SAuth: protecting user accounts from password database leaks

Elias Athanasopoulos

https://srec.cs.ucy.ac.cy

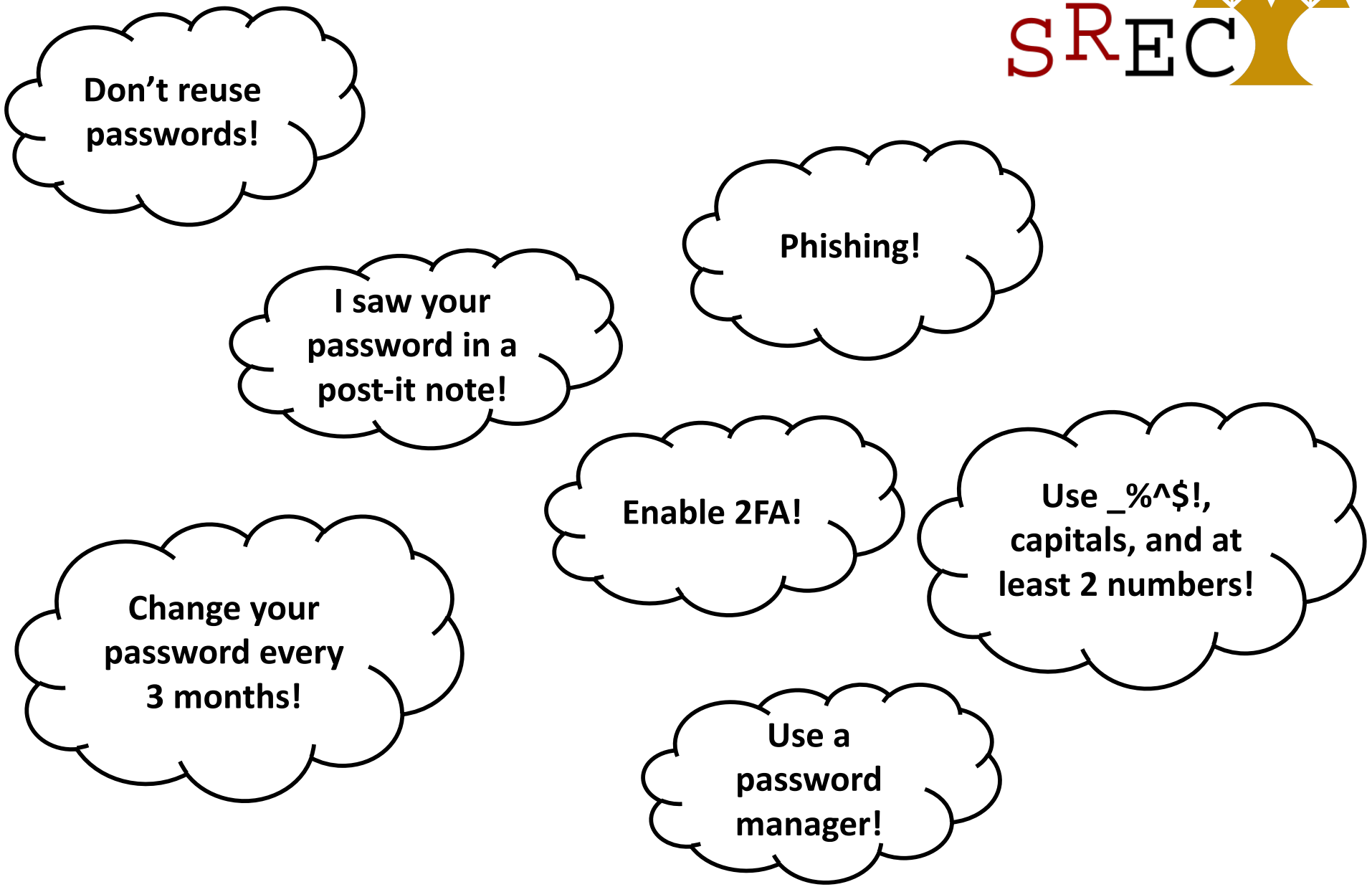
University of Cyprus

Who am I?

- University of Cyprus - Computer Science
  - Assistant professor
- SREC (Security Research) group
  - 4 PhD student
  - Several ugrads
  - Research theme is System Security and Privacy
  - More: https://srec.cs.ucy.ac.cy

University of Cyprus

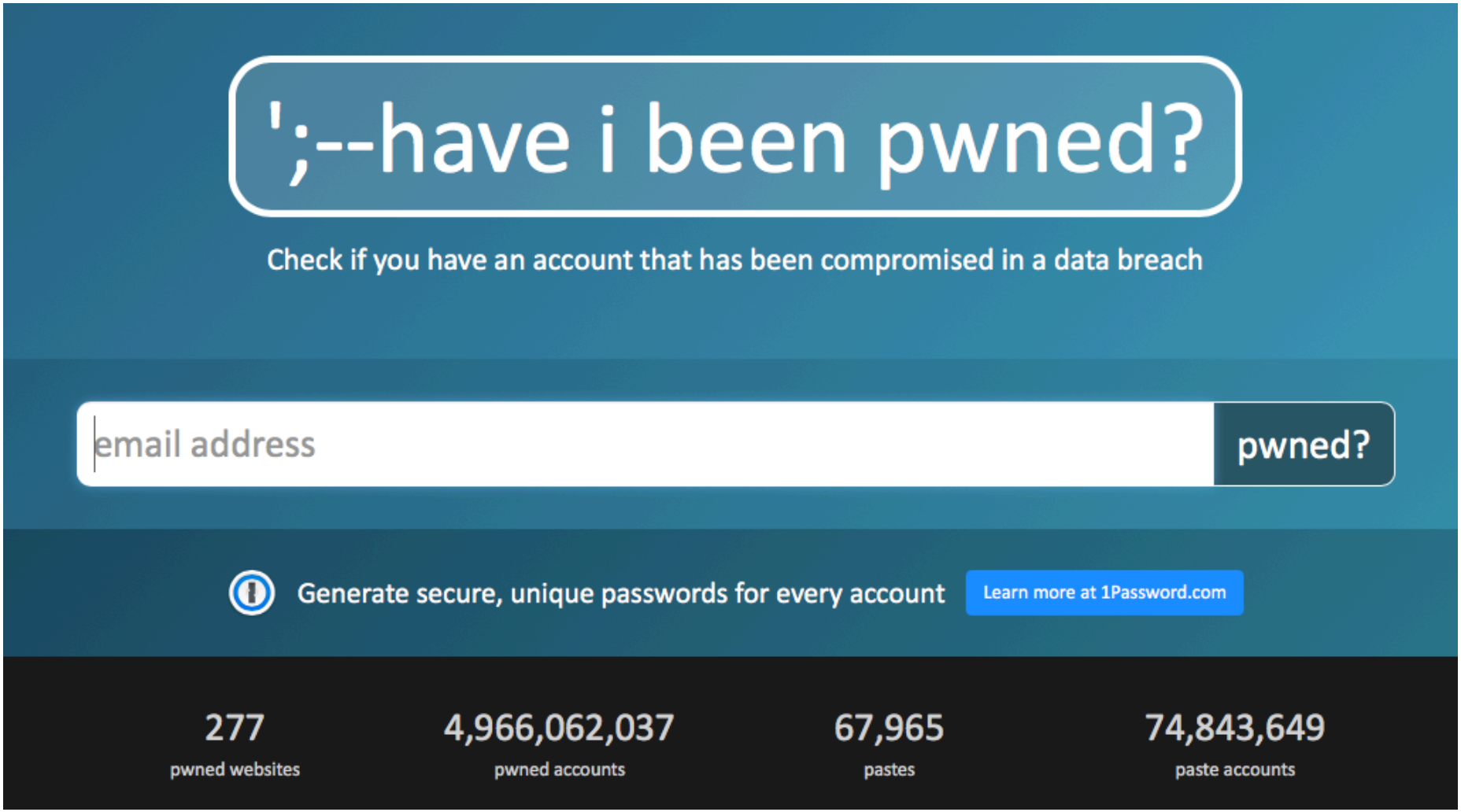
YET, ANOTHER PASSWORDS TALK


Don’t reuse passwords!
I saw your password in a post-it note!
Phishing!
Enable 2FA!
Use _%^$!, capitals, and at least 2 numbers!
Change your password every 3 months!
Use a password manager!


"Nobody gets hacked. To get hacked you need somebody with 197 IQ and he needs about 15 percent of your password."
OF AMERICA
C-SPAN
c-span.org
0:01 1.5M views
From Ira Goldman
12:28 AM · Oct 20, 2020 · Twitter Web App

YET, ANOTHER PASSWORDS TALK, BUT NOT A BLAME-THE-USERS TALK

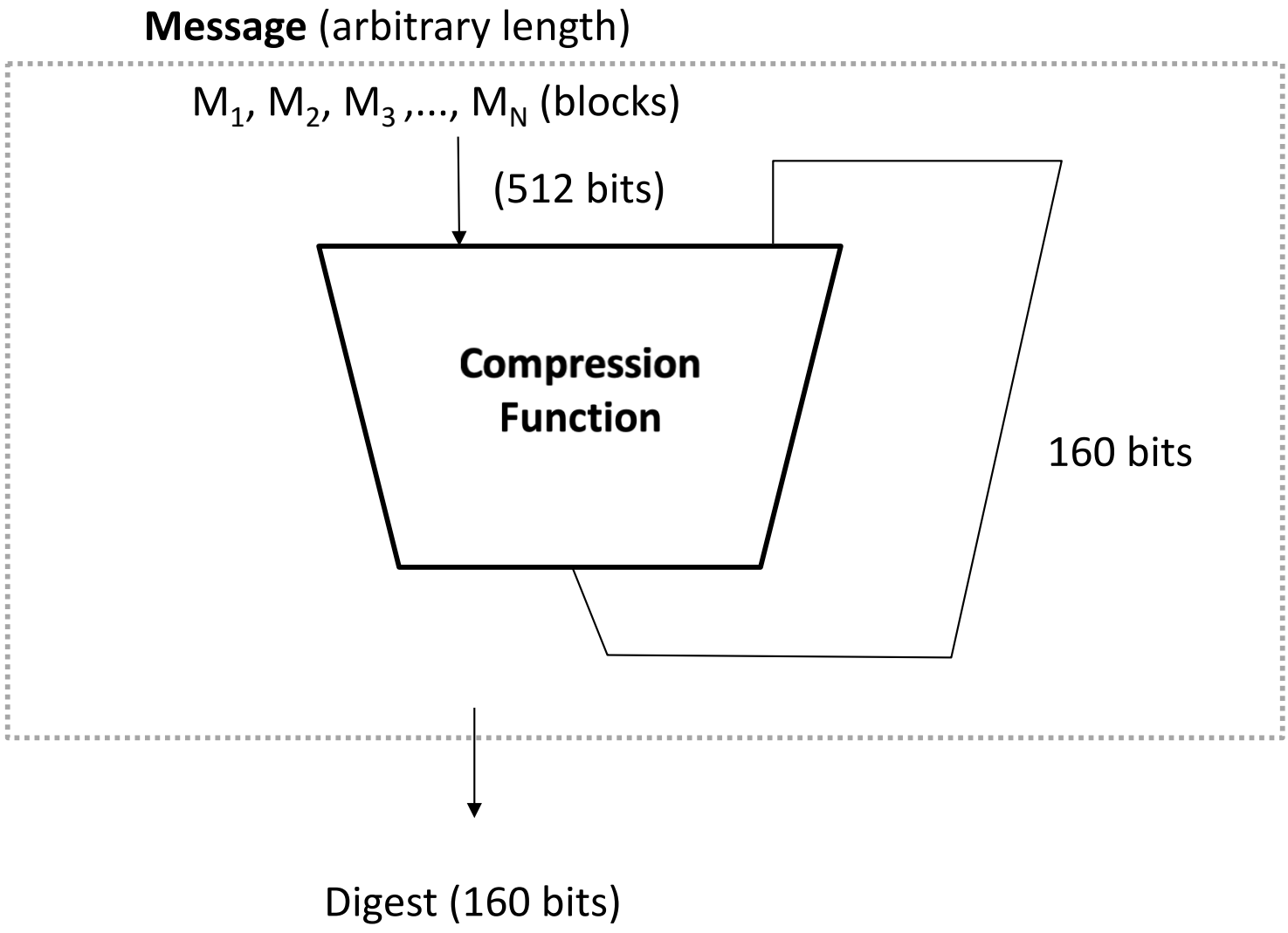
Blame the services!



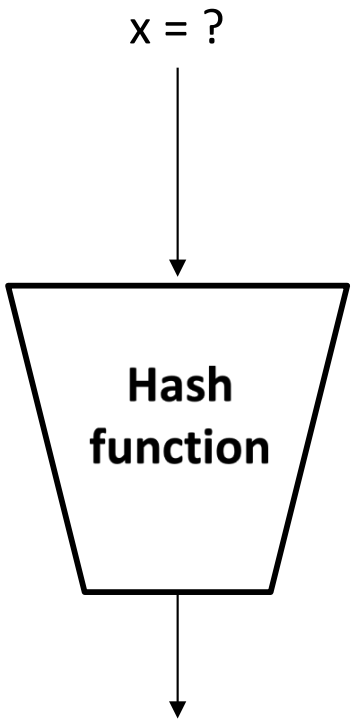
What can go wrong?

HOW TO PROTECT PASSWORDS?

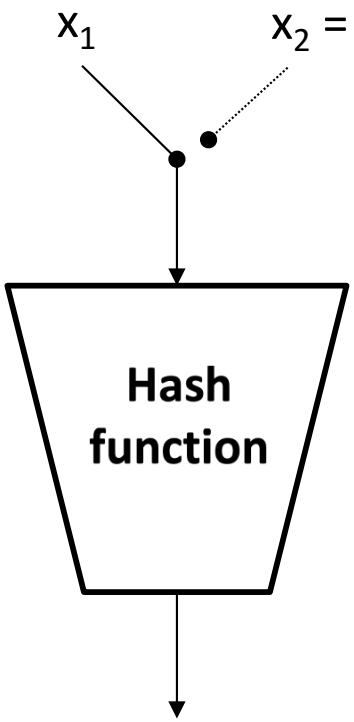
Cryptographic Hash Function



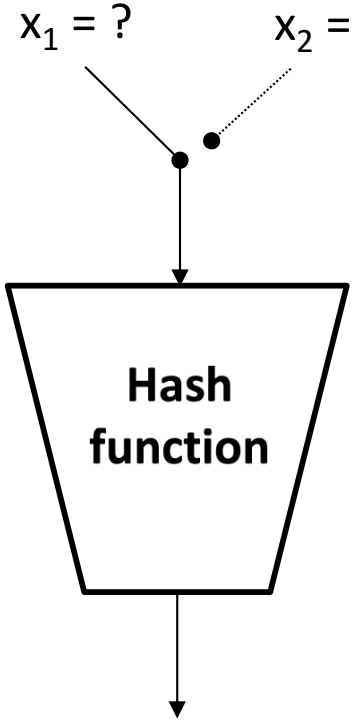
Basic Requirements



h(x)
preimage resistance



$h(x_1) = h(x_2)$
second preimage resistance



$h(x_1) = h(x_2)$
collision resistance

Threat Model

- Text-based passwords are stored in databases
- They are stored cryptographically hashed
- Databases can be leaked
- Cryptographic hashes can be cracked
- Other Interesting threat models
  - Phishing and social engineering attacks

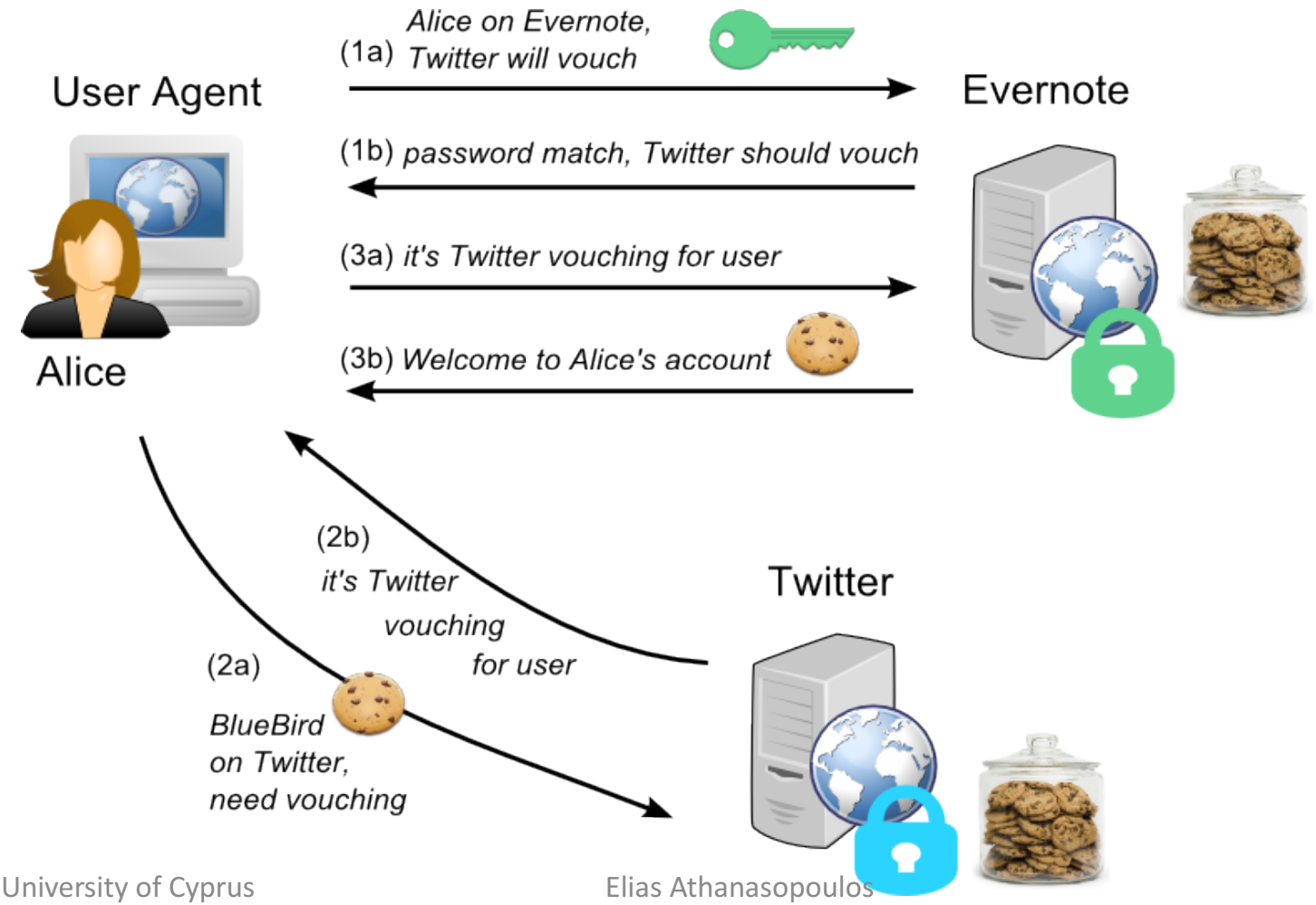
Can we harden authentication and keep the ”password” interface?

SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*

SAuth: Synergy-based Enhanced Authentication

- *We propose: cooperating sites pool authentication resources*



Why it works?

- Password reminders utilize vouching when the password is lost
- SAuth implicitly initiates such a password reminder on the background
- SAuth does this by pairing arbitrary services

Password Reuse Woes

Stolen passwords re-used to attack Best Buy accounts

Summary: *Customer re-use of the same user name and password across multiple sites is being blamed for attacks on customer accounts at BestBuy.com.*

Decoy Passwords

- Uncertainty about the actual password
- Store N-1 decoy passwords along
- Attack reduced to online guessing
- All decoys are valid passwords,
  server does not know the difference

Username	*P[0]*	*P[1]*	*P[…]*	*P[N]*

- How many decoys?
  - *16,384 for NIST L2 security when password is reused*

HoneyGen

- Generate honeywords using Machine Learning (ML) techniques

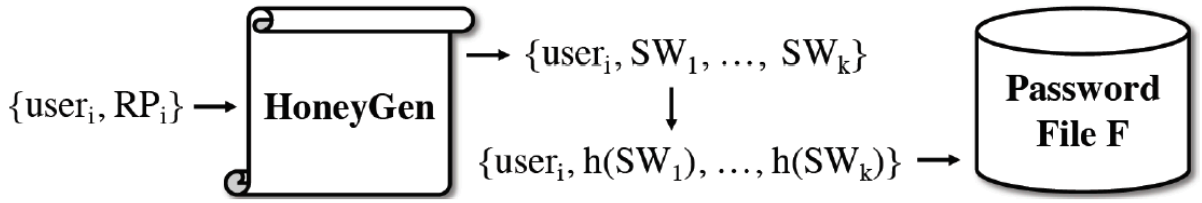


Figure 1: HoneyGen receives i_{th} user's Real Password (RP_i) and responds with an enriched with $(k-1)$ honeywords passwords list containing k sweetwords (SWs) in total. Then, the returned SWs are hashed (according to each operator's hash function $h()$) and stored in the password file F.

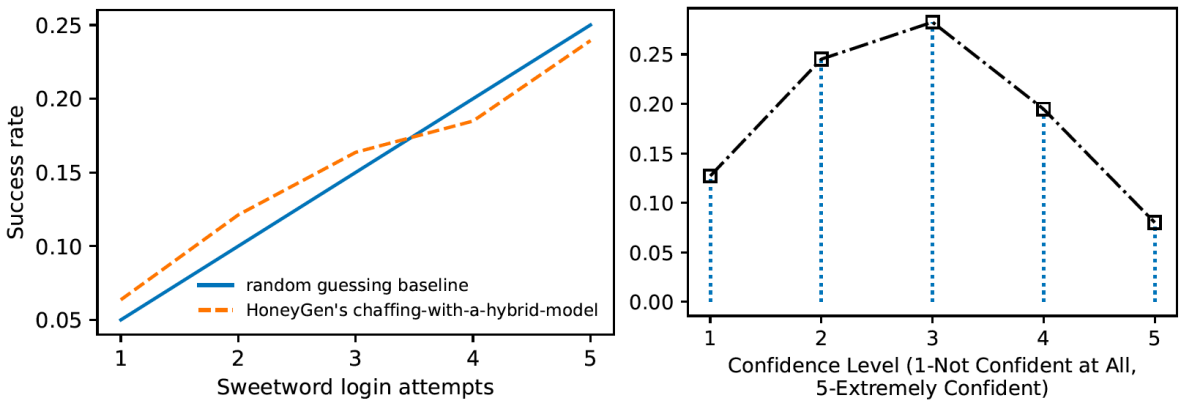
Idea 1:
Random perturbations

- Randomly replace characters from a given password
- Problems:
  - Random generated passwords can be easily distinguished
  - Random generated passwords might not comply with the websites' policies
  - Not generic enough

Idea 2: Probabilistic Model

- Train a model on each operator's password corpus and return as honeywords the top-k nearest neighbours of a given password
- Benefits:
  - Returned passwords match each website's password policies
  - This approach accurately models the password selection behaviour of each website
  - Realistic looking honeywords (they are actual passwords of other users)
- Problems:
  - limited honeywords generation spectrum (no new honeywords can be generated apart from those that already exist in each operator's password corpus)
  - The model-based approach can be reversed

Evaluation – User Study



(a) Flatness graph (b) Distribution of confidence

Figure 6: The flatness graph and the distribution of confidence for the user study's collected results for $T_1 = 1$ up to $T_1 = 5$, using chaffing-with-a-hybrid-model.

Take Away

- Passwords are associated with several problems
  - Not always the user's fault
- Combining services with vouching can be done not just for resetting the password
  - Raising the bar for the attackers
- Generating decoy passwords is a hard problem
  - ML can help