18733: Applied Cryptography

Human Computable Protocols: Password-based Authentication

Anupam Datta

With Jeremiah Blocki and Manuel Blum

Carnegie Mellon University

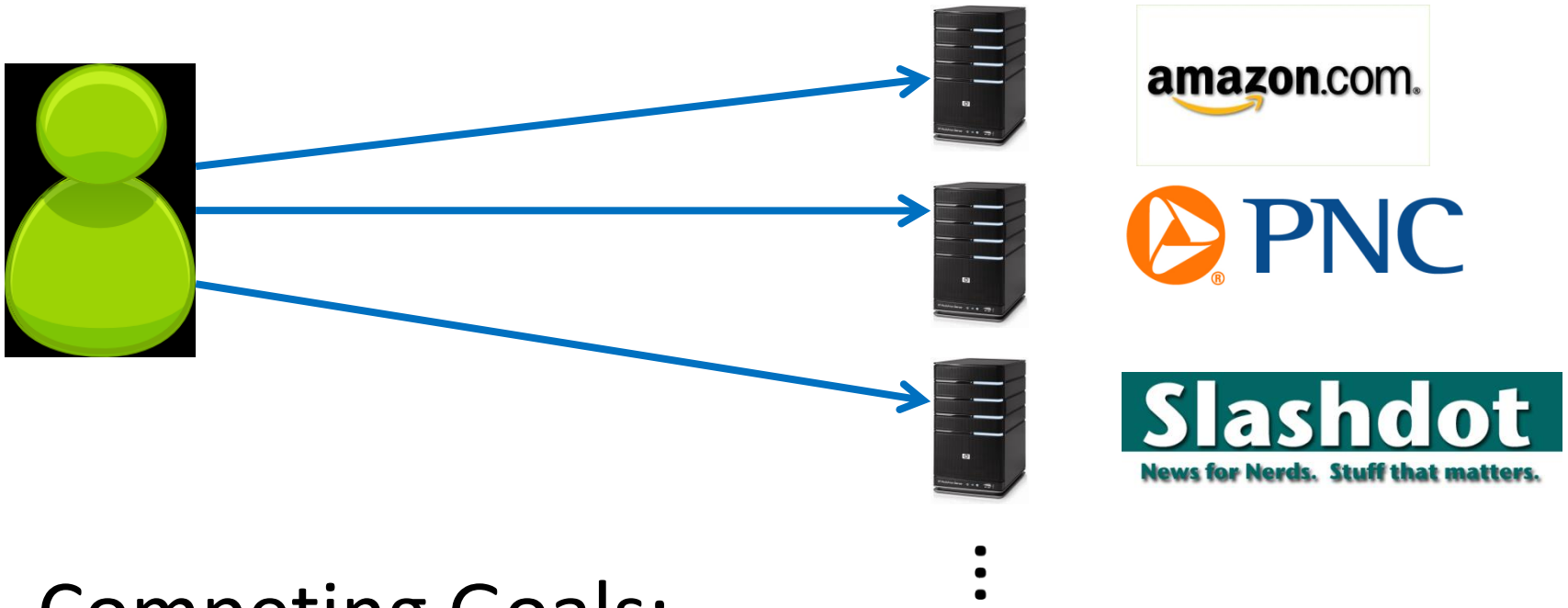
Memory Experiment 1

Person	Bill Clinton
Action	Tickling
Object	Peach

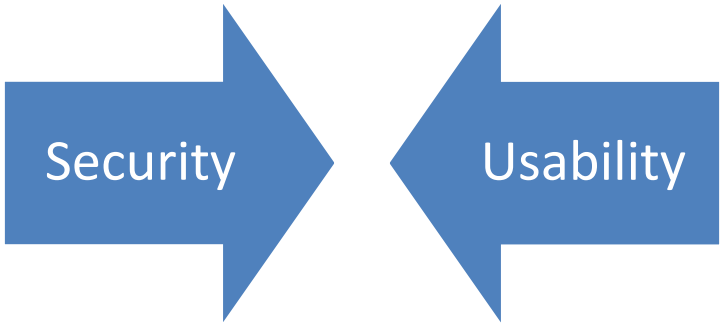
Memory Experiment 2

Person	Michelle Obama
Action	Bouncing
Object	Smore

Password Management



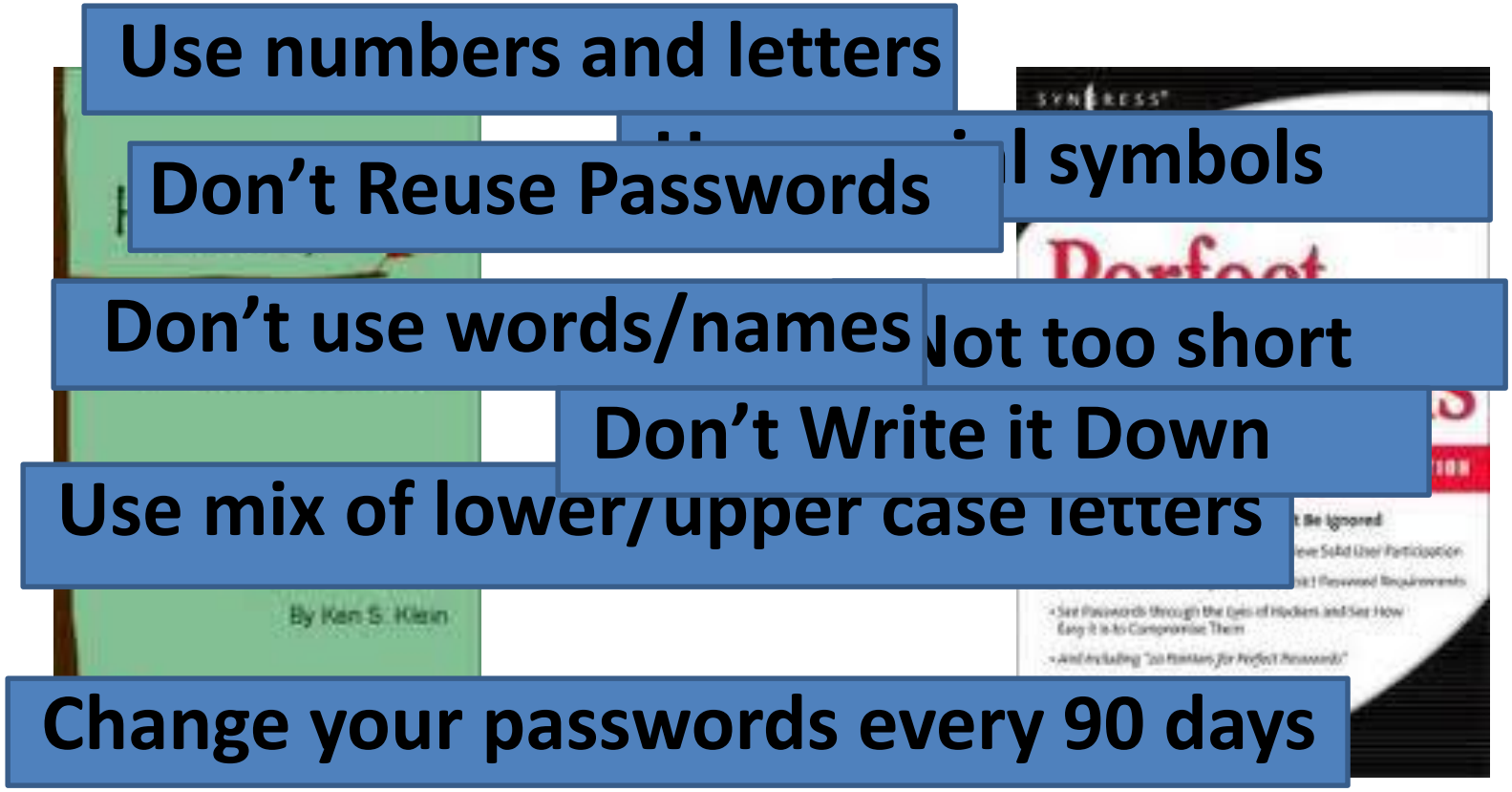
Competing Goals:



Security Problem

- Password breaches at major companies have affected millions of users.

Traditional Security Advice



Usability Problem

Fundamental Question

- How can we evaluate password management strategies?
  - Quantify Usability
  - Quantify Security

Outline

- **Introduction**
  - Motivation
  - **Our Approach**
  - Related Work
- Usability and Security Models
- Shared Cues
- Human Computable Passwords
- Conclusion and Future Vision

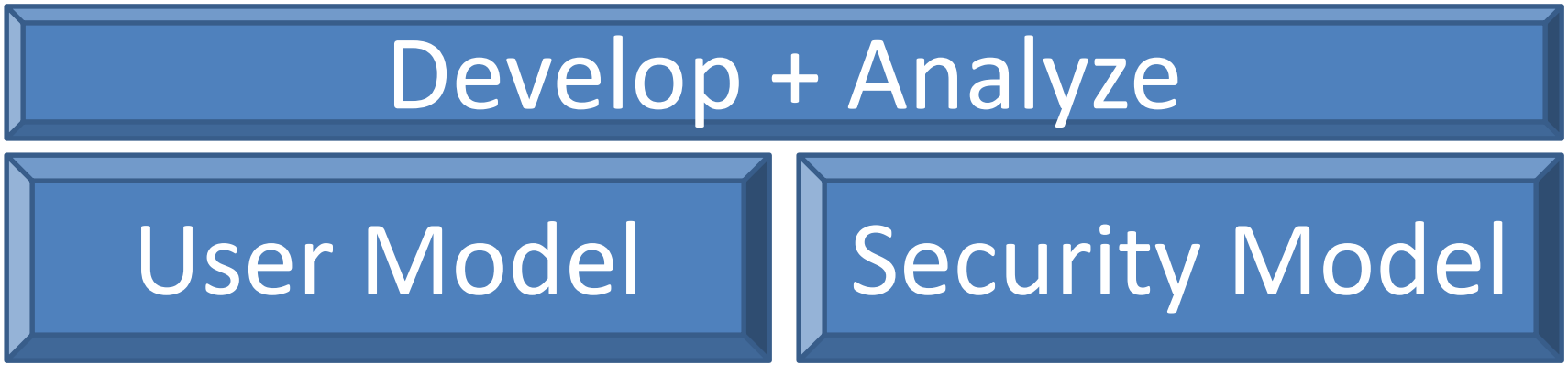
Traditional Approach

Propose New Password Management Scheme

User Study: Evaluate New Password Management Scheme

Our Thesis

User models and security models can guide the development of human authentication schemes with analyzable usability and security properties.



Our Approach: User Models

User Model

Capabilities + Behavior

Example Capability: Users can remember a random secret with enough rehearsal.

Example Behavior: How often a user visits each website on average.

Our Approach: User Models

User Model

Capabilities + Behavio

Develop + Analyze

User Model

Empirical Validation

Fast: Evaluation does not require expensive user studies

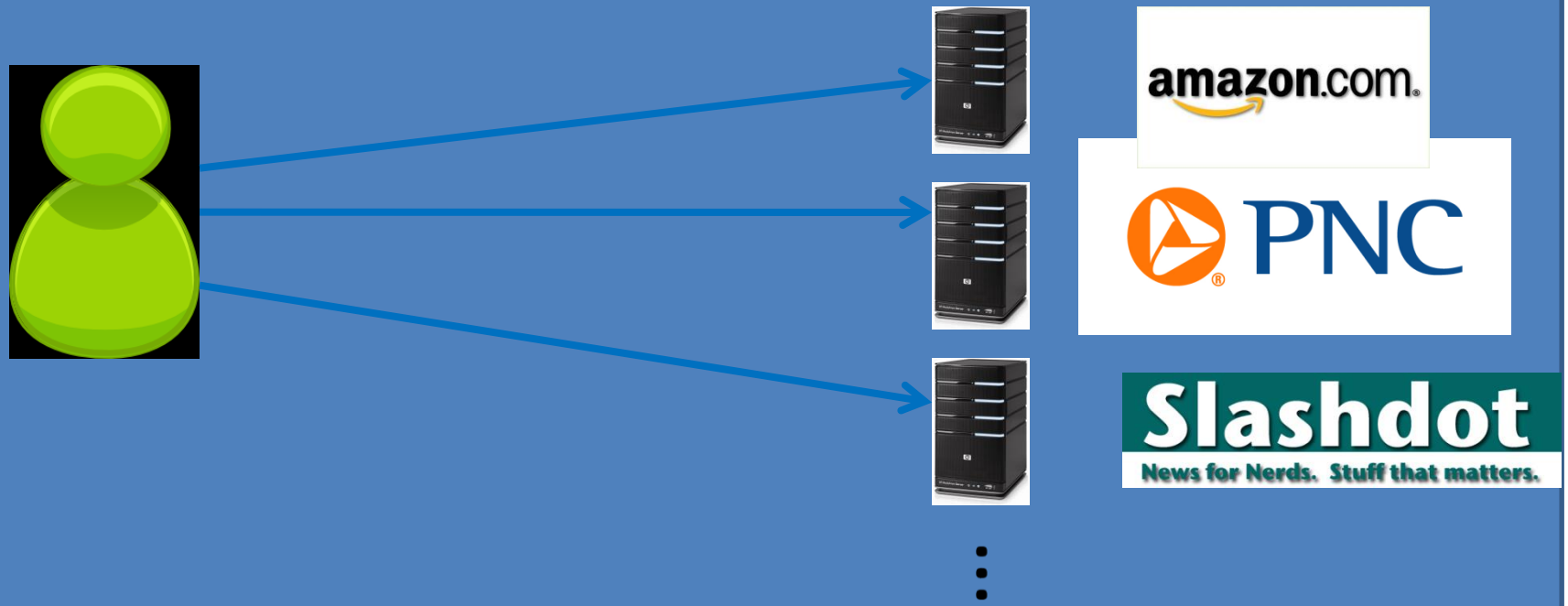
password management schemes

Outline

- **Introduction**
  - Motivation
  - Our Approach
  - **Related Work**
- Usability and Security Models
- Shared Cues
- Human Computable Passwords
- Conclusion and Future Vision

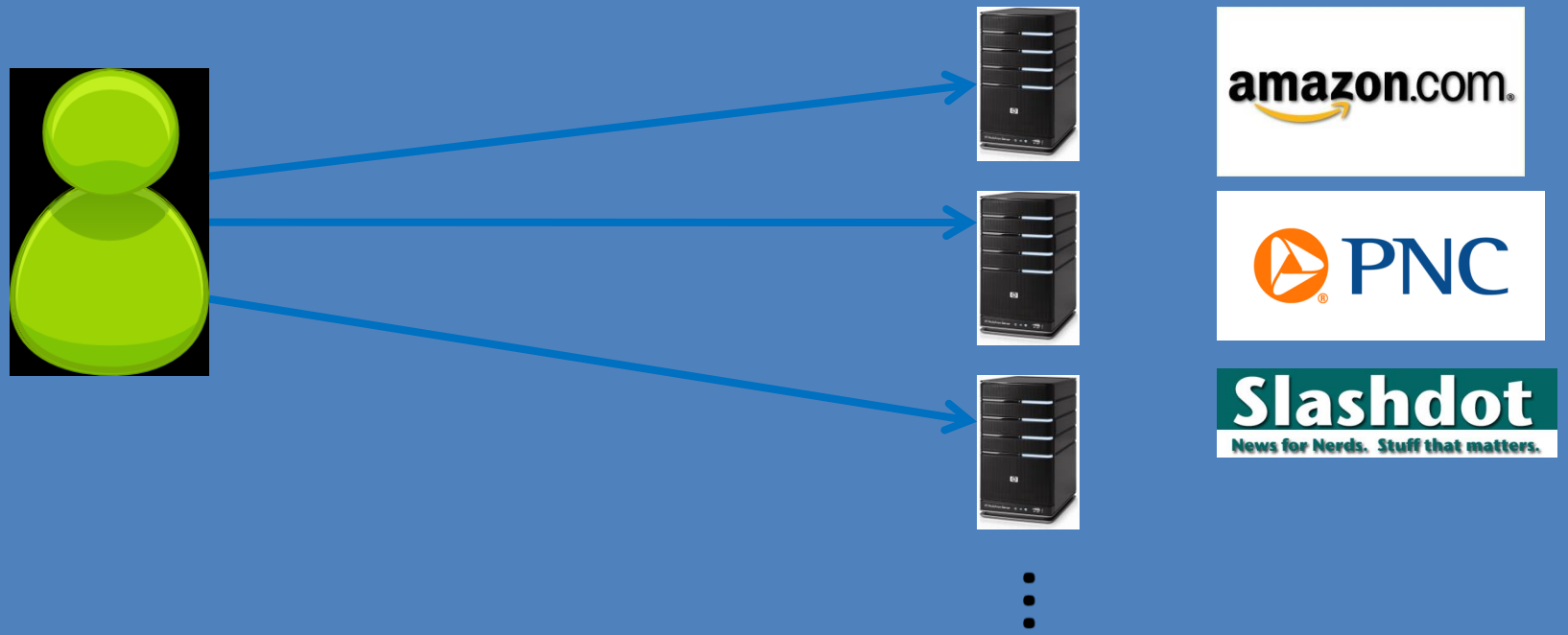
Related Work

Challenge: Multiple Passwords



Related Work

Goal: Quantify Security for Multiple Passwords



Related Work

Goal: Minimize Trust Assumptions about User's Computational Devices

Related Work

Quest to Replace Passwords [BHOS2012]

Outline

- Introduction
- **Usability and Security Models**
  - **Example Password Management Schemes**
  - Usability Model
  - Security Model
- Shared Cues
- Human Computable Passwords
- Conclusion and Future Vision

Scheme 1: Reuse Strong Password

- Pick four random words w_1, w_2, w_3, w_4

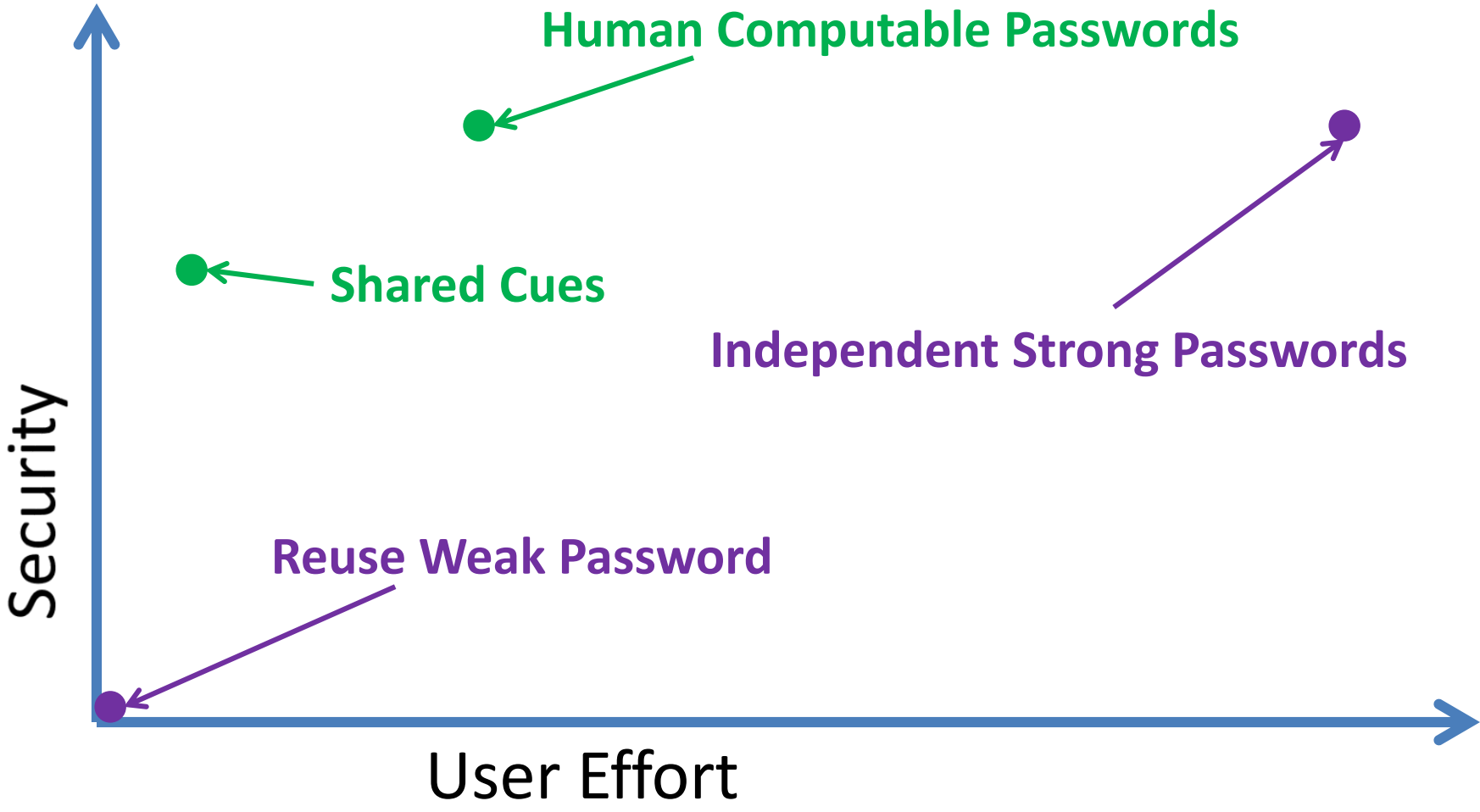
Account	Amazon	Ebay
Password	$w_1w_2w_3w_4$	$w_1w_2w_3w_4$

Scheme 2: Strong Random Independent

Four Independent Random Words per Account

Account	Amazon	Ebay
Password	$w_1w_2w_3w_4$	$x_1x_2x_3x_4$

Preview of Results



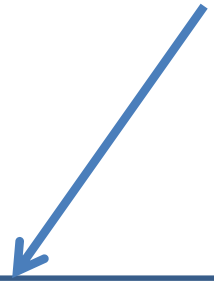
Outline

- Introduction
- **Usability and Security Models**
  - Example Password Management Schemes
  - **Usability Model**
  - Security Model
- Shared Cues
- Human Computable Passwords
- Conclusion and Future Vision

First Attempt: Chunking

- Memorize: nbccbsabc
- Memorize: tkqizrlwp
- 3 Chunks vs. 9 Chunks!

Incomplete

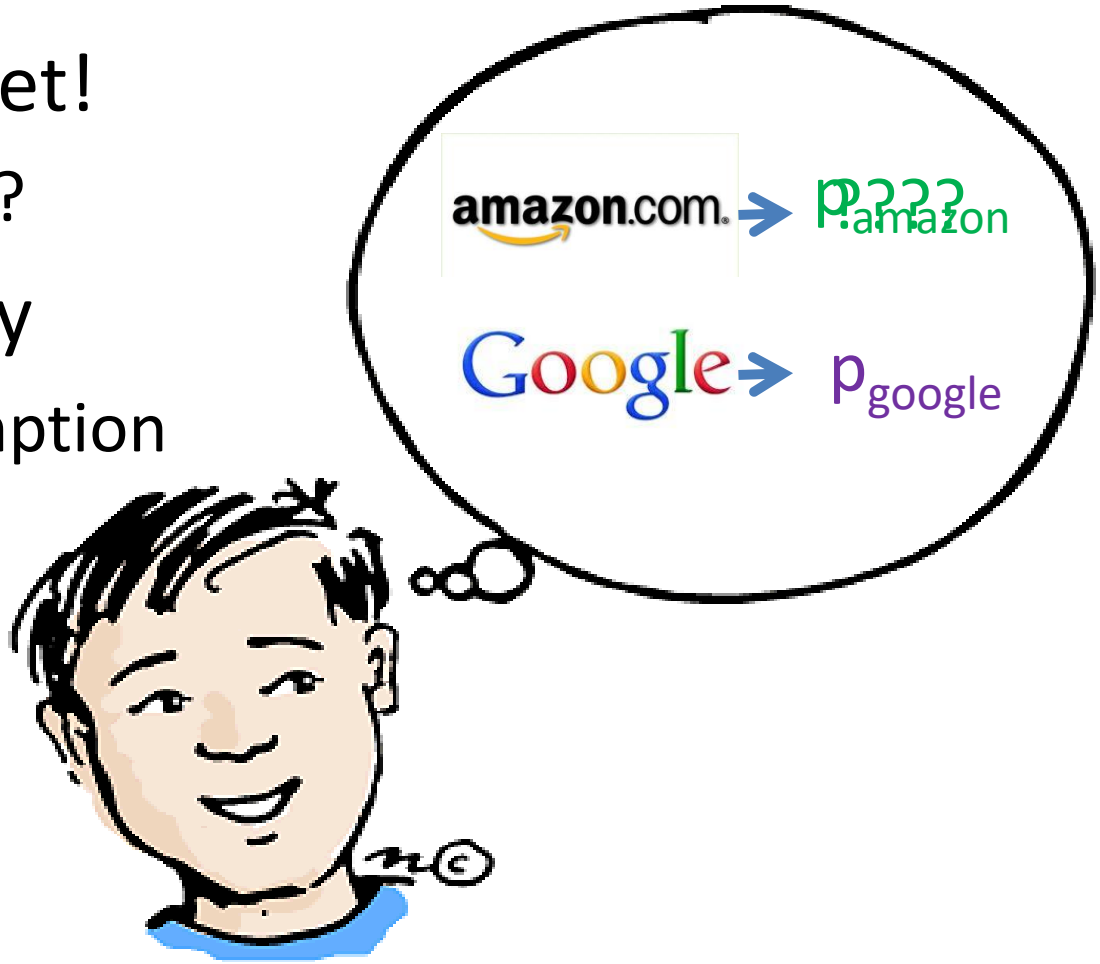


Usability Goal: Minimize Number of Chunks in Passwords

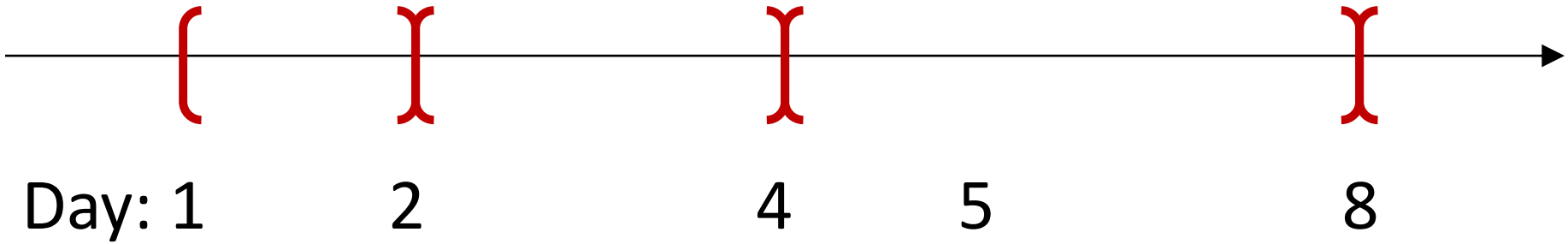
Source: *The magical number seven, plus or minus two* [Miller, 56]

Human Memory: Vast, but Lossy

- Rehearse or Forget!
  - How much work?
- Quantify Usability
  - Rehearsal Assumption



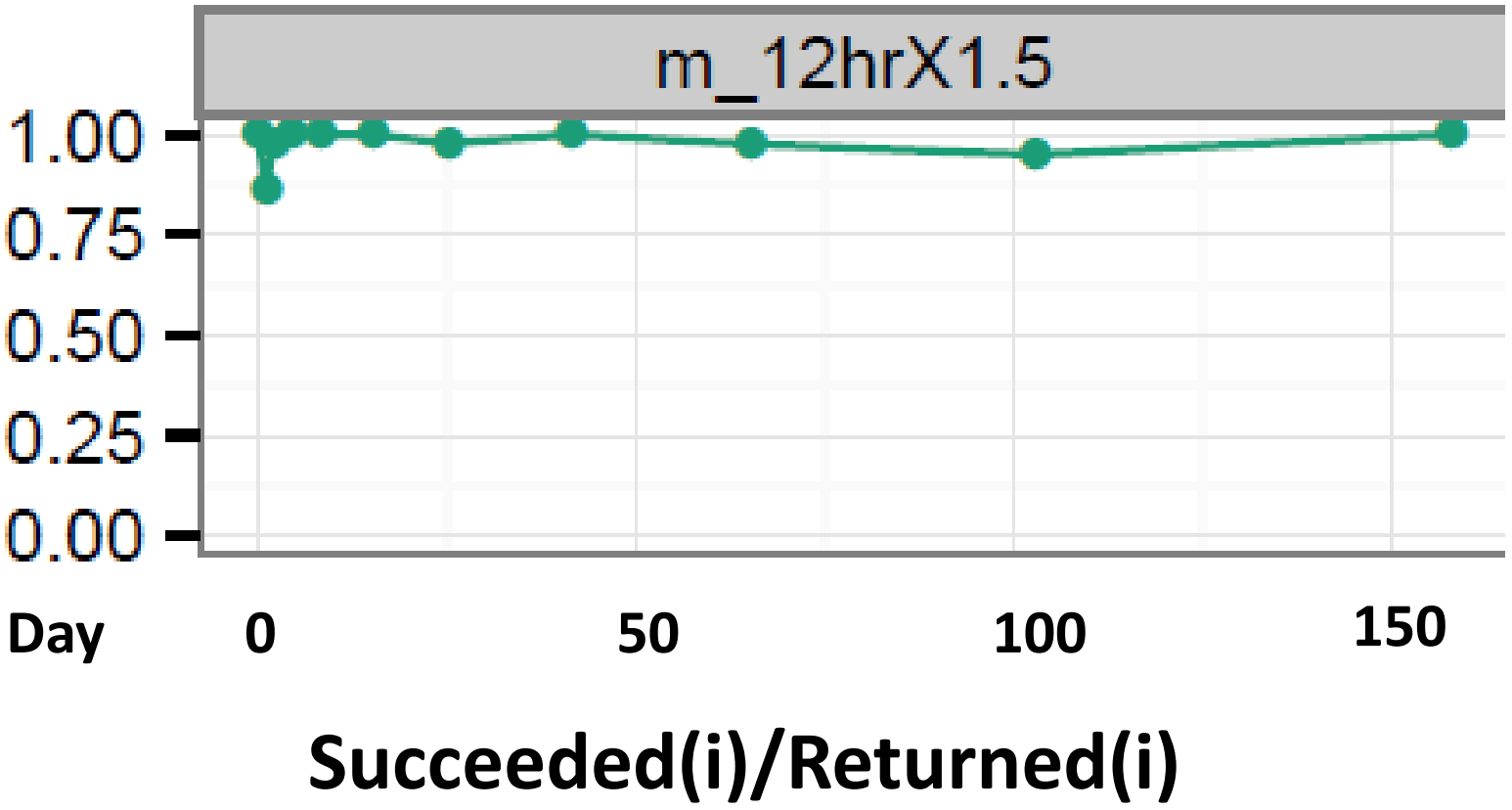
Memory Capability



Expanding Rehearsal Assumption: user maintains cue-association pair by rehearsing during each interval $[s^i, s^{i+1}]$.

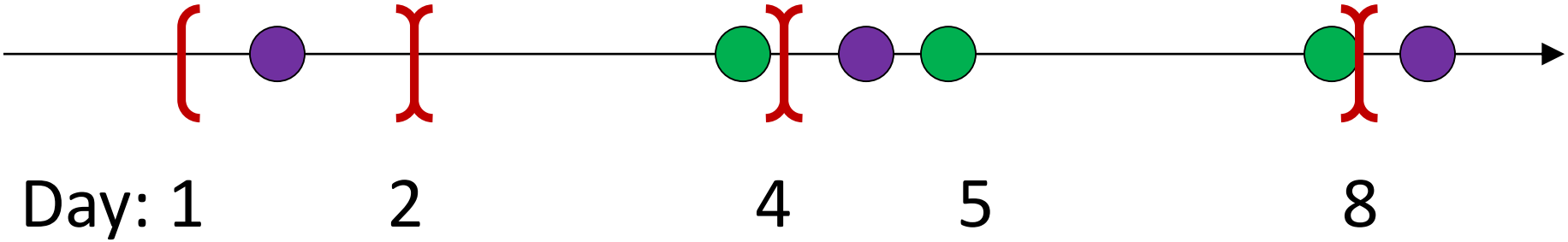
Source: *Optimization of Repetition Spacing in the Practice of Learning* [WG, 94]

Memory Capability



Source: Spaced Repetition and Mnemonics Enable Recall of Multiple Strong Passwords [BKCD15]

Natural Rehearsal



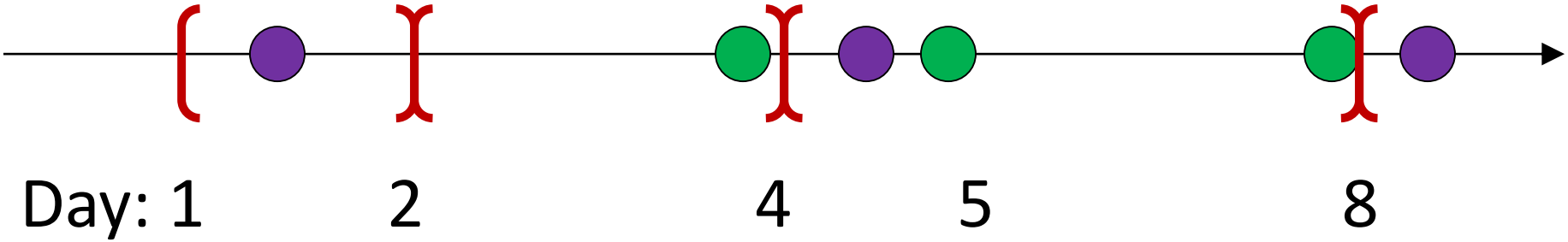
Expanding Rehearsal Assumption: user maintains cue-association pair by rehearsing during each interval $[s^i, s^{i+1}]$.

Visit Amazon: Natural Rehearsal Google

$\mathbf{X_t}$: extra rehearsals to maintain *all* passwords for t days.

Source: *Optimization of Repetition Spacing in the Practice of Learning* [WG, 94]

Extra Rehearsals



X_t: extra rehearsals to maintain *all* passwords for t days.

Usability Measure: X_t	**Reuse Password**	**Independent Passwords**
X_8	0	2

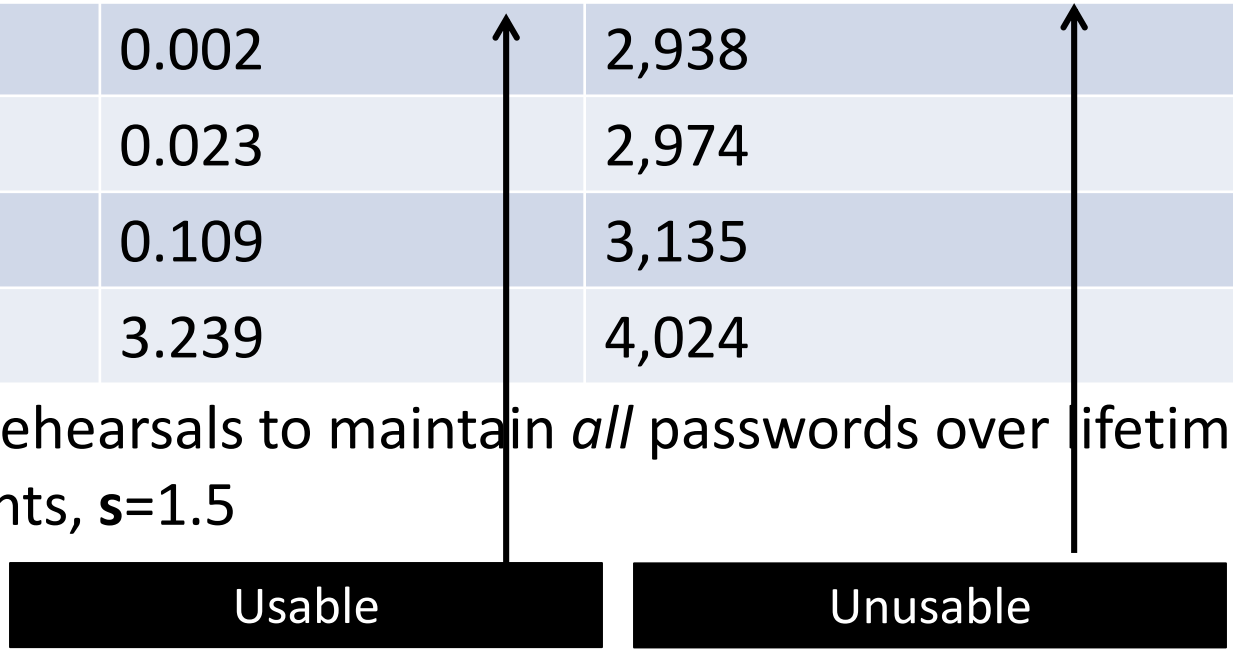
Usability Goal: Minimize X_t

Usability Results

	Reuse Strong	Strong Random Independent
Active	0.002	2,938
Typical	0.023	2,974
Occasional	0.109	3,135
Infrequent	3.239	4,024

E[$\mathbf{X}_{\infty}$]: Extra Rehearsals to maintain *all* passwords over lifetime.

m = 75 accounts, **s**=1.5

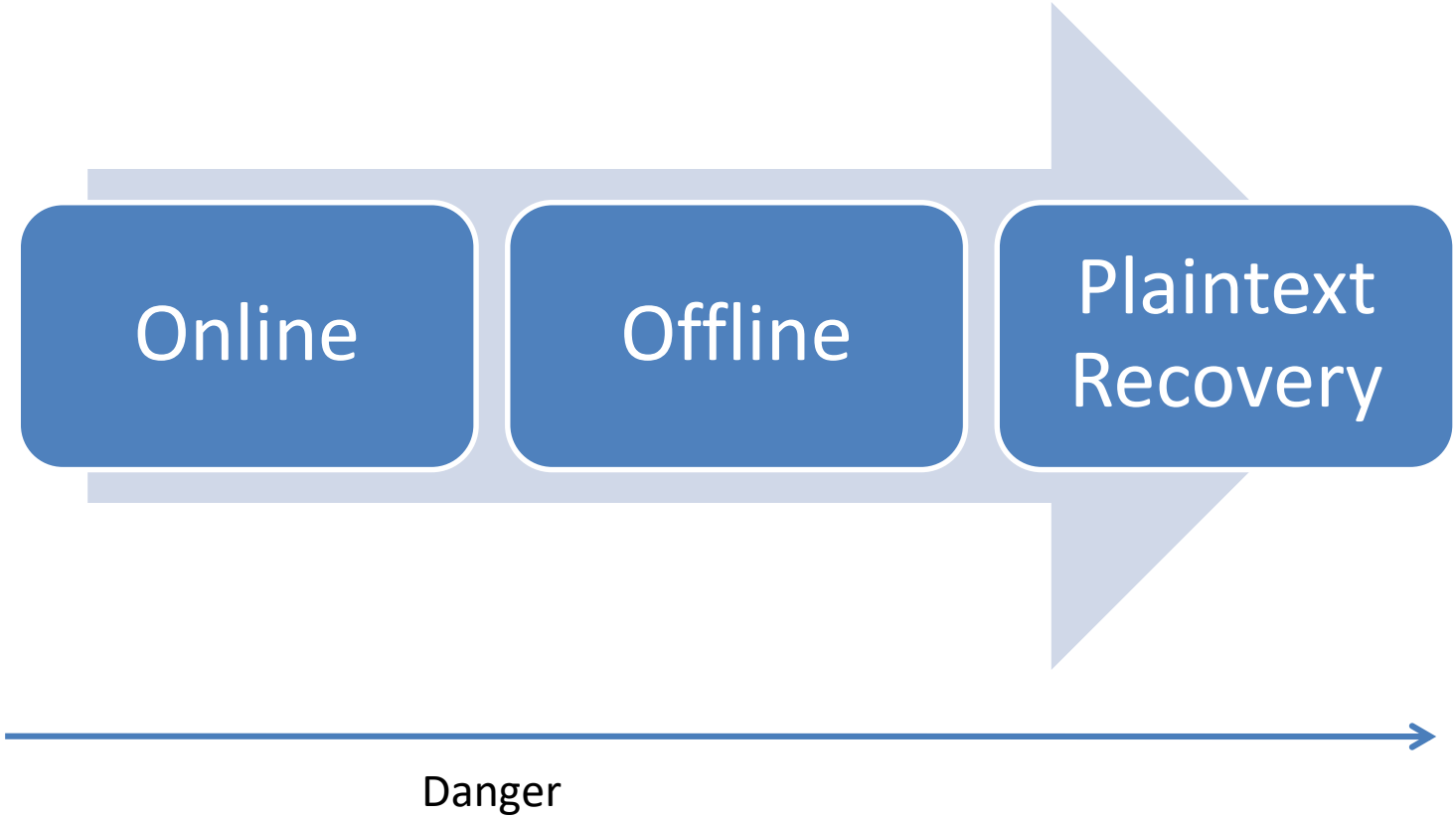


Outline

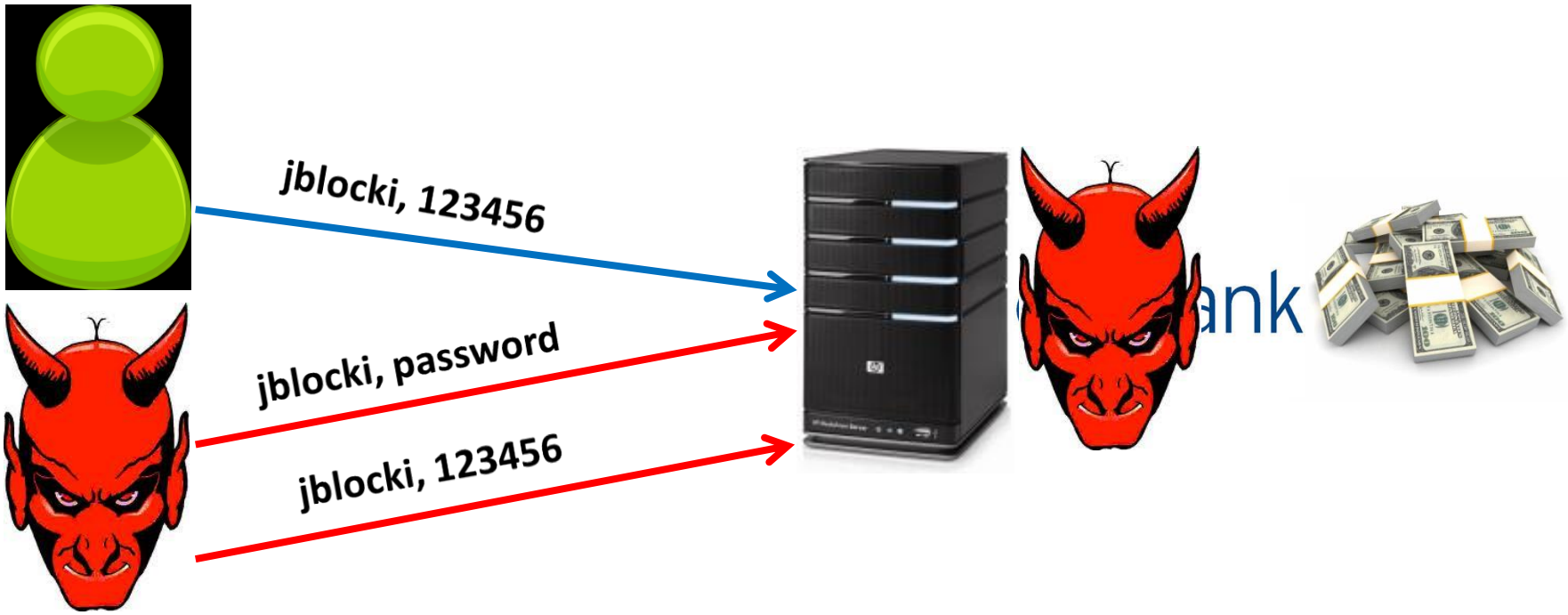
- Introduction
- **Usability and Security Models**
  - Example Password Management Schemes
  - Usability Model
  - **Security Model**
- Shared Cues
- Human Computable Passwords
- Conclusion and Future Vision

Security (what could go wrong?)

Three Types of Attacks

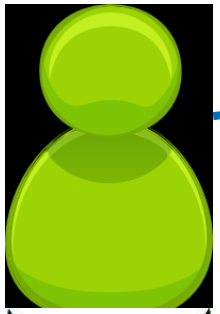


Online Attack



Guess Limit: k-strikes policy

Offline Dictionary Attack



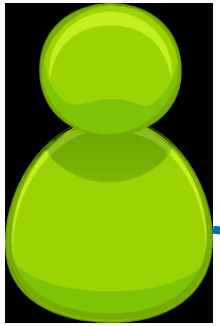
jblocki, Unbr3akabl3

Username	Salt	Hash
jblocki	89d978034a3f6	75fe9ccf4a568f31e66b8597b8eb97c2e915e6b1

+

SHA1(Unbr3akabl389d978034a3f6)=
75fe9ccf4a568f31e66b8597b8eb97c2e915e6b1

Plaintext Recovery Attack



pwd

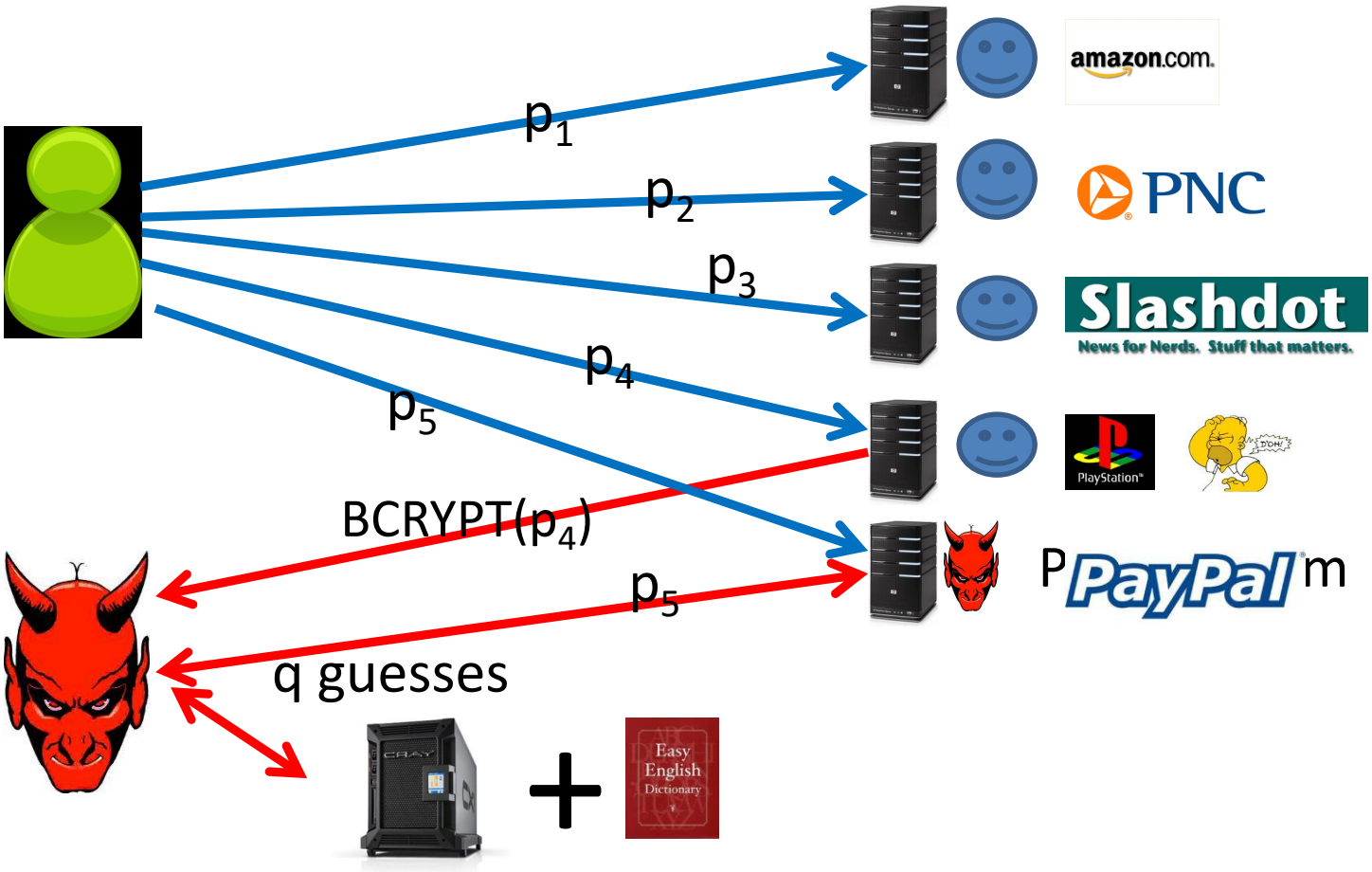
PayPal aul.com

pwd

Security Philosophy

- Dangerous World Assumption
  - Targeted adversary has background information about our user (e.g., Hobbies, Birthdate)
  - Adversary can adapt after learning the user's new password management strategy
- Limit Damage when something goes wrong
  - Offline attacks should fail with high probability
  - Contain damage of a successful phishing attack

Security as a Game



(q, δ,m,s,r,h)-Security

For any adversary **Adv**

$$\boldsymbol{Pr}\left[\ \text{  }\ \middle|\ \textbf{Adv},\text{q,m,s,r,h}\right] \leq \delta$$

q = # offline guesses

s = # online guesses

r = # 

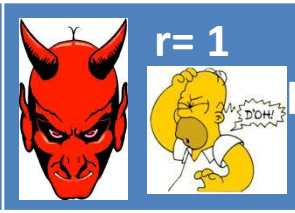
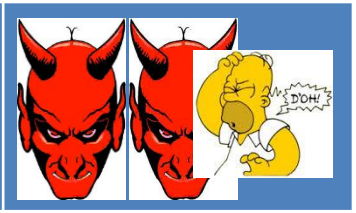
Plaintext Recovery Accounts

h = # 

Offline Attack Accounts

m = # of accounts

Security Results

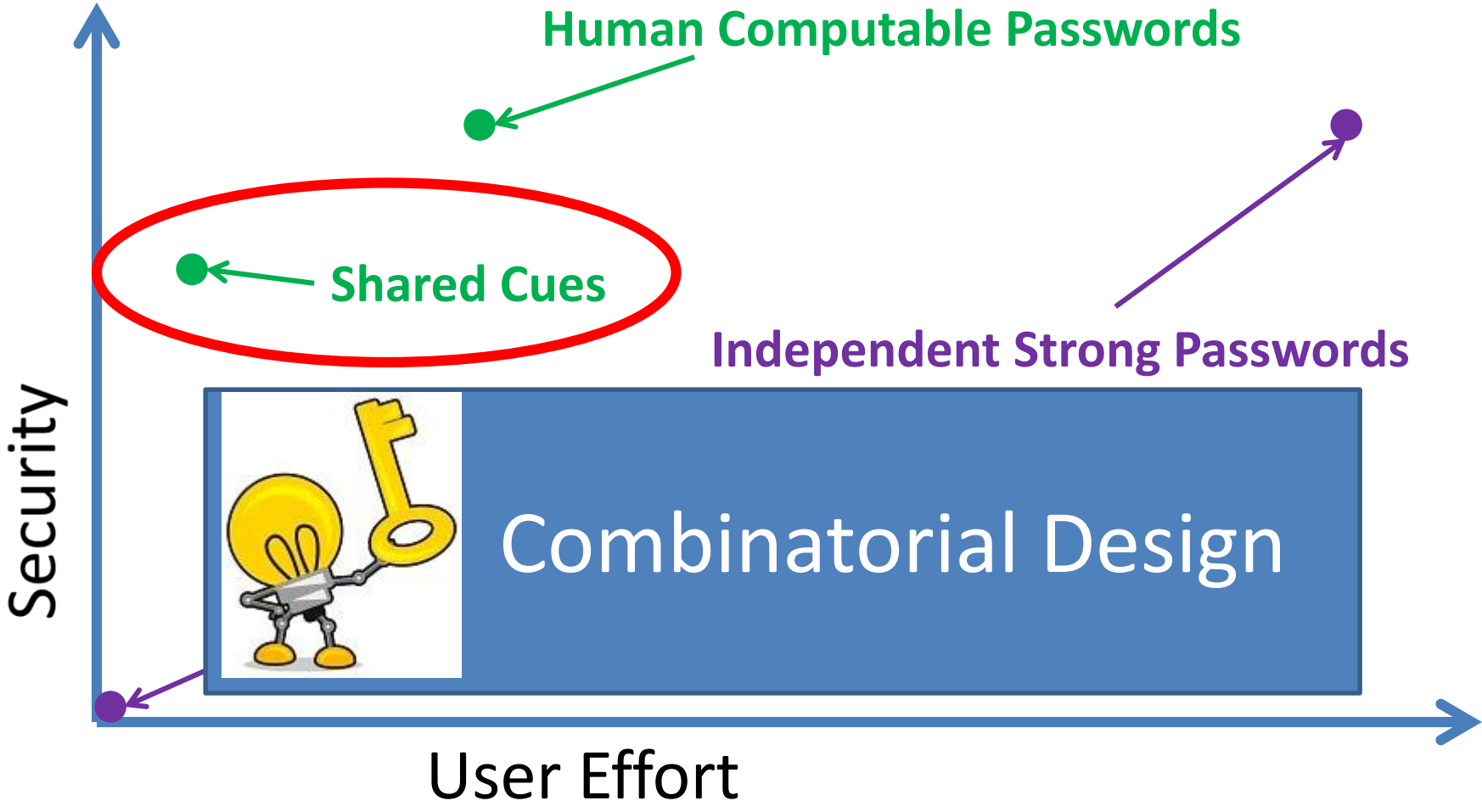
Attacks	 r= 1	 r= 1 h=1	 r=2	
Reuse	**No**	**No**	**No**	**No**
Strong Random Independent	**Yes**	**Yes**	**Yes**	**Yes**

Usable + Insecure

Unusable + Secure

$(10^{10}, \delta, m, 3, r, h)$-security

Preview of Results



Outline

- Introduction
- Usability and Security Models
- **Shared Cues**
- Human Computable Passwords
- Conclusion and Future Vision

Our Approach

Public Cue

Private

Action: kicking

Object: penguin

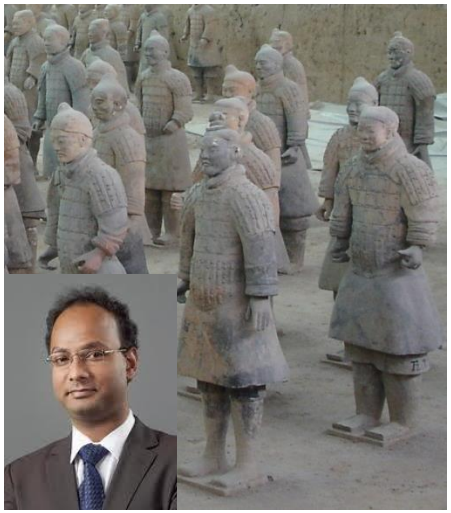
Login

amazon.com.

Action

Object

...



Object

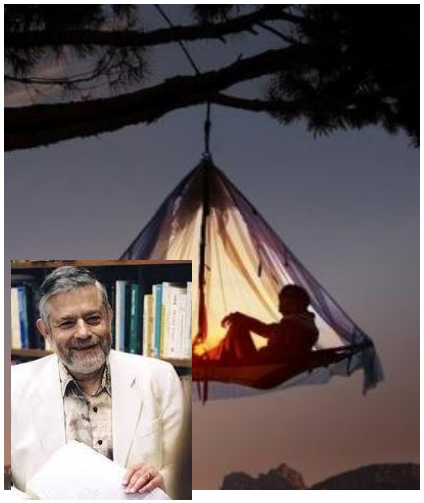
Pwd

Kic + Pen + ... + Pir

Login

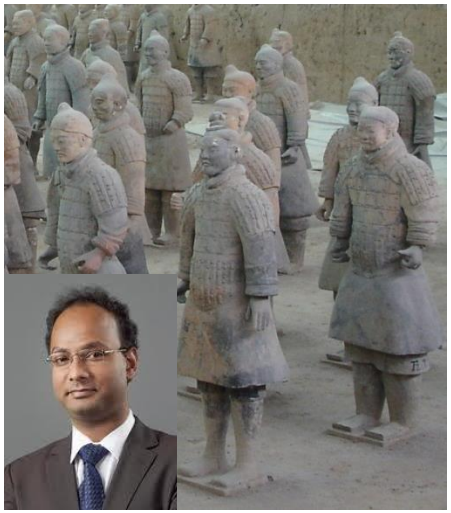
PayPal

Action



Object

...

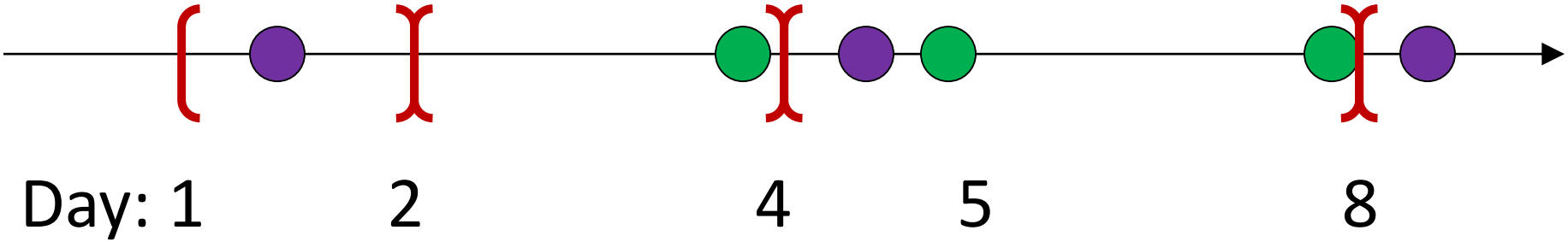


Action

Pwd

Kic + Lio + … + Kis

Sharing Cues

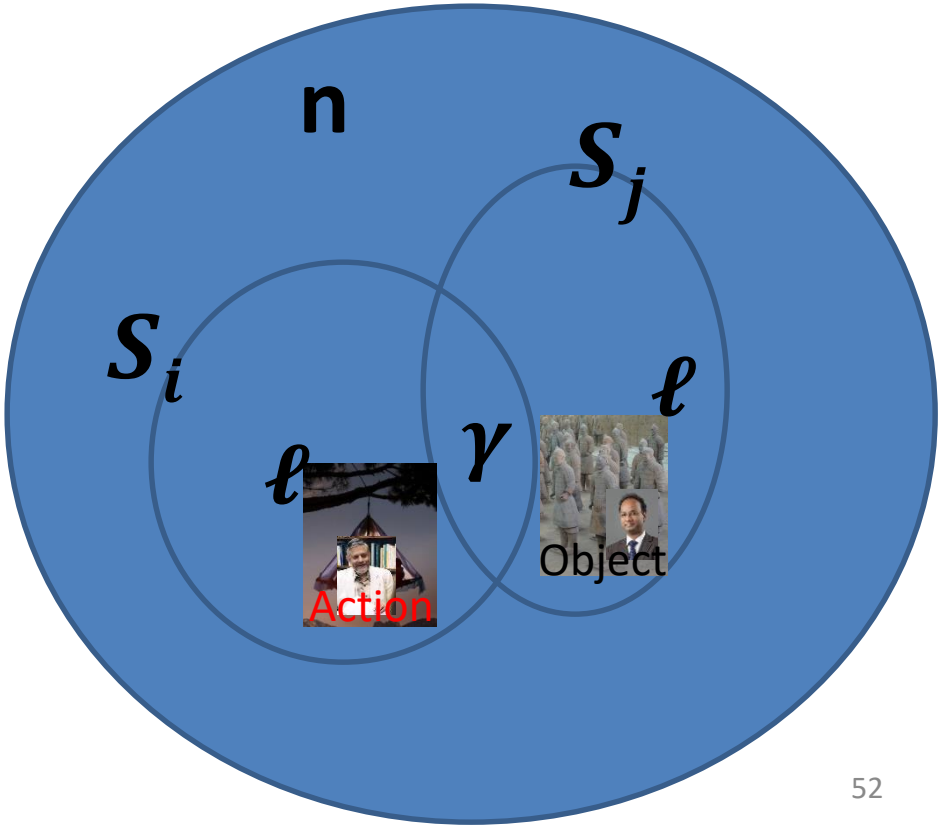


- Usability Advantages
  - Fewer stories to remember!
  - More Natural Rehearsals!
- Security?

(n,$\boldsymbol{\ell}$, $\boldsymbol{\gamma}$)-Sharing Set Family

Definition: A (n,$\boldsymbol{\ell}$,$\boldsymbol{\gamma}$)-Sharing Set Family of size m is a family of sets {S_1,…,S_m} with the following properties

- $\forall \mathbf{i} \neq \boldsymbol{j}, |\boldsymbol{S_i} \cap \boldsymbol{S_j}| \leq \boldsymbol{\gamma}$
- $\forall \mathbf{i}\ |\boldsymbol{S_i}| = \boldsymbol{\ell}$
- $\left|\cup_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{S_i}\right| = \boldsymbol{n}$



(n,ℓ, γ)-Sharing Set Family

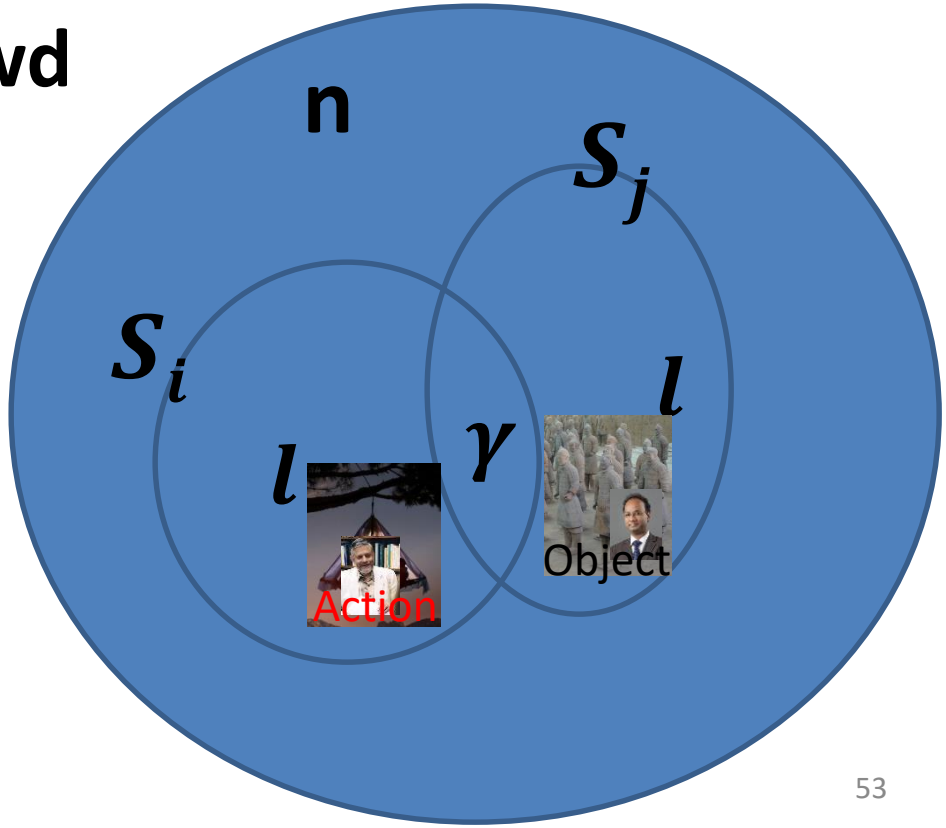
m – number of passwords {S_1,…,S_m}.

n/2 – total #PAO stories

ℓ – #words in each pwd

γ – max intersection

S_i – PAO stories for account i.



Sharing Cues

Thm: There is a (43,4,1)-Sharing Set Family of size m=90

- Proof?
  - Chinese Remainder Theorem!
  - Notice that 43 = 9+10+11+13 where 9, 10, 11, 13 are pair wise coprime.
  - A_i uses cues: {i mod 9, i mod 10, i mod 11, i mod 13}

Chinese Remainder Theorem

By the Chinese Remainder Theorem there is a unique number x s.t

1) $1 \leq x \leq 90$

2) $x \equiv i \; mod \; 9$

3) $x \equiv i \; mod \; 10$

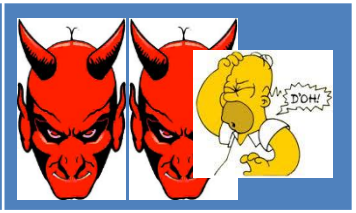
Hence, for $i \neq j$ accounts A_i and A_j cannot use the same red cue and blue cue.

Usability Results

	Reuse	Strong Random Independent	[SC-1] 15 PAO Stories	[SC-0] 7 PAO Stories
Active	?0	2,938	9.8	4.0
Typical	?0	2,974	11.8	4.5
Occasional	?0	3,135	15.2	5.5
Infrequent	3.2	4,024	93.2	25.7

E[$\mathbf{X}_{\infty}$]: Extra Rehearsals to maintain *all* passwords over lifetime.

Summary: Shared Cues

Attacks	 r= 1	 r= 1  h=1	 r=2	
(n,ℓ,ℓ)-Sharing [Reuse]	No	No	No	No
(n,ℓ,0)-Sharing [Independent]	Yes	Yes	Yes	Yes
(n,8,3)-Sharing [SC-1]	Yes	Yes	Yes	No
(n,5,3)-Sharing [SC-0]	Yes	No	No	No

Usable + Insecure

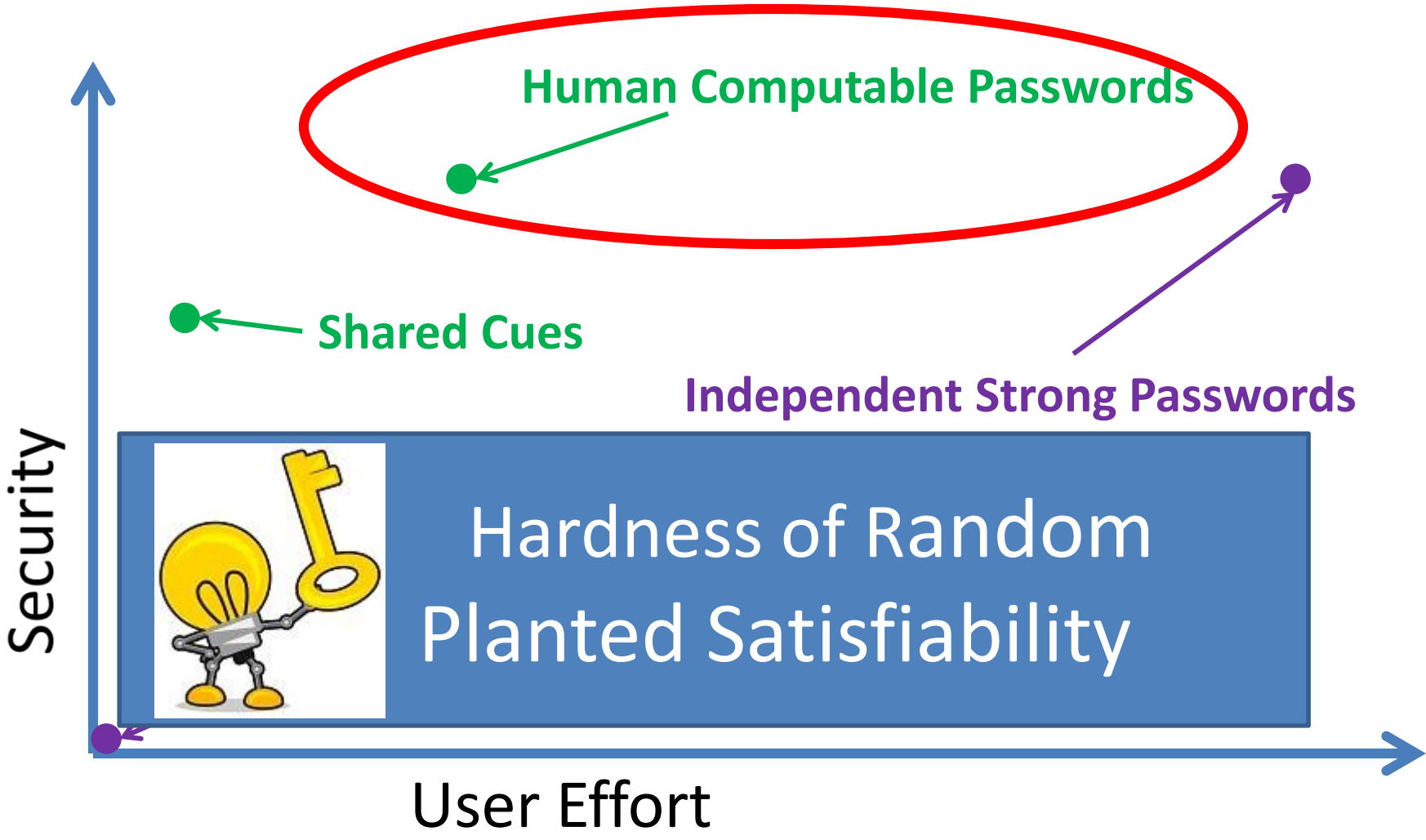
Unusable + Secure

Usable + Secure

Usable + Secure

$(10^{10}, \delta, m, 3, r, h)$-security

Preview of Results



Outline

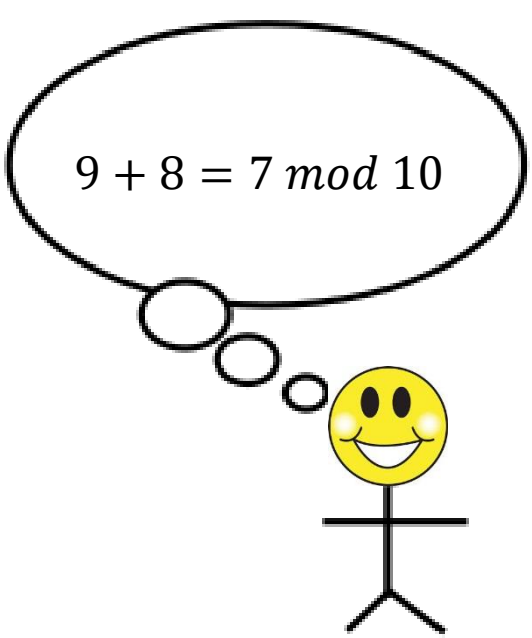
- Introduction
- Usability and Security Models
- Shared Cues
- **Human Computable Passwords**
- Conclusion and Future Vision

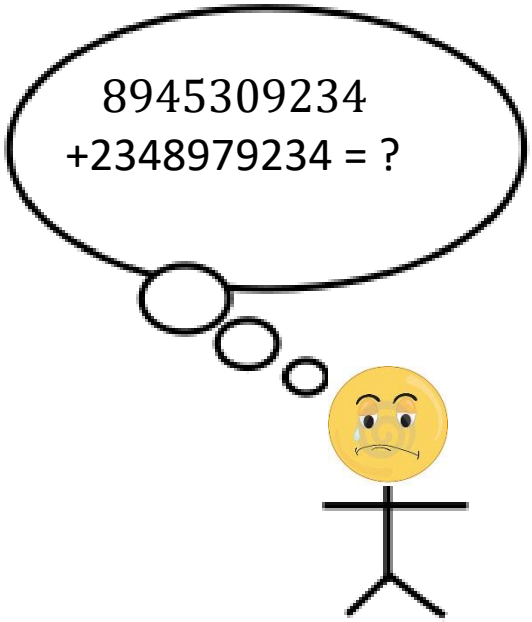
Our Scheme: Human Computable Passwords

- Remains secure many breaches (e.g., 100)
  - Heartbleed
- Passwords computed by responding to public challenges
  - Computation done in user's head
- Required Operations
  - Addition modulo 10
  - Memorize a random mapping

Human Computation

- Restricted Capabilities
  - Simple operations (addition, lookup)
  - Operations performed in memory (limited space)





Random Mapping

Image I			…	
□(I)	9	3	…	6

Initialization:

User Memorizes Random Mapping

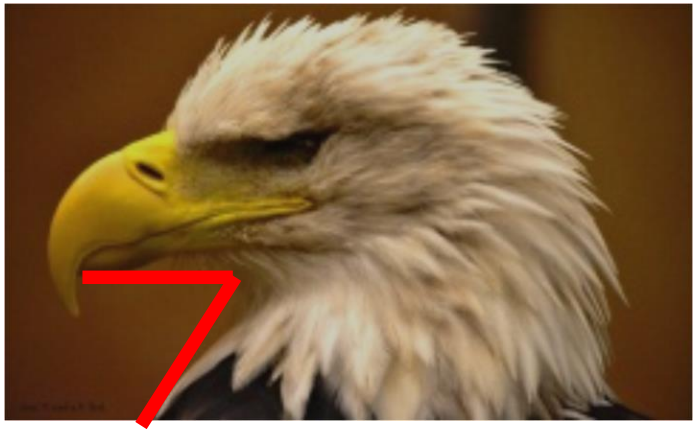
□: $\{I_1,\ldots,I_n\} \rightarrow \{0,1,\ldots,9\}$

Example: n=30 images

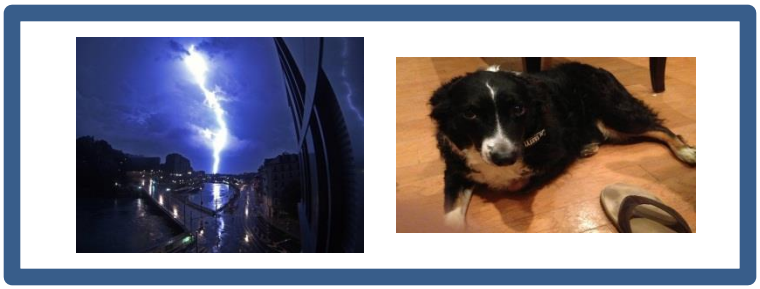
Mnemonics

?  = 7

Instruction: Trace the eagles body from the bottom of the eagle's beak down to the bottom of the picture. It looks like the number 7.



Single-Digit Challenge



Computing the Response:

= 9+3 mod 10 = 2

0

1

2

3

4

5

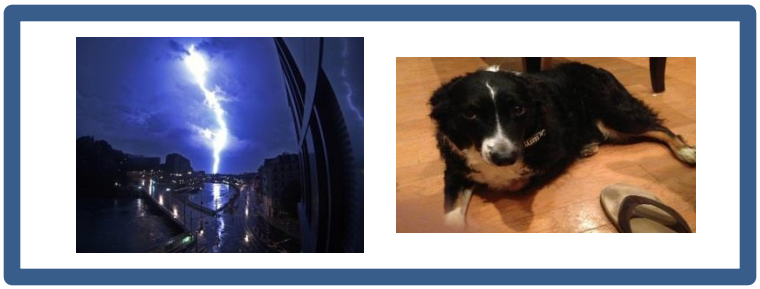
6

7

8

9

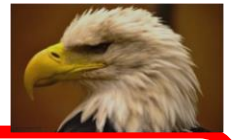
Single-Digit Challenge



Response:

? () + ? () mod 10

= 9+3 mod 10 = 2

0		5	
1		6	
2		7	
3		8	
4		9	

Single-Digit Challenge




Final Response:

?  + ?  + ? 

= 7 + 4 + 5 mod 10 = 6

0 
1 
2 
3 
4 
5 
6 
7 
8 
9 

Passwords




Username: jblocki

Password:

0 

1 

2 

3 

4 

5 

6 

7 

8 

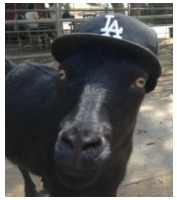
9 

Passwords




Username: jblocki

Password: *

0 

1 

2 

3 

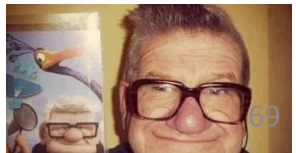
4 

5 

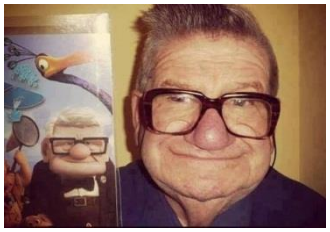
6 

7 

8 

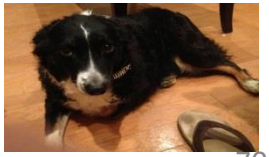
9 

Passwords




Username: jblocki

Password: **

0		5	
1		6	
2		7	
3		8	
4		9	

SAT Solver Attack

Secret Length (N)	#Leaked Challenge Response Pairs (M)					
	M=50	100	300	500	1000	10000
N=26	23.5 hr	40 min	4.5 hr	29 min	10 min	2 min
N=30	--------	**UNSLV**	2.3 hr	36 min	10 min	20 s
N=50	--------	--------	--------	--------	**UNSLV**	7 hr
N=100	--------	--------	--------	--------	--------	**UNSLV**

UNSLV – Solver did not find secret mapping in 2.5 days

-------- – Instance is harder than unsolved instance

Usability

Example Authentication Time:

- 7.5 seconds/digit
- 30 seconds for a 4-digit password
- 1.25 minutes for a 10-digit password

Memorizing the Secret Mapping:

- Memorized 100 image/digit pairs in 2.5 hours
- One Time Cost

Usability (Memorization)

	Human Computable Passwords			Shared Cues	
	N = 100	N = 50	N=30	SC-1	SC-0
Active	0.40	?0	?0	3.93	?0
Typical	2.14	?0.04	?0	10.89	?0
Occasional	2.50	?0.05	?0	22.07	?0
Infrequent	70.7	?22.3	?6.1	119.77	2.44

E[$\mathbf{X}_{365}$]: Extra Rehearsals to maintain *all* passwords over the first year.

Theoretical Security Guarantees

Thm (Informal): Any statistical algorithm needs to see at least $m = \tilde{O}(n^{1.5})$ passwords before it can even approximately guess the secret mapping σ.

Example: n=30 images

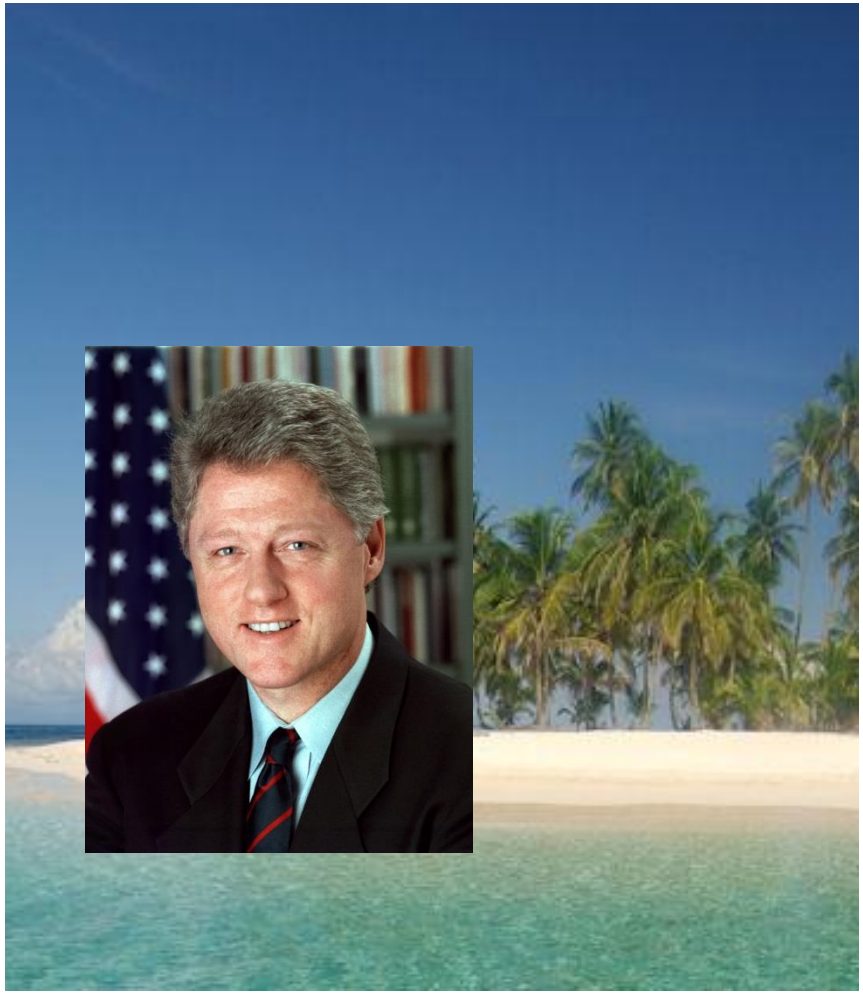
Security

Thm (Informal): Any statistical algorithm needs to see at least $m = \tilde{O}(n^{1.5})$ passwords before it can even approximately guess the secret mapping σ.

Thm (Informal): Any polynomial time adversary needs to see $m = \tilde{O}(n^{3})$ passwords before he can use Gaussian Elimination to approximately guess the secret mapping σ.

Thm (Informal): Any polynomial time adversary who can guess the user's passwords with accuracy much better than random guessing can also approximately recover the secret mapping σ.

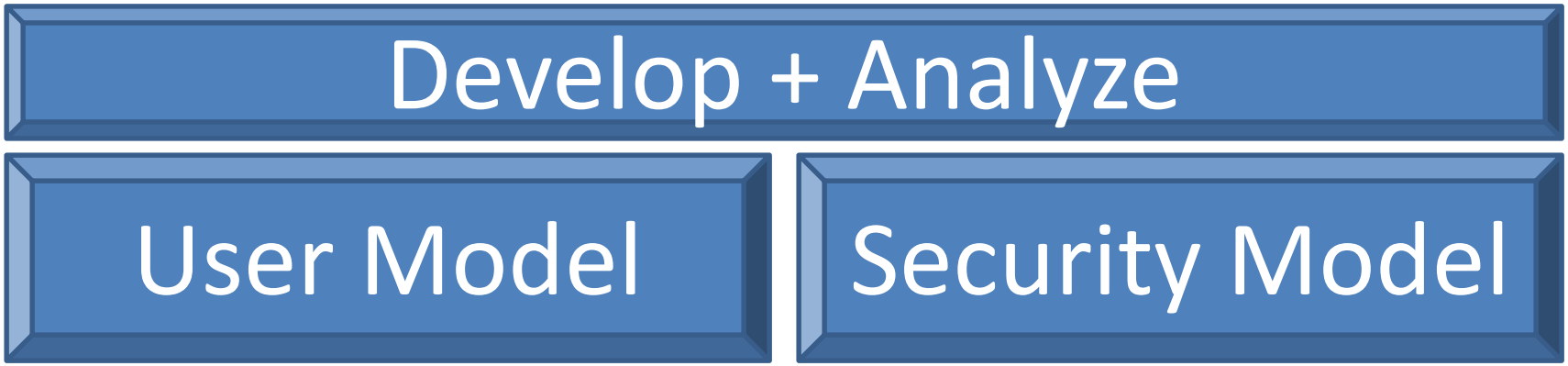
Memory Experiment 1



Memory Experiment 2

Our Thesis

User models and security models can guide the development of human authentication schemes with analyzable usability and security properties.

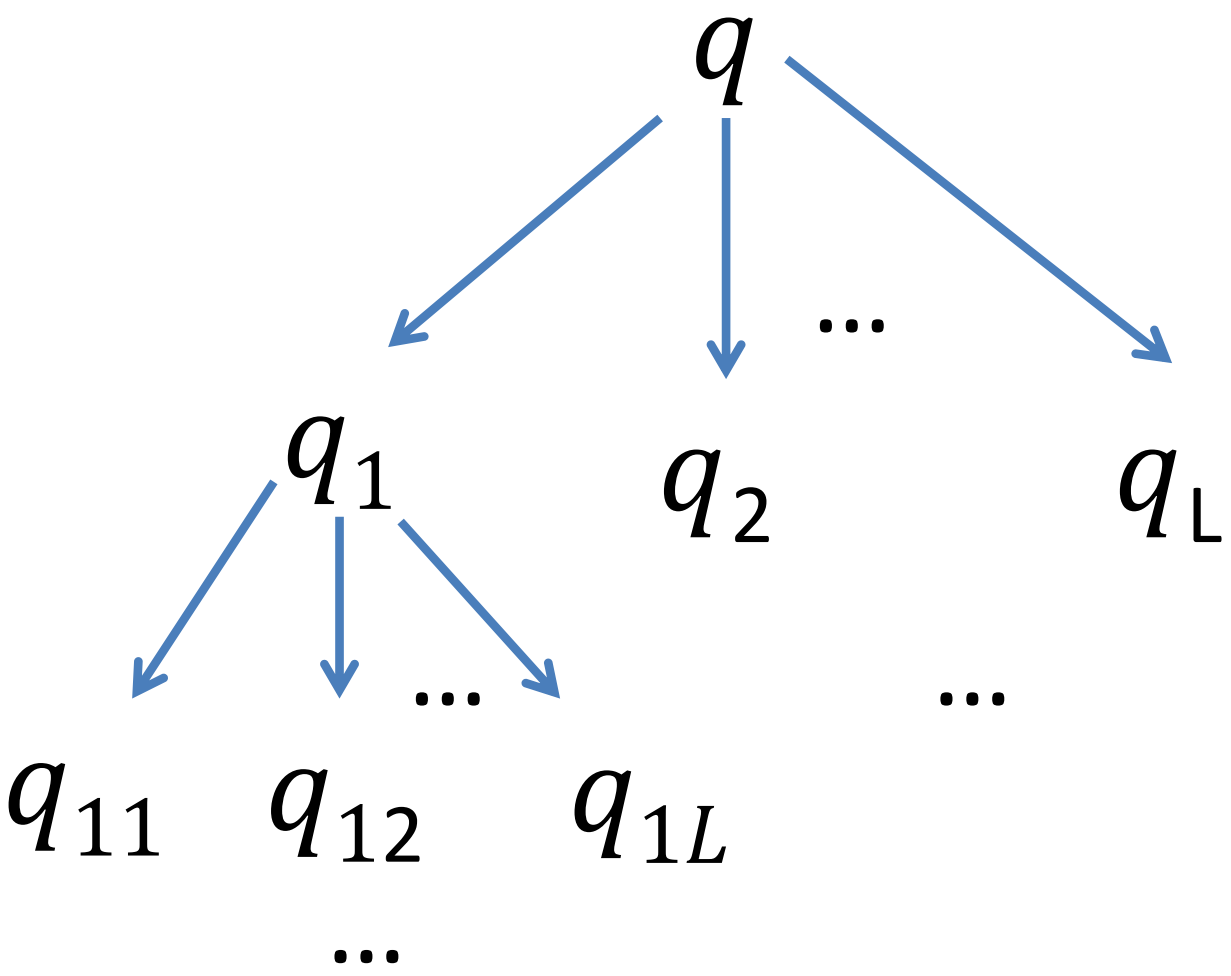


Thanks for Listening!

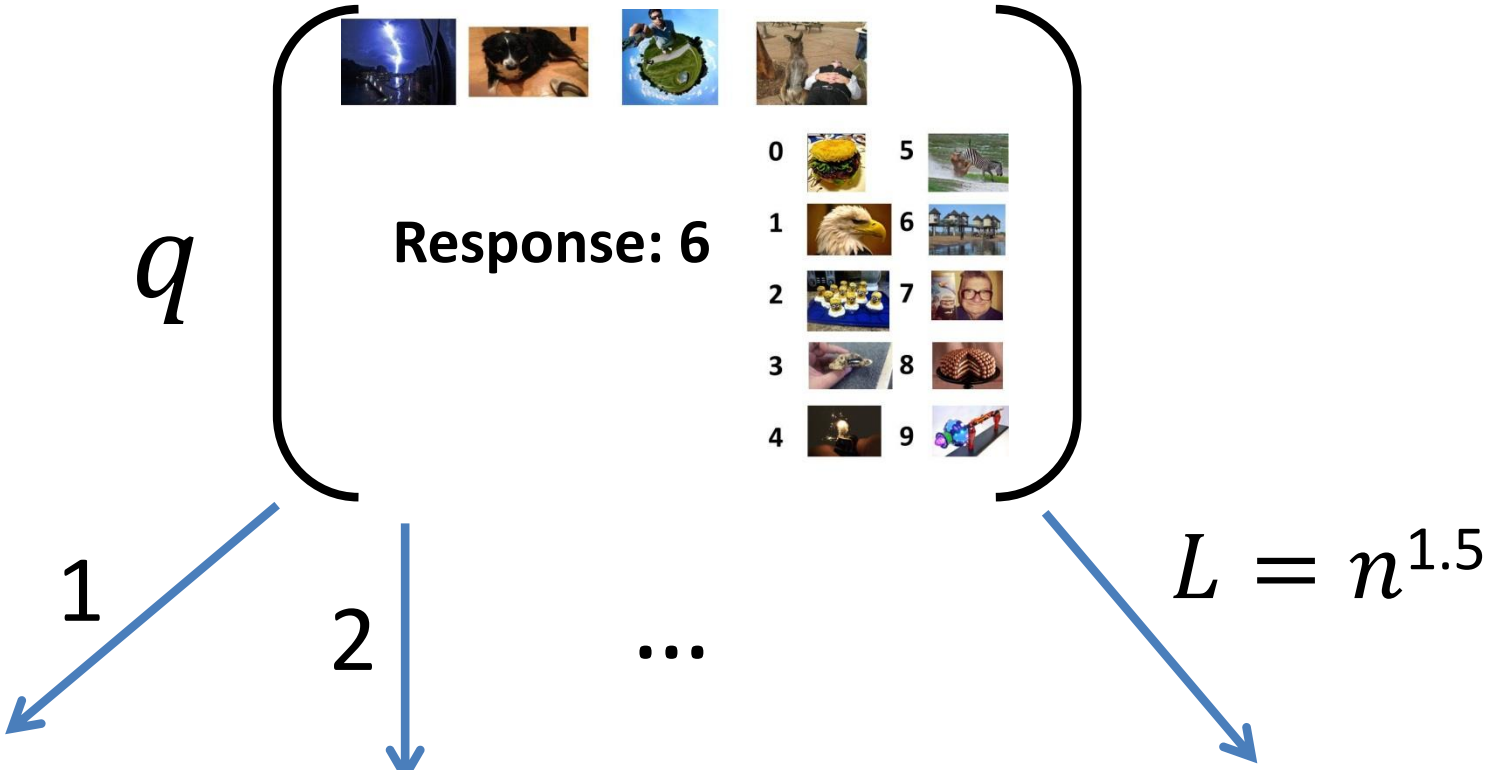
Technical Tools

- Discrimination Norm
  - On average how much different would the answers to a query q be if we picked a random challenge and a random response?
  - Small discrimination norm => Statistical Algorithm must use deep tree. [FPV13]
- Fourier Analysis
  - Express discrimination norm as a low degree function
- Generalized Hypercontractivity Theorem
  - Bounds the expected value of low degree functions

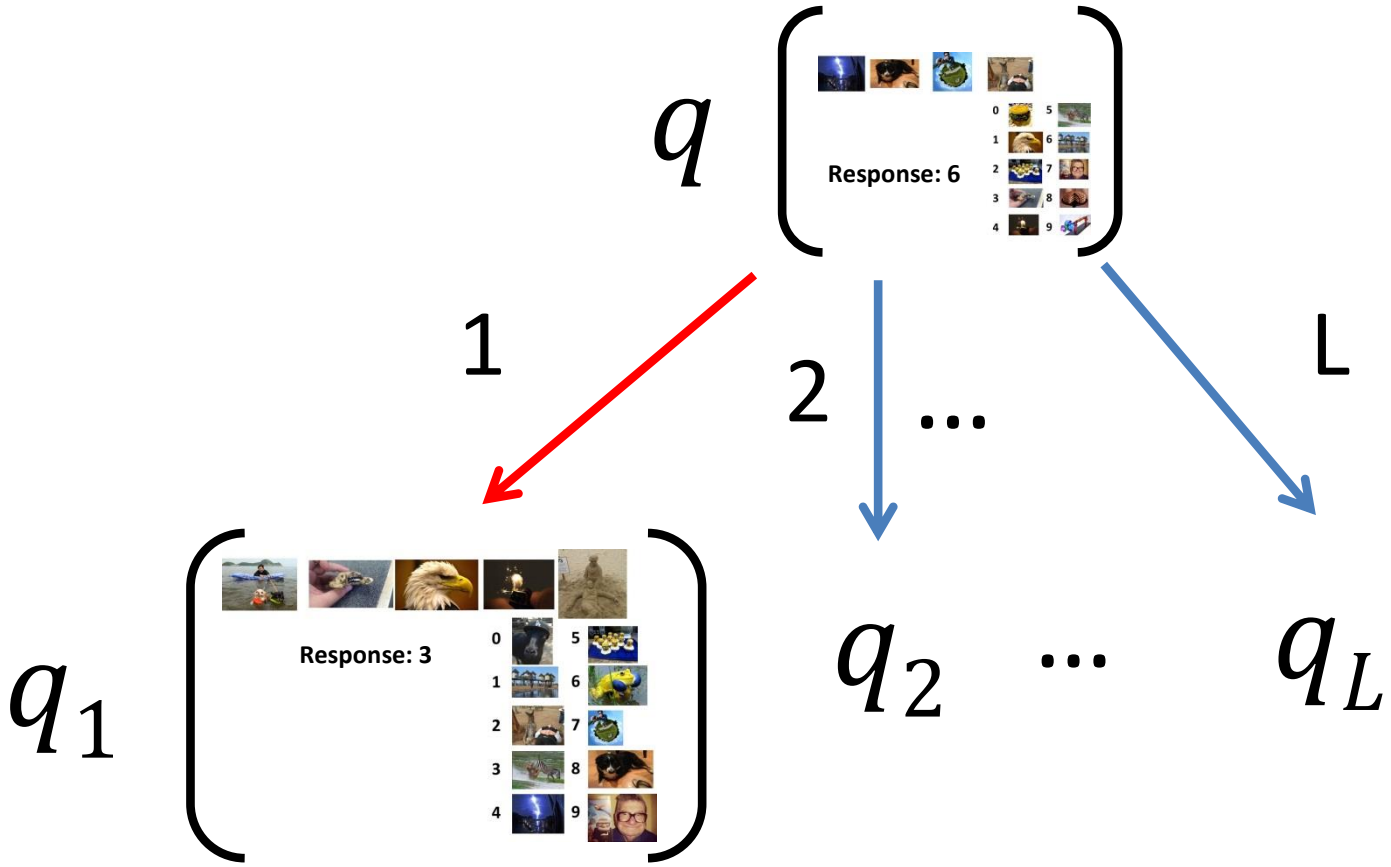
Statistical Algorithm



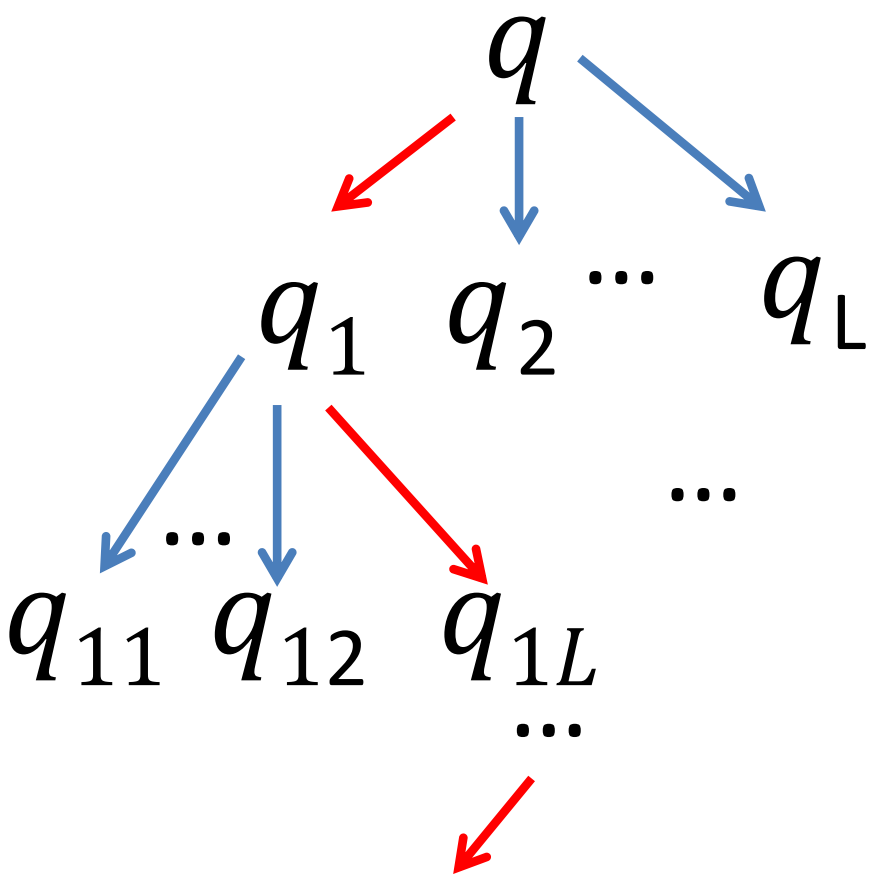
Statistical Algorithm



Statistical Algorithm



Statistical Algorithm



Security

Thm (Informal): Any statistical algorithm needs to see at least $m = \tilde{O}(n^{1.5})$ passwords before it can even approximately guess the secret mapping σ.

Example: n=30 images

Almost all known algorithmic techniques

Spectral Methods

Local Search

Expectation Maximization

First and Second Order Methods for Convex Optimization

~~Gaussian Elimination~~