Passwords Decay Words Endure: Secure and Re-usable Multiple Password Mnemonics

Umut Topkara, Mikhail J Atallah, Mercan Topkara

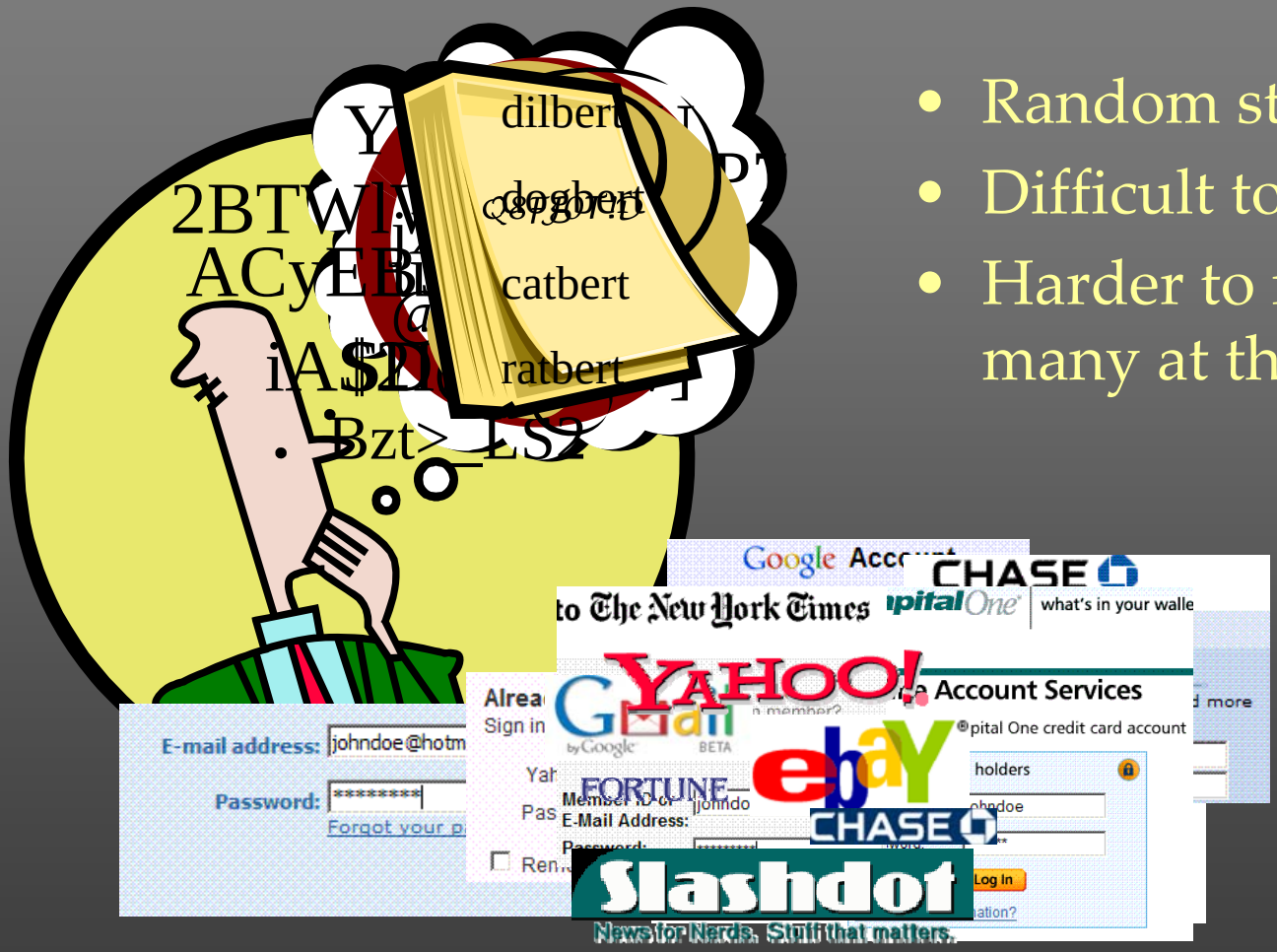
Department of Computer Science
Purdue University
West Lafayette, 47906, IN, USA

PURDUE UNIVERSITY

CERIAS

EMPATHE

What is your password?



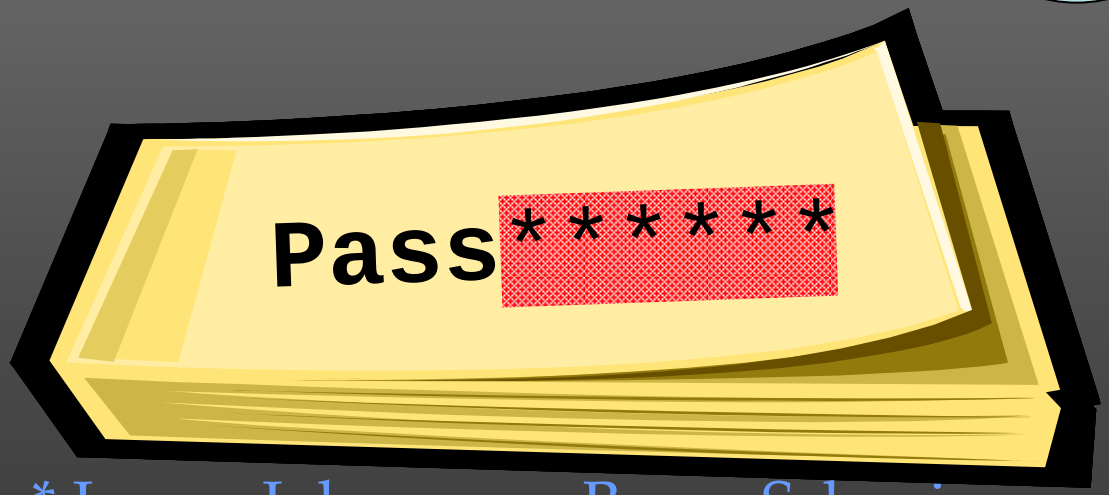
- Random strings
- Difficult to remember
- Harder to remember many at the same time

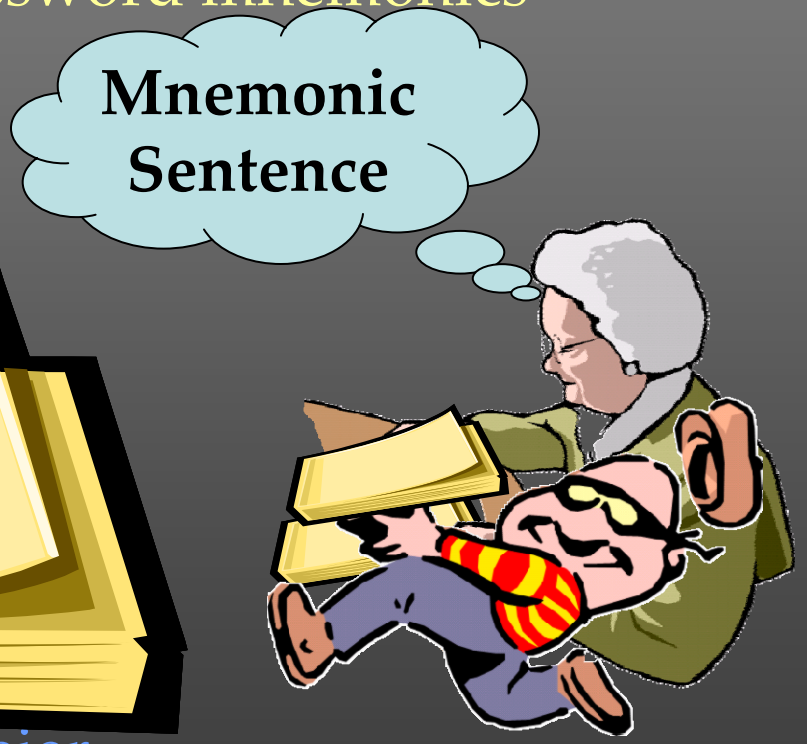
Alternate Solutions

- Require change in infrastructure
  - Graphical passwords, tokens, etc.
- Require trusted computing power
  - Browser plugins, PDA, tokens, etc.

Security Usability Trade-off

- Our Approach: Make passwords more usable
  - Write down your passwords *
  - Complement with secure password mnemonics
  - Something you **have**
  - Something you **know**





* Jesper Johansson, Bruce Schneier

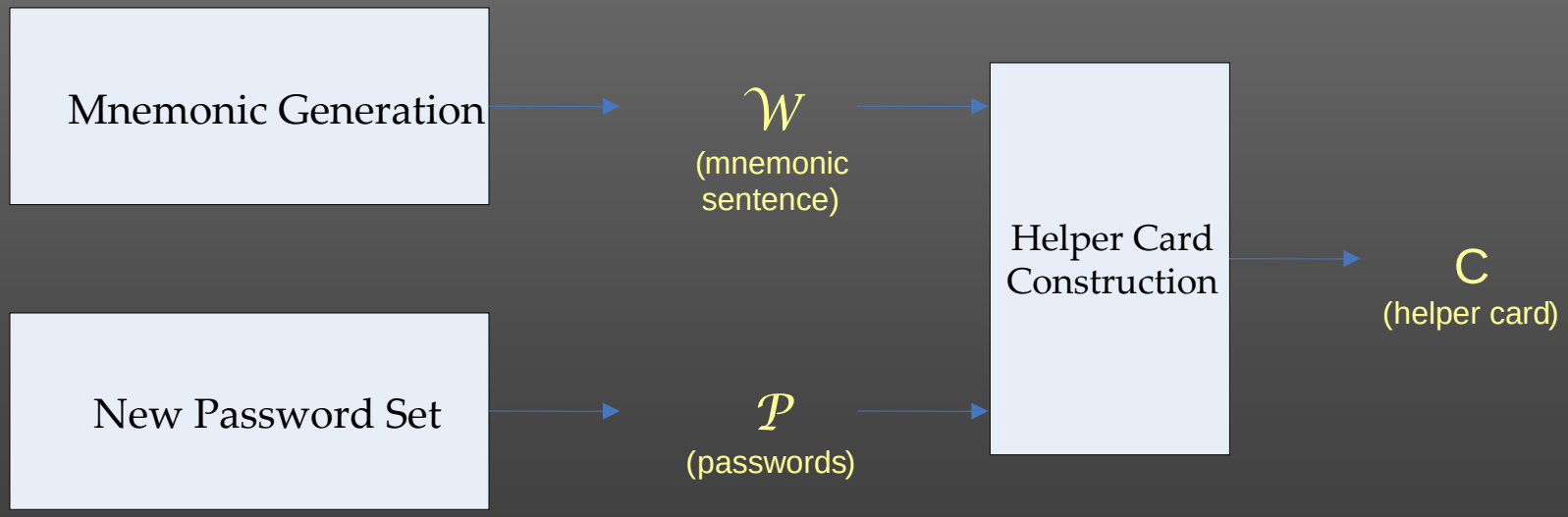
EMPATHE:

rEusable Mnemonics for Password AuTHEntication

- Automatically generated mnemonics
- Support truly random passwords
- Handle multiple passwords with same mnemonic
- No domino effect in password compromise
- Write down a complete description of the password
- Easy user reconstruction of the passwords
- No requirement of additional computing power
- No change in existing infrastructure
- Compatible with existing passwords
- Keep the mnemonic, change the password

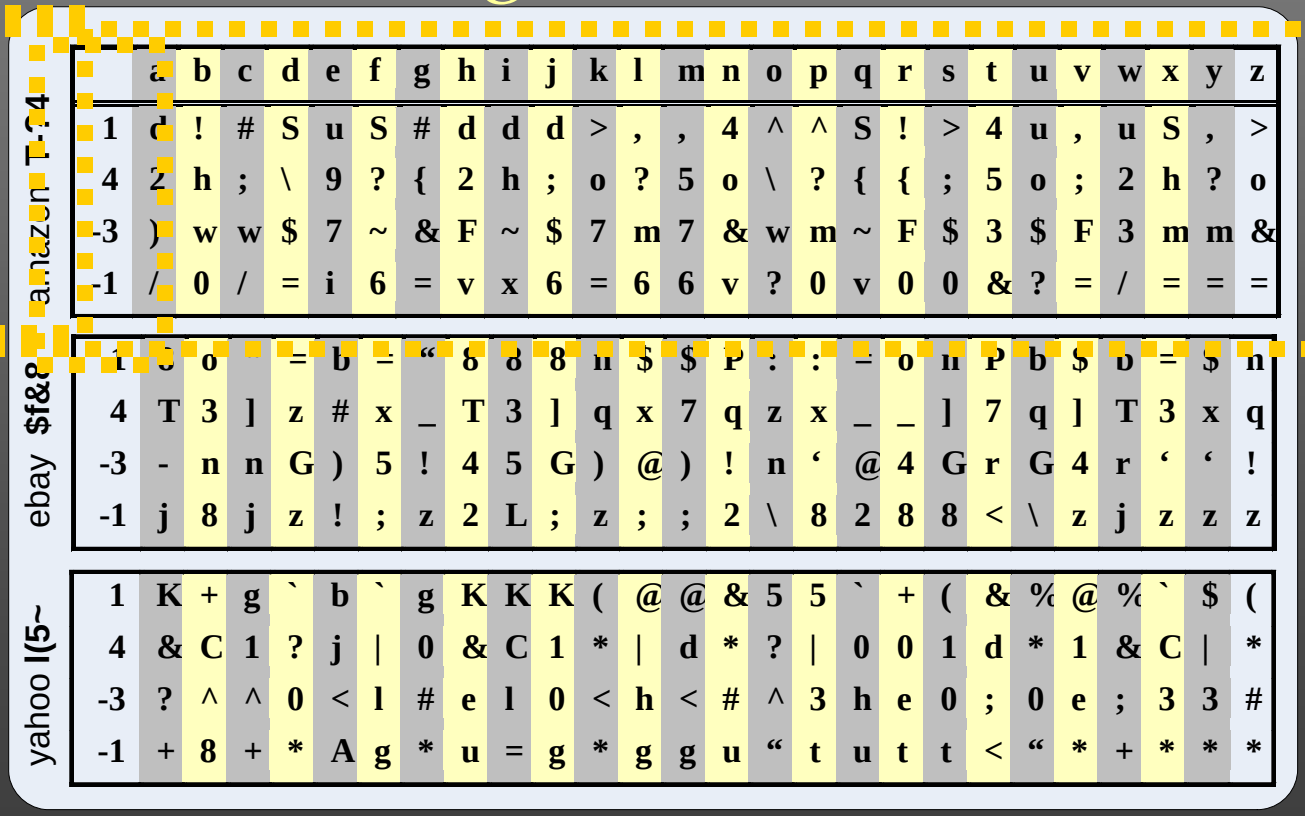
Usage Scenario

- Select a mnemonic sentence
- Choose strong passwords
- Print a helper card
- Authenticate



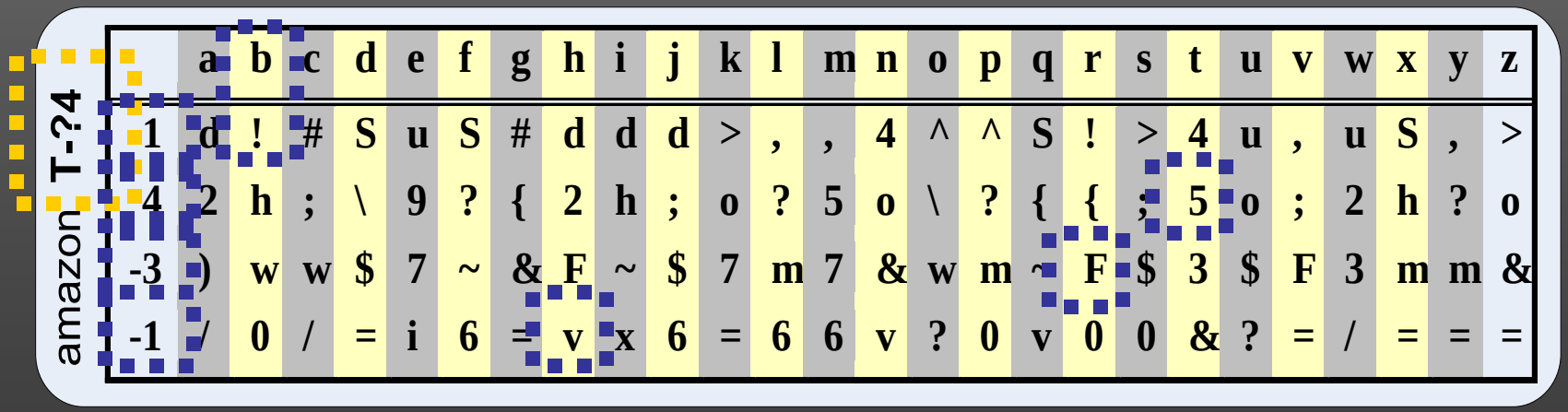
Authentication with Helper Card

- "The birth of ice-cream: why and how we sneeze at midnight."



Authentication with Helper Card

1	2	3	4	5
b	**i**	**r**	**t**	**h**
-5	-4	-3	-2	-1



Helper Card

- Secret Sharing
- Without Encryption
  - Random **P**
  - Random R_s
  - Public (P XOR R_s = R_h)
- $e(R_s) = W = d^{-1}(R_s)$ (e.g., W="birth")
- $R_s = d(W) = d(W[1]).d(W[4]).d(W[-3]).d(W[-1])$
  d(birth) = d(b).d(t).d(r).d(h)
- $p_i = d(W[j])$ XOR $r_{h,i}$ = lookup(i,W[j])
- lookup(i,W[j])= d(W[j]) + z mod(95)

Helper Card Security

- Adversary with password file:
  - Offline cracking
  - 95^8 equally probable passwords
- Adversary with helper card:
  - Login trials
  - 9^4 possible passwords per account
  - Most systems regulate login trials

Multiple Passwords

- Adversary with one password
  - No domino effect
  - Passwords independently chosen
- Adversary **also** with helper card
  - T-?4!5Fv → "birth"
  - P(ice-cream) = P(ice-cream | birth)
  - Mnemonic sentence generation

Mnemonics

- Reminder for hard to remember information
  - e.g. unrelated sequence of objects
  - **M**y **V**ery **E**ager **M**other **J**ust **S**ewed **U**s **N**ew **P**ajamas
- [Miller 1956, Human Memory And The Storage Of Information]:
  - Semantic association: Associate a meaning
    - e.g., Manhattan, Italy Map
  - Progression of ideas: Connect as a story
    - e.g., May I have a large container of coffee? (3.1415926)
  - Syntactic Coherence: As grammatical as possible
  - Short encoding: Size of the story
    - e.g., 1101 0110 1100 vs 13 6 12

What's wrong with mnemonics?

- Has to be easy to remember:

  An apple a day sends the doctor away
- Has to be hard to guess:
  - Regularity in language can be a pitfall:
    - **My (dog|cat|pet)'s name is (fido|dusty…) → M(d|c|p)ni?**
    - P(mother) ≠ P(mother | birth)
  - Need high entropy word sequences
- Mnemonic fatigue:
  - Hard to come up with new memorable mnemonics
  - Multiple accounts
  - Periodic password reset

Mnemonic Generation

- Start with:
  - Set of human-crafted memorable sentences
  - Set of words, $R_{s,i} = d(W_i)$
- Result:
  - Set of mnemonic sentences for $R_{s,I}$
  - Grammatical
  - Probably easy to remember
- Use a subset of English
  - Restricted, $P(W_i) = P(W_i | W_j)$
  - Large enough for passwords

Conclusion

- People are good in keeping cards secure
- Many already use mnemonics
- First step
  - Single mnemonic for multiple Passwords
  - Good security and usability
- Periodic password changes now easy
- Shared passwords possible
- Recall rarely used passwords

Future work

- Further improve against brute force
- More effective use of Natural Language Processing
- USENIX'07: Authentication In Constrained Environments

Thanks

- Anonymous referees
- Questions?