fluid in lungs consistent with drowning
  - Chose plausibly remote location with German agent
  - Fabricated letter from Martin’s ‘father,’ love letters (“What are those horrible dark hints… about being sent off…?”), bill for engagement ring, photo of ‘fiancée’, etc., etc.

Operation Mincemeat

- Operation Mincemeat saved an estimated 40,000 Allied lives.
- It also gave rise to a movie… *The Man Who Never Was*

“In wartime, truth is so precious that she should always be attended by a bodyguard of lies.”

–Winston Churchill

Decoys

- Decoys are fake objects designed for deceit to look real.
- Examples:
  - Inflatable tanks and fighter jets
  - Bait money
- Various objectives:
  - Guide attackers away from real objectives
  - Learn about attackers' behavior
  - Detect stealthy attacks

Decoys

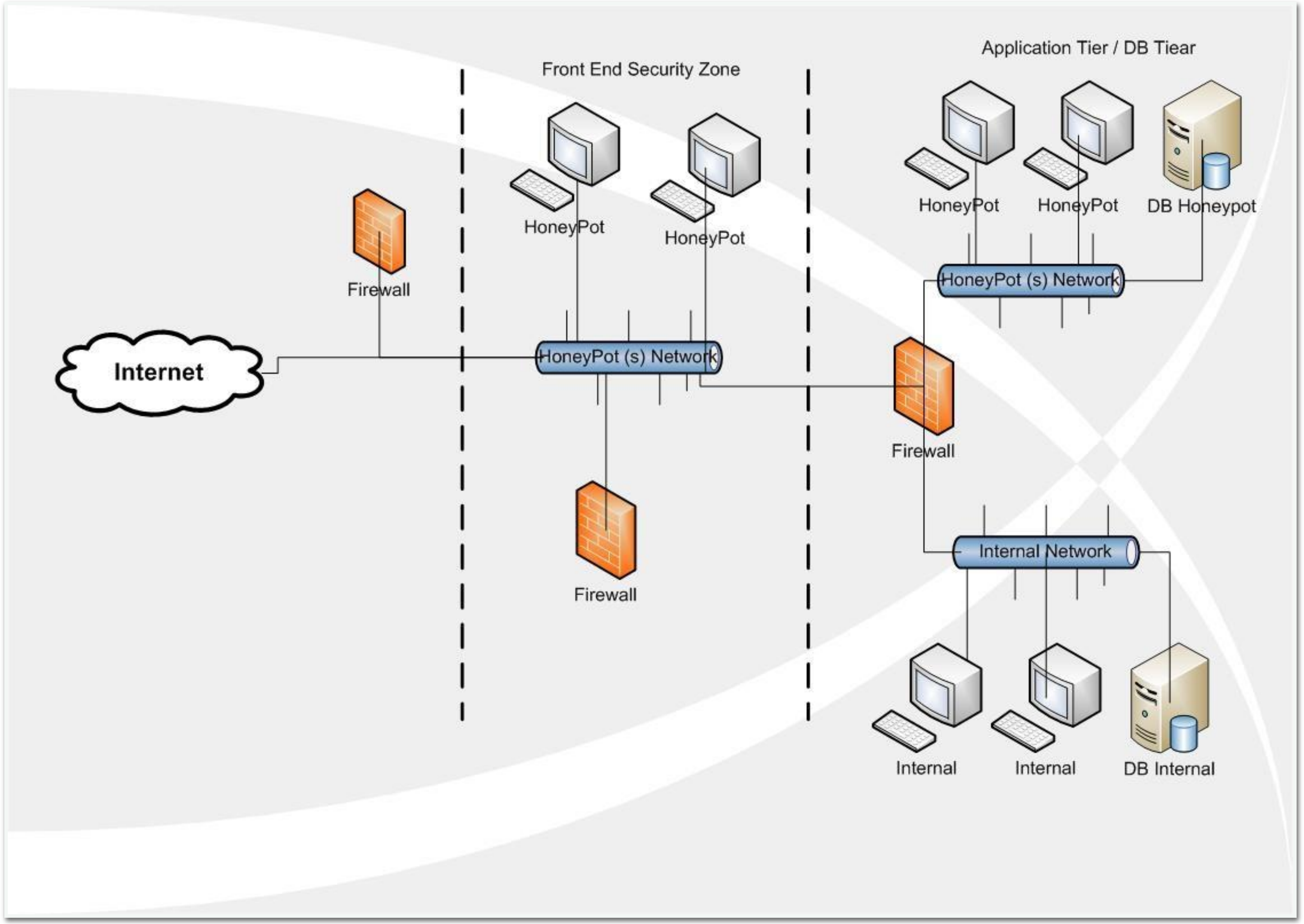
- Where were first decoys deployed?

South African speckled emperor moth
National Geographic, Aug. 2006

In computer security, we have “honey objects”

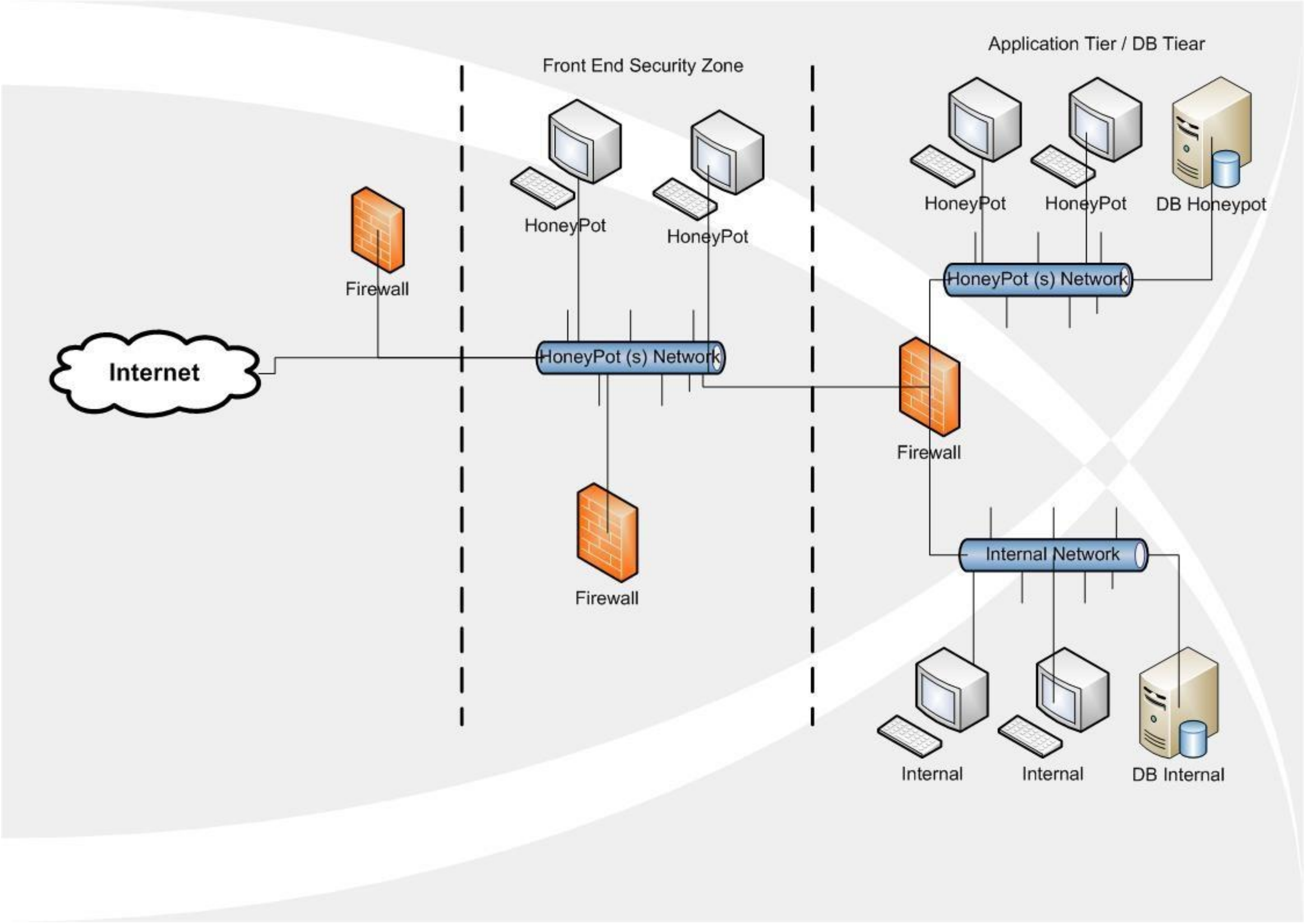
Honeypots

- Servers set up to lure attackers for observation
- What might you learn?
  - Detect specific attack
    - E.g., database honeypot looks for SQL injection attacks
    - (Basic firewalls don't protect against such application-level attacks.)
  - Understand intruder tactics
    - What resources is the adversary looking for?
    - Where is the attack originating? What's the vector of attack?



Honeypots

- Honeypots are counter-intelligence
- An adversary that detects honeypots can bypass them or show false behavior
- Counter-counter-counter-intelligence
  - So… set up some honeypots that actually look like honeypots
    - E.g., Port 365 claimed by Deception Toolkit (DTK)
  - Adversary may then think he/she has found the real honeypots when he/she hasn't… or may just back off



Honeytokens

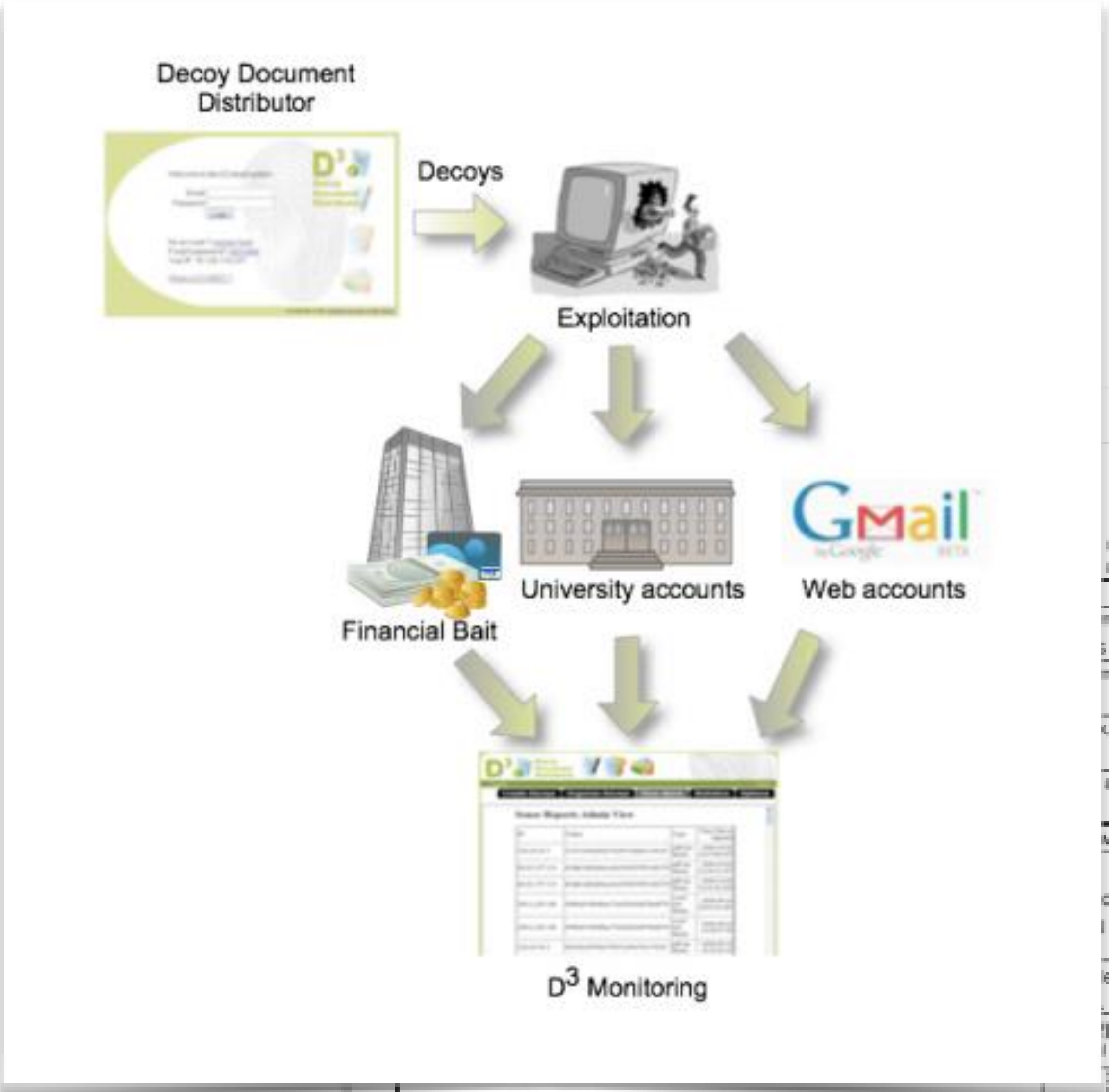
- Help detect breaches or other forms of compromise.
- Example: Lace a credit-card database with fakes.

`Nemo Nemosious MC 5466 1602 8888 8888 exp: 05/2017 CVV: 913`

- If a Nemo Nemosiosis transaction turns up, you know the database has been breached.
- Not totally straightforward. Why?

Decoy documents

- B. M. Bowen, S. Hershkop, A. D. Keromytis, S. J. Stolfo: Baiting Inside Attackers Using Decoy Documents. SecureComm, pp. 51-70, 2009.
- Help detect *insider attacks*
- Fake documents deployed in real user settings



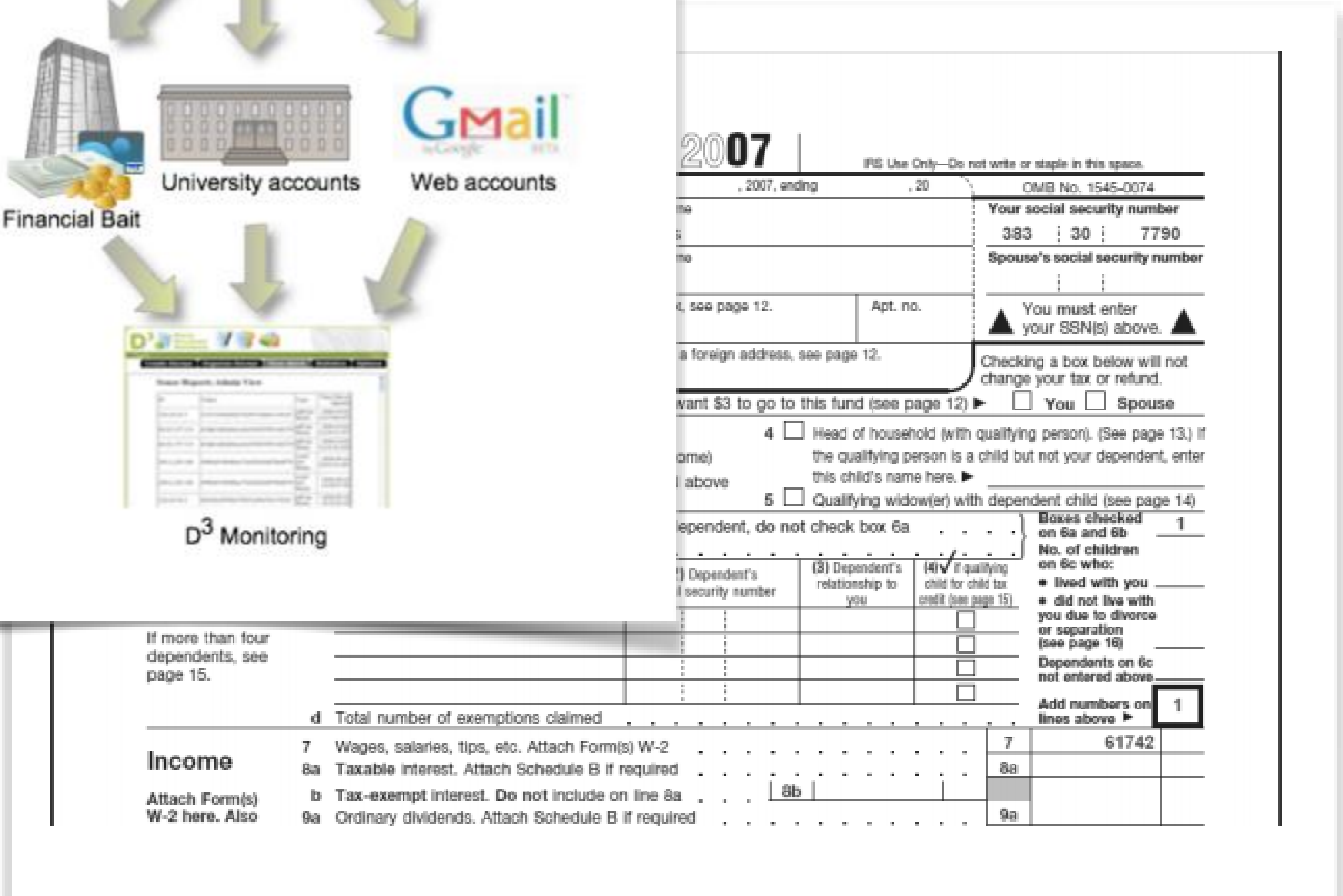
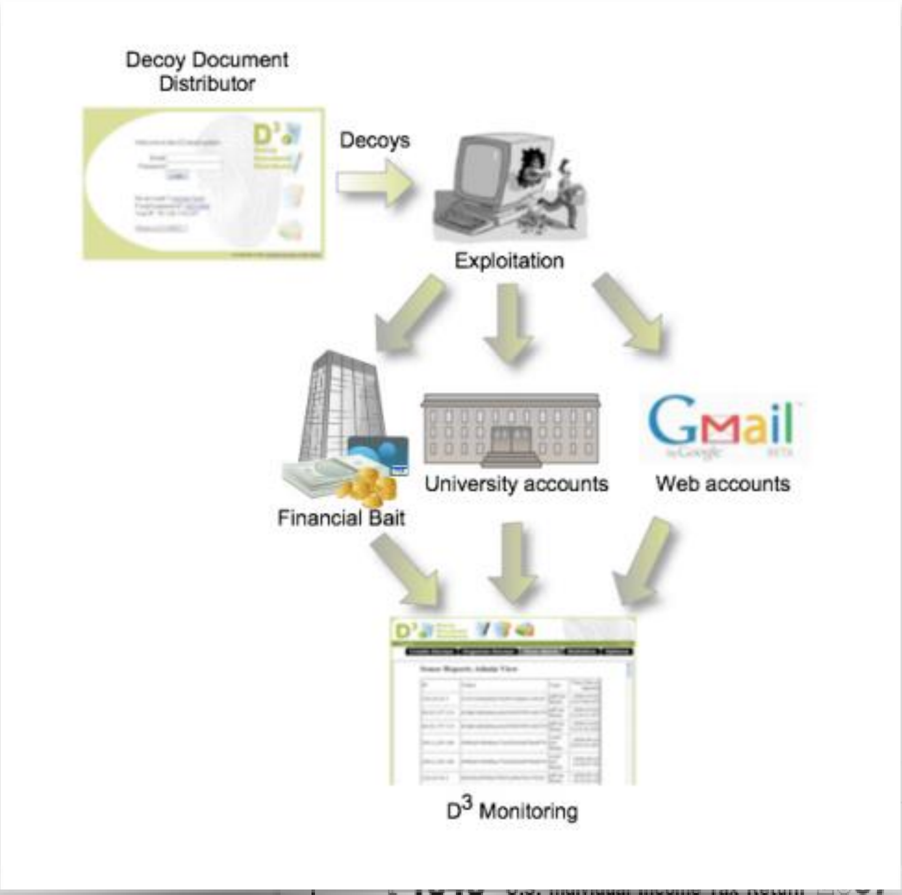
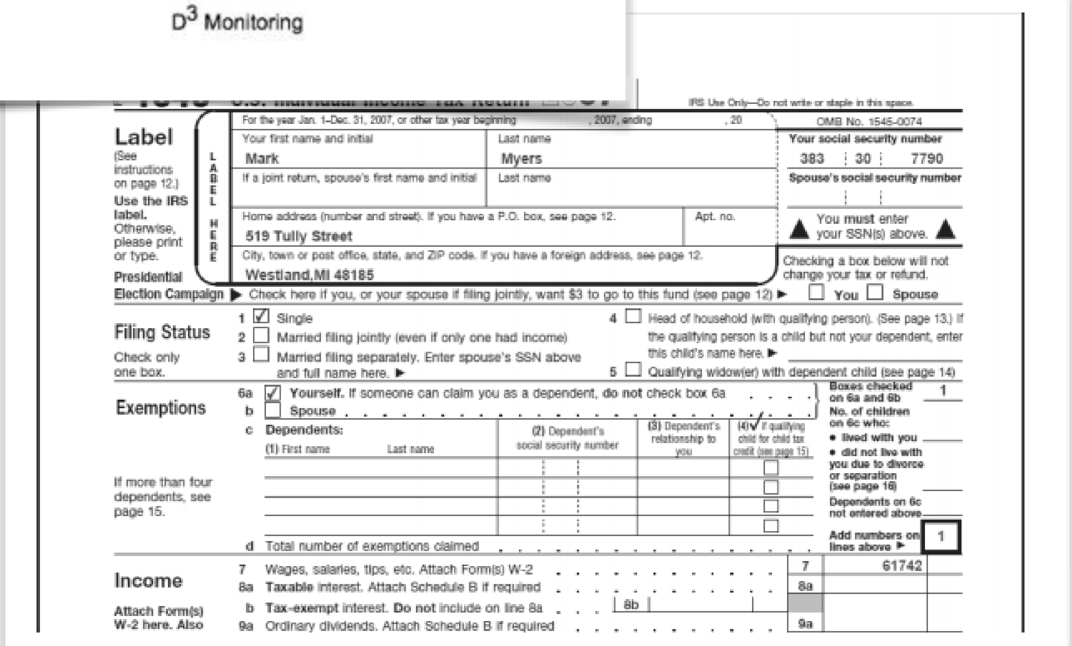


Fig. 3. Decoy tax document with bogus user information.

Decoy documents

- Detection via
  - Egress monitoring
  - Embedded “beacon”
  - Honeytokens
- Challenge: Non-interference / false positives
- Claim: Decoys can be created that are highly believable but have low interference (with normal activity of user)





IRS Use Only—Do not write or staple in this space.

For the year Jan. 1–Dec. 31, 2007, or other tax year beginning , 2007, ending , 20

OMB No. 1545-0074

Label (See instructions on page 12.) **Use the IRS label.** Otherwise, please print or type.

LABEL HERE

Your first name and initial: Mark — Last name: Myers

Your social security number: 383 | 30 | 7790

If a joint return, spouse's first name and initial — Last name

Spouse's social security number

Home address (number and street). If you have a P.O. box, see page 12. — Apt. no.

519 Tully Street

▲ You must enter your SSN(s) above. ▲

City, town or post office, state, and ZIP code. If you have a foreign address, see page 12.

Westland,MI 48185

Checking a box below will not change your tax or refund.

Presidential Election Campaign ▶ Check here if you, or your spouse if filing jointly, want $3 to go to this fund (see page 12) ▶ ☐ You ☐ Spouse

Filing Status Check only one box.

1 ☑ Single
2 ☐ Married filing jointly (even if only one had income)
3 ☐ Married filing separately. Enter spouse's SSN above and full name here. ▶
4 ☐ Head of household (with qualifying person). (See page 13.) If the qualifying person is a child but not your dependent, enter this child's name here. ▶
5 ☐ Qualifying widow(er) with dependent child (see page 14)

Exemptions

6a ☑ **Yourself.** If someone can claim you as a dependent, **do not** check box 6a
b ☐ Spouse
c **Dependents:** (1) First name, Last name; (2) Dependent's social security number; (3) Dependent's relationship to you; (4) ✓ if qualifying child for child tax credit (see page 15)

If more than four dependents, see page 15.

d Total number of exemptions claimed

Boxes checked on 6a and 6b: 1
No. of children on 6c who:
• lived with you
• did not live with you due to divorce or separation (see page 16)
Dependents on 6c not entered above
Add numbers on lines above ▶ 1

Income

Attach Form(s) W-2 here. Also

7 Wages, salaries, tips, etc. Attach Form(s) W-2 — 7 — 61742
8a **Taxable** interest. Attach Schedule B if required — 8a
b **Tax-exempt** interest. **Do not** include on line 8a — 8b
9a Ordinary dividends. Attach Schedule B if required — 9a

Fig. 3. Decoy tax document with bogus user information.

Honeywords

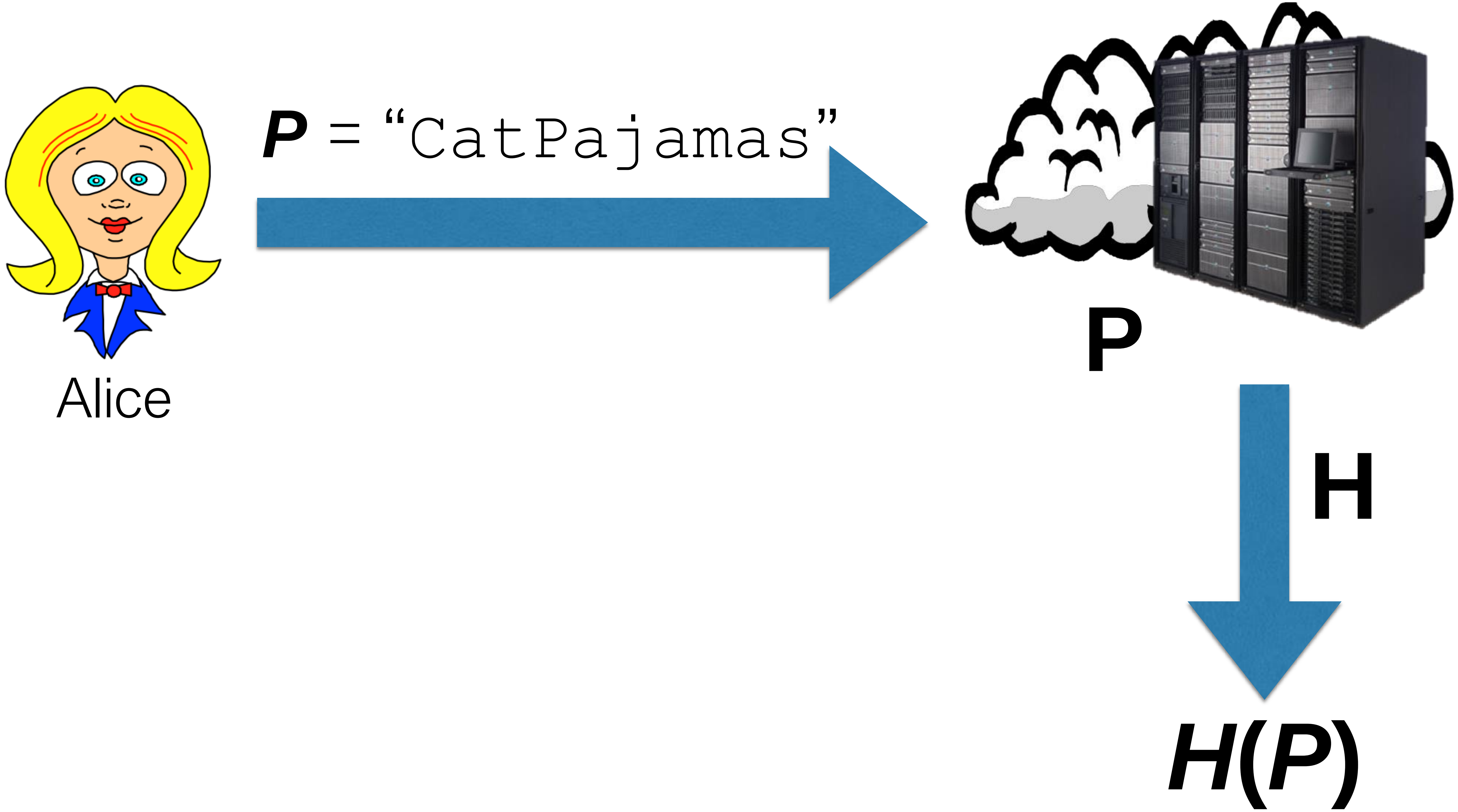
A. Juels and R. Rivest. Honeywords: Making Password Cracking Detectable. ACM CCS, 2013.

Good news and bad about password breaches

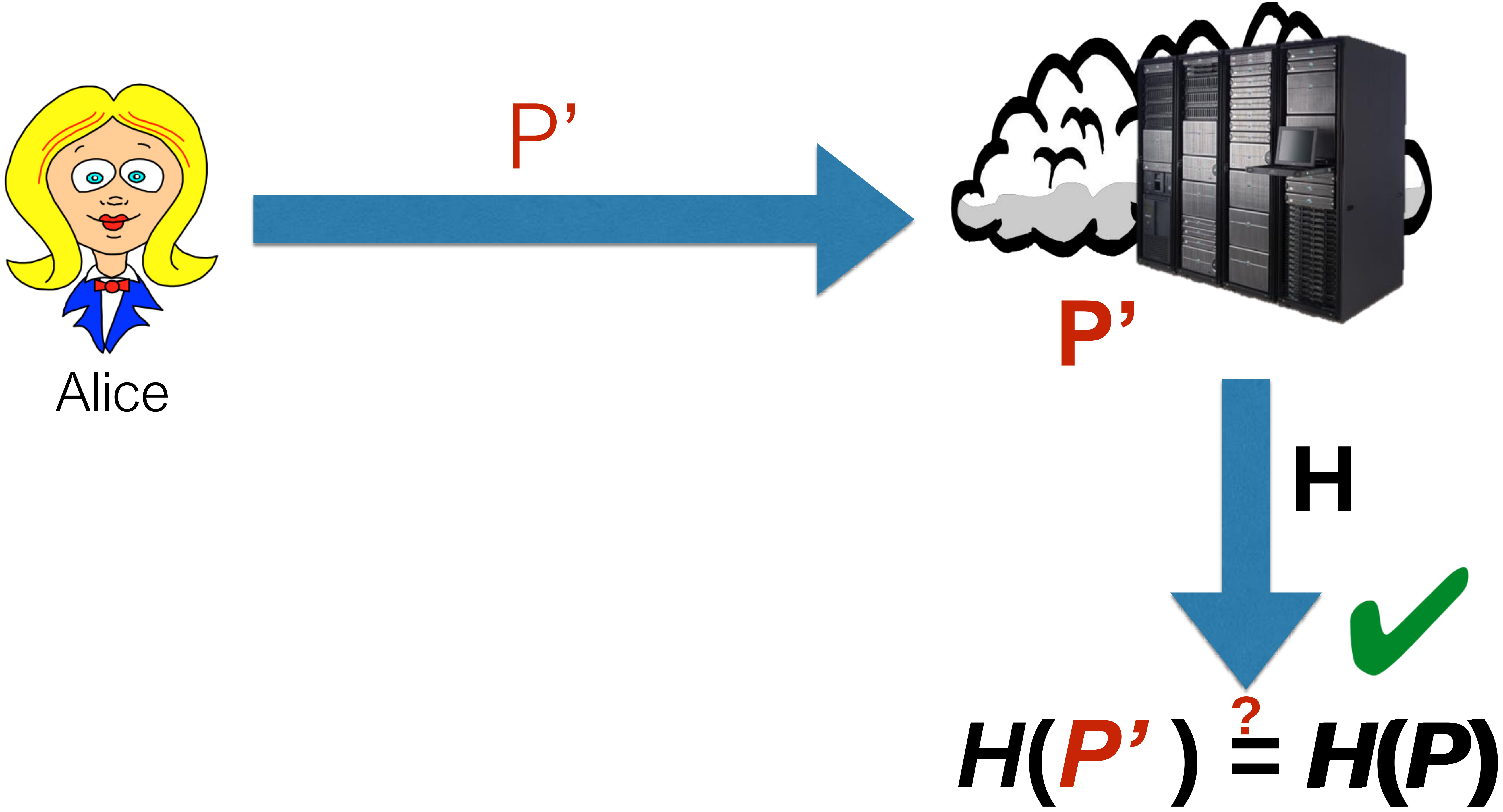
- The good news: Whenever you want to talk about password (or PII) breaches, there are very good, recent examples.

- The bad news: This is all bad news.

Reminder: Passwords are generally protected via hashing



To verify an incoming password…



Recall: Password hashing

- Hashing (plus salting) forces an attacker that learns hashes to determine passwords by brute-force (offline) guessing
- Brute-force guessing means the attacker repeatedly makes a guess *P’* and checks if H(*P’*) = H(*P*)
- Additionally, hashing can be hardened (slowed) in various ways (e.g. bcrypt)
- This all seems good, but…

Password hashing

- Remember: real passwords are weak and easily guessed.
  - Guessing probability (GP) in RockYou was 0.9%
  - Consistent across studies, e.g., Bonneau’s 69+ million Yahoo! password study was 1.08%
- Even good (& salted) hashes are often inadequate.
- Let’s just assume that hashes can be cracked and passwords are effectively in the clear.

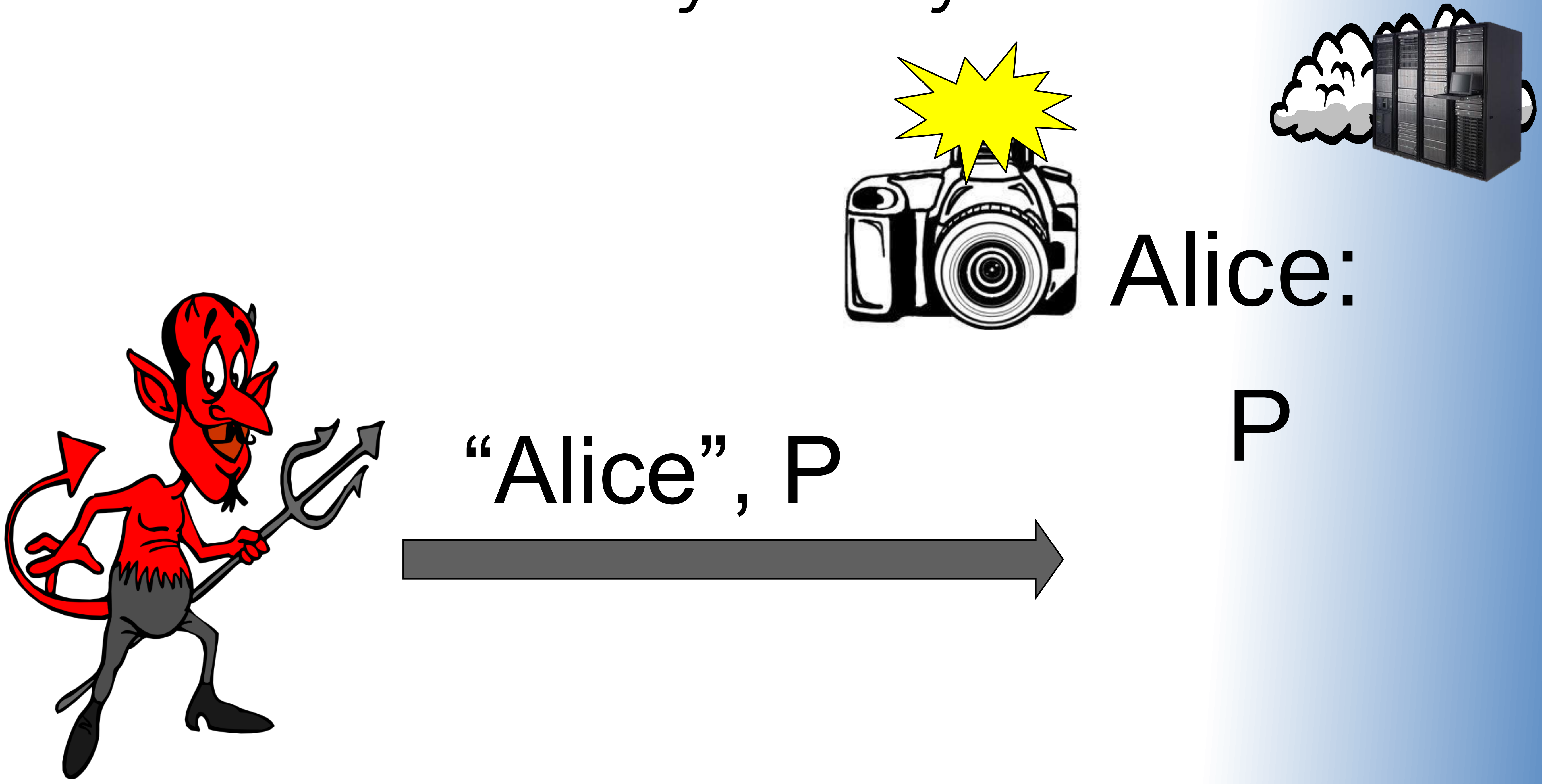
Adversarial model

- "Smash-and-grab" attack
  - The adversary compromises the system ephemerally (usually passively).
- The adversary:
  - Steals a snapshot of password file;
  - Impersonate user(s)

Alice:

P

Adversary always wins



Honeywords

Alice:

P_1

P_2

…

P_n

Honeywords

Alice:

P_1

P_2

…

True password → $P_i = P$

…

P_n

Honeywords

Alice:

P_1

P_2

…

$\boldsymbol{P_i = P}$

…

P_n

Honeywords (decoys)

Honeywords

Alice:

Sweetwords

P_1

P_2

…

P_n

The adversarial game

What is i?

“Alice”, P_j

Alice:

P_1

P_2

…

P_n

Given ideal honeywords, the attacker will guess correctly, $j = i$, with (small!) probability, about $1/n$.

The adversarial game

Which is the (real) password?

Alice:

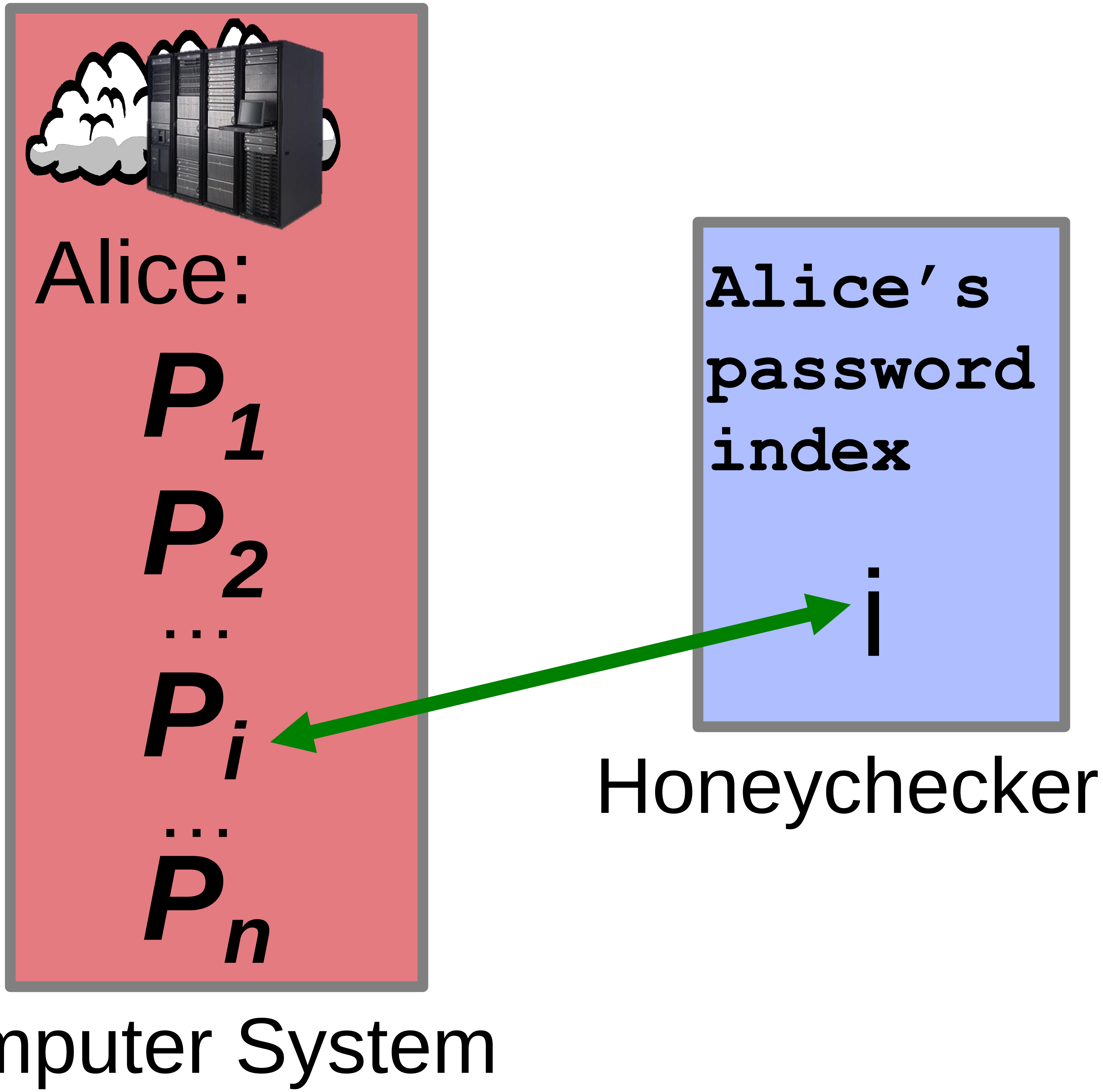
- 5512lockerno.
- tribal_3
- cshcsh.meowr.18
- 28/07/89rm
- anto_2001_jesu
- CRFRALAASS$4
- !v0nn3
- ponk.m4t

Honeyword design questions

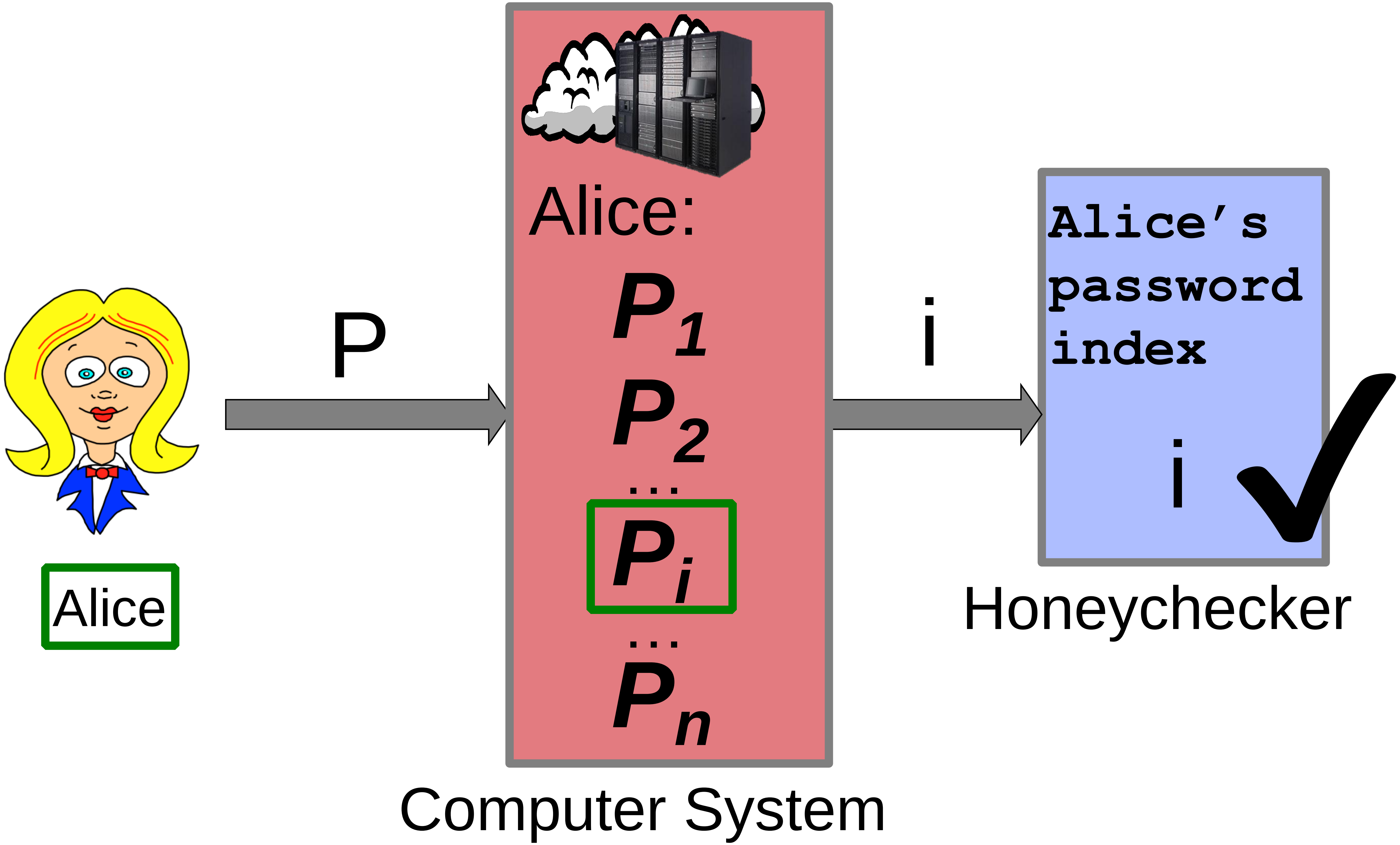
1. Verification: How does the system check whether a submitted password P' is the true password P_i?
   - How is index i verified without storing i alongside passwords?
2. Generation: How are honeywords generated?
   - How do we make bogus passwords look real?

(Many other design questions, e.g., how to respond when breach is detected using honeywords…)

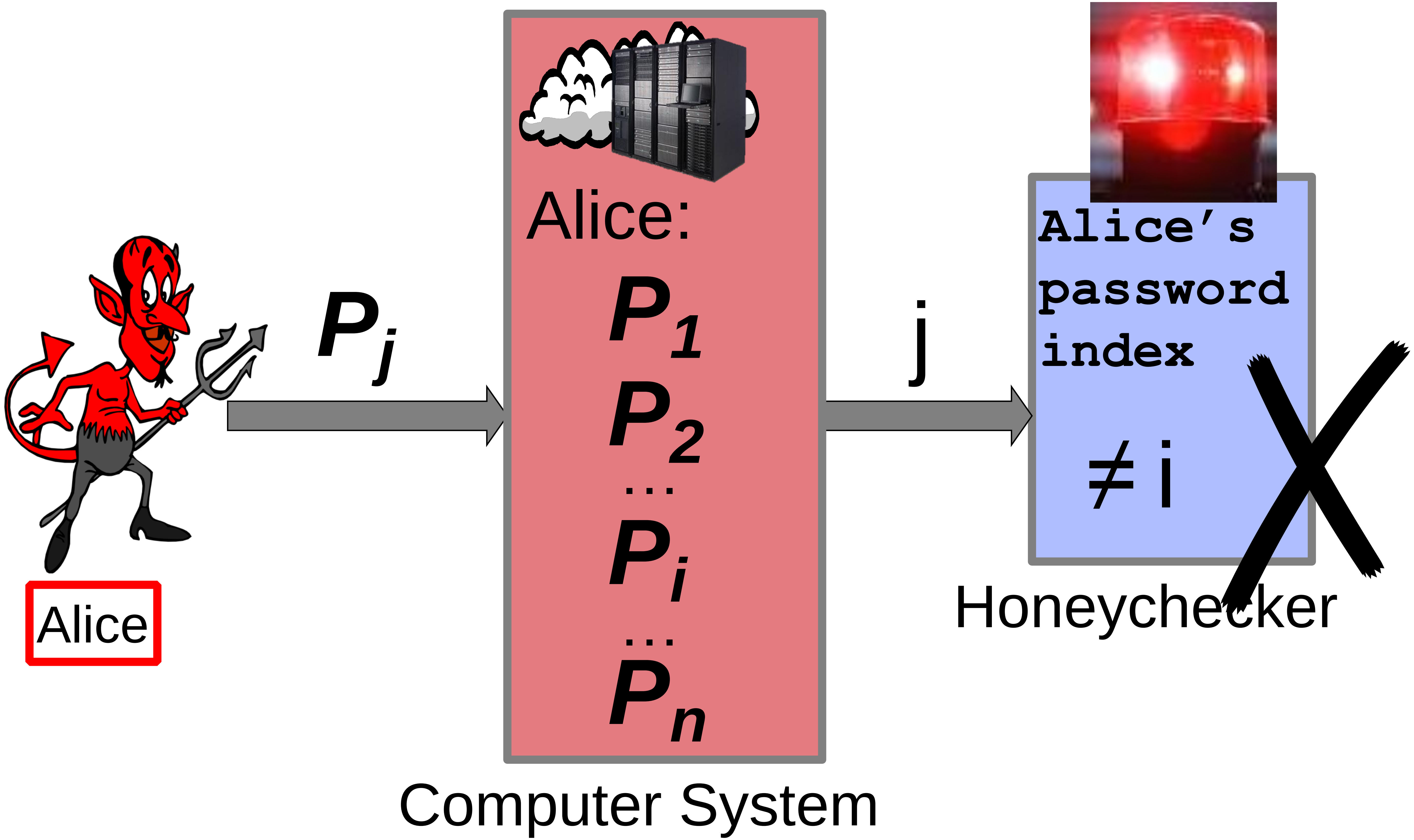
Honeywords: Verification



Honeywords: Verification



Honeywords: Verification

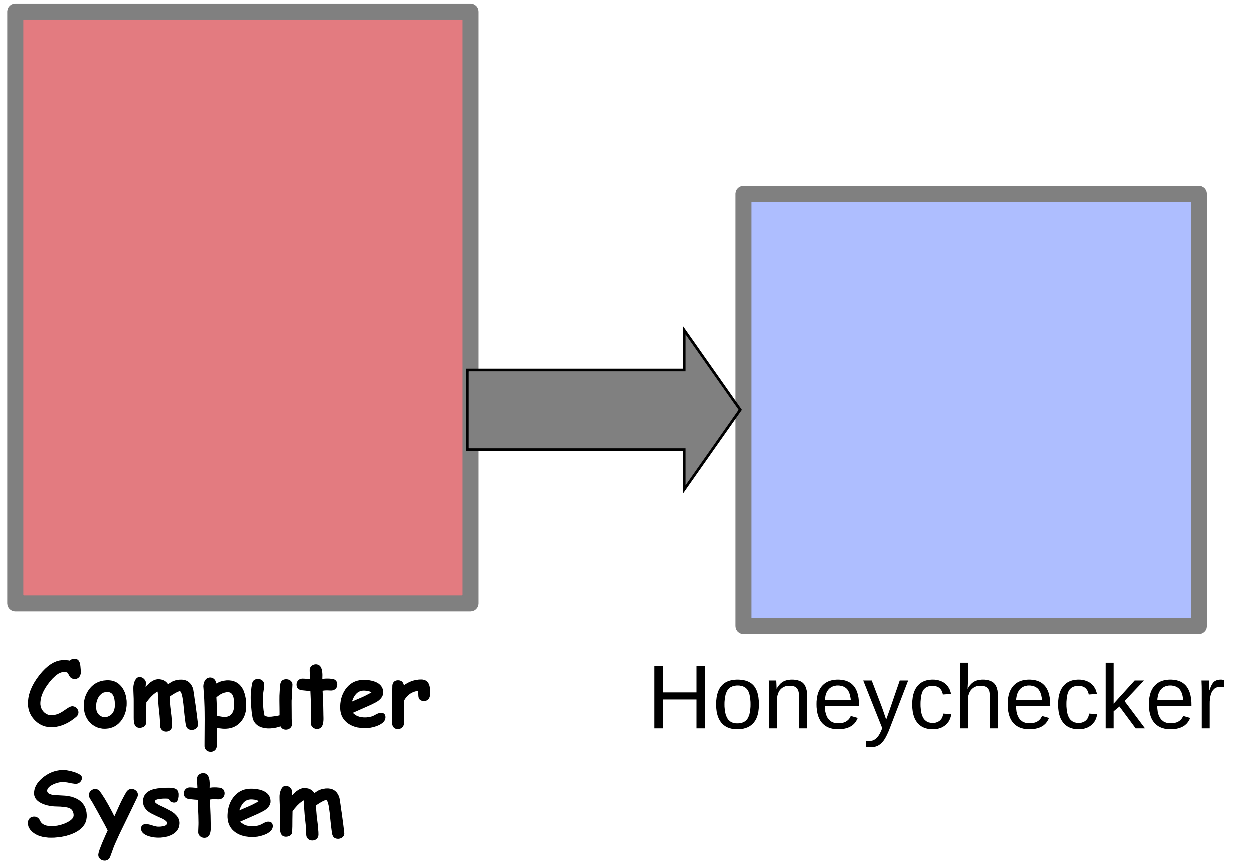


Honeywords: Verification Rule

- If the true password P_i is submitted, the user is authenticated.
- If a password P' $\notin \{P_1 \ldots P_n\}$ is submitted, it's treated as a normal password authentication failure.
- If a honeyword $P_j \neq P_i$ is submitted, an alarm is raised by the honeychecker.
  - This is likely to happen only after a breach!
  - Honeywords (if properly chosen) will rarely be submitted otherwise.
- Note: No change in the user experience!

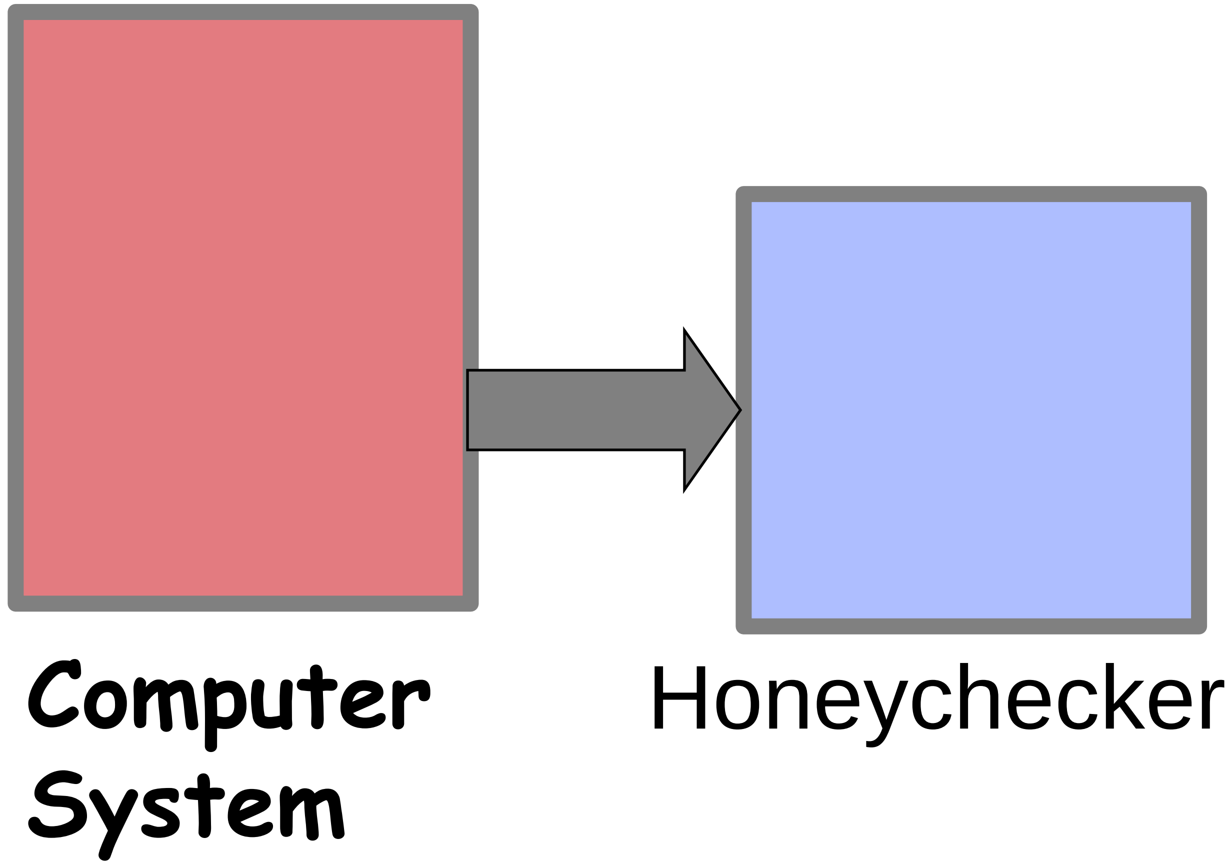
Some nice features of this design

- Computer system does nothing but transmit sweetword index j
  - Little modification needed
- We get the benefits of distributed security
  - Compromise of either component isn't fatal
  - No single point of compromise
  - Compromise of both brings us back to hashed case
- Honeychecker can be minimalist, (nearly) input-only
  - Only (rare) output is alarm



Some nice features of this design

- Honeychecker can be offline
  - E.g., honeychecker sits downstream in security operations center (SOC)
  - Not active in authentication itself, but gives rapid alert in case of breach
  - If honeychecker goes down, users can still authenticate



Honeyword generation

Which is Alice's real password?

Alice:
• **QrMdmkQt**
• **AP9LXEEa**
• **m7xnQVV4**
• **kingeloi**
• **y5BJKWhA**

Honeyword generation: "Chaffing with a password model"

- Password-hash crackers learn model from lexicon of breached passwords (e.g., RockYou database)
  - Make guesses from model probability distribution
- Idea: Repurpose cracker as generator!
- Simple (splicing) generator yields…

Alice:

- qivole
- paloma
- 123asdf
- Compaq
- asdfway

But there are problem cases…

Which is Alice's real password?

Alice:
• hi4allaspls
• #1spongebobsmymansodonttouchhim
• Travis46
• #1bruinn
• KJGS^!*ss

Honeyword generation: Chaffing by tweaking

- [ZMR10] observed users tweak passwords during reset (e.g., HardPassword1, HardPassword2, …)
  - Proposed tweak-based cracker
- Idea: Tweak password to generate honeywords!
- E.g., tweak numbers in true password…

Alice:

- **yamahapacificer32145678987654321**
- **yamahapacificer12345678987654321**
- **yamahapacificer12345678901234567**
- **yamahapacificer62145678987654322**

Honeyword generation: A research challenge

- Blink-182 is a rock band
- This password is semantically significant
  - Tweaking would break it
  - Generation is unlikely to yield it
- Dealing with such passwords is a special challenge—relates to natural language processing
- Subject of an upcoming paper…
  - E.g., use other people's passwords as honeywords…

Alice:

- Blink123
- Graph128
- Froggy%71
- Blink182
- Froggy!83

How good does honeyword generation have to be?

- Let U be a probability distribution on user password selection
  - i.e., user chooses P w.p. $U(P)$
- Let G be a distribution on honeyword generation
  - i.e., honeyword P generated w.p. $G(P)$
- Given list $P_1, \ldots, P_n$, adversary's optimal strategy is to guess P_j that maximizes $U(P_j) / G(P_j)$
- Thus, given chaffing-with-a-password-model, a particularly dangerous password is, e.g.:

#1spongebobsmymansodonttouchhim

How good does honeyword generation have to be?

- We might imagine practical choice of, say,
  - *n* = 20
- With a “flat” honeyword distribution, U $\approx$G, adversary hits a honeyword w.p. 95%
- Perfect flatness isn’t required
- Even if adversary can rule out all but two sweetwords, we can still detect a breach systemically with high probability
  – E.g., 50% guessing success means prob. 2^{-m} of compromising m accounts without detection

How good does honeyword generation have to be?

- Generation strategies can be hybridized as a hedge against failure of one strategy, e.g.,

• **qivole!**	• **qivole#**
• **123asdf**	• **111asdf**
• **IBetNSACantCrackThisPassword89**	• **IBetNSACantCrackThisPassword12**
• **Froggy%71**	• **Froggy!88**

?

Takeaways

- Deception is an age-old tactic
  - Pioneered by Mother Nature
- It is very useful in computer security
  - Honeypots, honeytokens, honeywords, honey encryption
- You'll get to play with it for the next month…