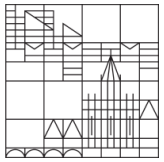
ACSAC 25

Universität Konstanz



HOCHSCHULE DER MEDIEN

NANYANG TECHNOLOGICAL UNIVERSITY

On the Security of PAS
(Predicate-based Authentication Service)

Shujun Li[1], Hassan Jameel Asghar[2], Josef Pieprzyk[2],
Ahmad-Reza Sadeghi[3], Roland Schmitz[4], Huaxiong Wang[5]

[1]University of Konstanz, Germany
[2]Macquarie University, Australia
[3]Ruhr-University of Bochum, Germany
[4]Stuttgart Media University, Germany
[5]Nanyang Technological University, Singapore

10 December, 2009

Outlines

- Observer attacks: An unsolved problem

- PAS: A very recent solution (ACSAC'2008)
- Our finding: PAS ≈ OTP with lower usability
  - Brute-force attack
  - SAT (Satisfiability) attack
  - Random-guess attack
  - Usability
  - A probabilistic attack

- Summary / Take-home message

Universität Konstanz

MACQUARIE UNIVERSITY

HOCHSCHULE DER MEDIEN

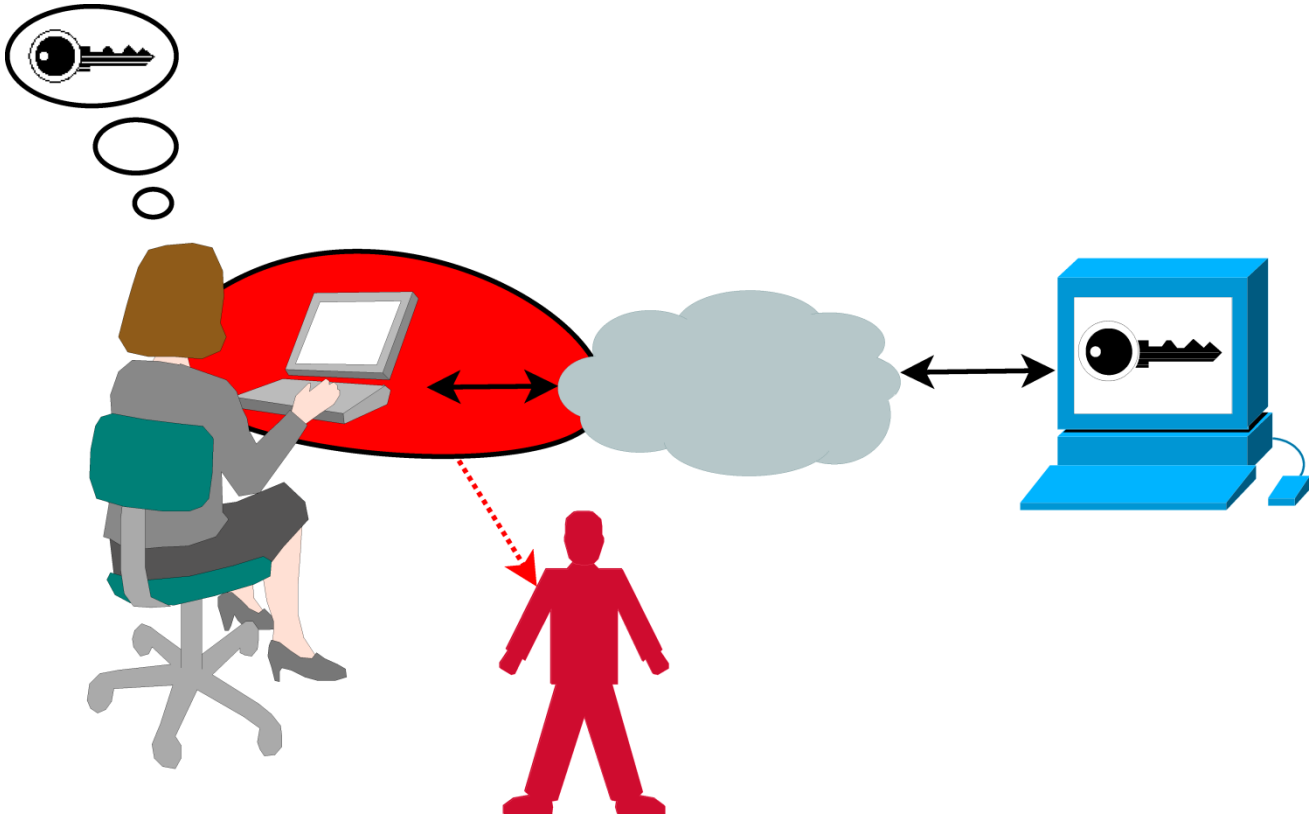
NANYANG TECHNOLOGICAL UNIVERSITY

Observers:
From Shoulder-Surfers to Malware

Observers: From shoulder-surfers to malware

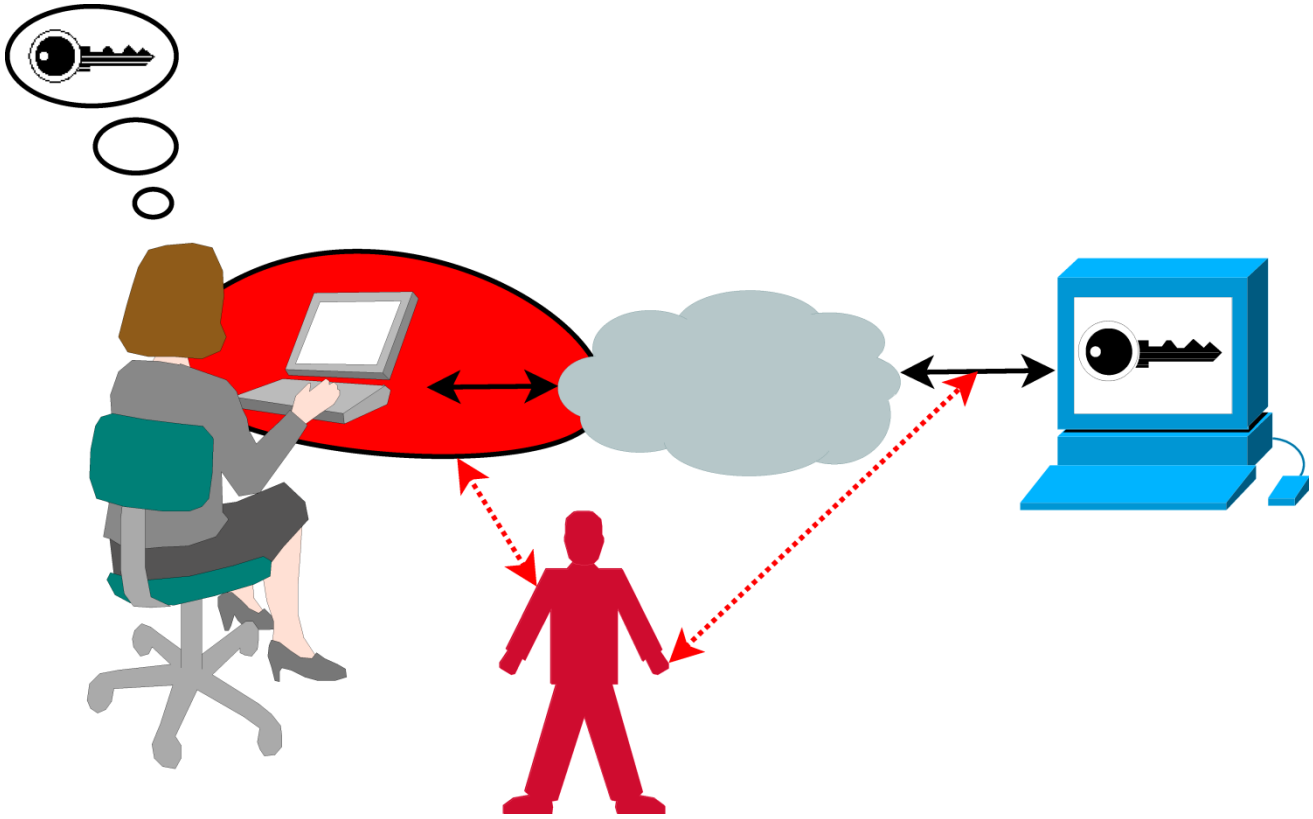
- Who?
  - Alice and Eve
- What?
  - Alice is typing her password.
  - Eve is looking at Alice's fingers.
- How?
  - Shoulder-surfer
  - Hidden camera
  - Malware
  - …

Matsumoto-Imai model (EuroCrypt'91): Passive adversary



Shoulder-surfers, keyloggers, network sniffers, TEMPEST – electronic/optic/acoustic emanations, …

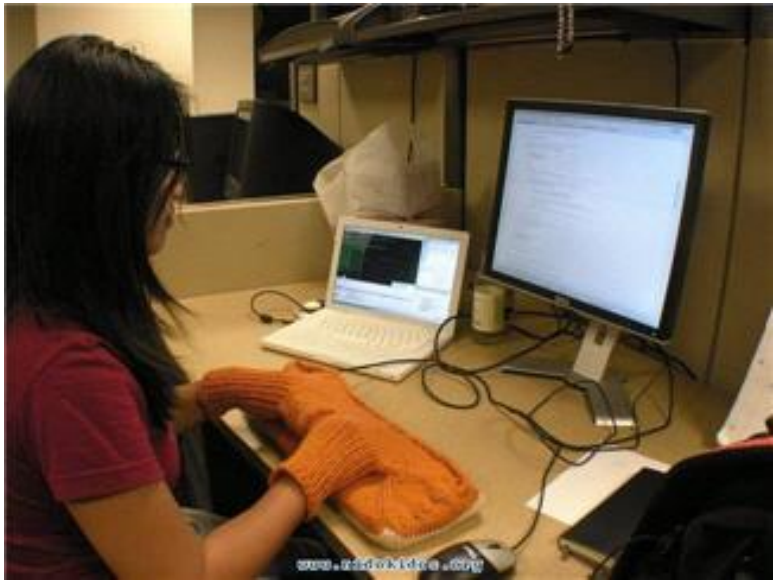
Matsumoto-Imai model (EuroCrypt'91): Active adversary



Man-in-the-middle, phisher/pharmer, Trojan horses, …

Existing user authentication systems vs. observers

- "What you know" authentication
  - Static passwords: not secure 
  - We need something beyond static passwords!



http://www.isgafrica.org/blog/?p=138

Existing user authentication systems vs. observers

- "What you have" = Hardware
  - OTP (One-time passwords) generators
  - Advanced user authentication protocols
  - Problems
    - Prone to theft and loss
    - Higher implementation costs
    - Worse portability

- "Who you are" authentication = Biometrics
  - You can't change your secret!
  - Privacy concerns
  - Higher implementation costs

An ideal solution?

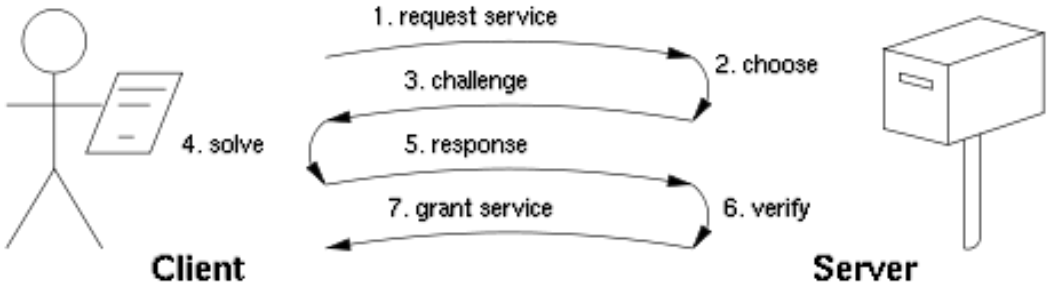
- “What you know” user authentication
  - Like static passwords
  - No additional hardware
  - No biometrics

  - Secure against passive observers
  - Secure against active observers
  - Secure against multiple (ideally, ∞) observations

  - Usable for most common users

"What you know" solution: Challenge-response protocol

- A secret *S* shared between prover/user (P) and verifier/server (V)



- Authentication is a challenge-response protocol
  - V $\Rightarrow$ P: t challenges $C_1(S)$, …, $C_t(S)$
  - P $\Rightarrow$ V: t responses $R_1=f_1(C_1,S)$, …, $R_t=f_t(C_t,S)$
  - V: Accept P if all the t responses are correct; otherwise reject P.

Observer attack to challenge-response protocol

- Observer attack 
  - Given n observed successful authentication sessions, solve the secret S.
  - $\{R_1^{(i)}=f_1^{(i)}(C_1^{(i)},S), \ldots, R_t^{(i)}=f_t^{(i)}(C_t^{(i)},S)\}$ $(i=1,\ldots,n) \Rightarrow S = ?$

- Possible measures against observers
  - Hiding responses R_i
    - Introducing noises/errors in response R_i
    - Postprocessing response R_i to bring ambiguity
  - Hiding (part of) challenges C_i
  - Complicating f_i
  - …

Some solutions against observers

- Matsumoto-Imai scheme (EuroCrypt'91)
  - NOT secure (Wang et al., EuroCrypt'95)
- Enhanced Matsumoto-Imai scheme (EuroCrypt'95)
  - Too complicated for users ⇒ NOT usable
- Matsumoto protocols (CCS'96)
  - NOT secure (Hopper & Blum 2001; Li & Shum 2003)
- Hopper-Blum protocols (AsiaCrypt'2001)
  - NOT usable (166 seconds for login)
- Cognitive Authentication Scheme (S&P'2006)
  - Neither usable nor secure (S&P'2007)
- Undercover (CHI'2008)
  - Not secure with untrusted computer
- …

A still unsolved problem…

- Challenge 1: Security vs. Usability
- Challenge 2: Weak humans vs. Powerful attackers

- The question
  - Is it possible to design a really practical hardware-free solution against a large number of observations?
  - Practical = Comparable to textual passwords

Universität Konstanz

MACQUARIE UNIVERSITY

HOCHSCHULE DER MEDIEN

NANYANG TECHNOLOGICAL UNIVERSITY

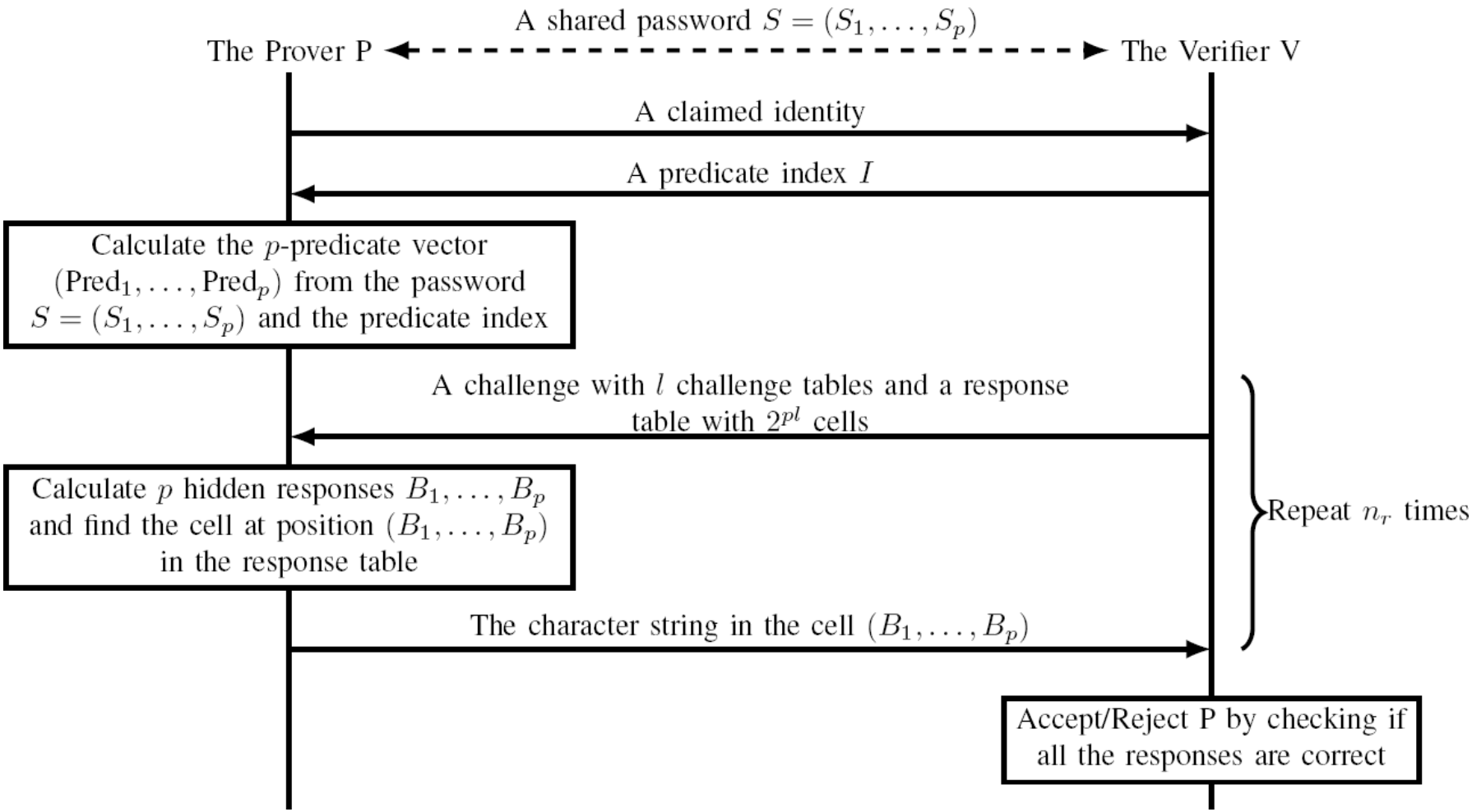
PAS

What is PAS?

- PAS = Predicate-based Authentication Service
- Presented at ACSAC'2008 by Bai et al.

- A solution claimed to be secure against passive observers
- User study was done to show usability
- Password renewal after a number of successful logins

The big picture of PAS: A challenge-response protocol



Password and predicates in PAS

- Password = (S_1, …,S_p)
  - S_i=((u_i,v_i),W_i=w_i[1]…w_i[*len*]), 1≤u_i≤m, 1≤v_i≤n, and w_i[j]∈$\mathbb{H}$
  - Example (p=2, *len*=10, m=n=5, $\mathbb{H}$={A,…,Z}):
    S_1=((1,1), CATCHINGME), S_2=((5,5), BEATINGHIM)

- Predicates
  - Given an integer as the predicate index I, 1≤I≤*len*
  - $S_i \Rightarrow$ Pred$_i$=((u_i,v_i), h_i=w_i[I])
  - Example (I=2):
    Pred$_1$=((1,1), A), Pred$_2$=((5,5), E)

Password and predicates in PAS

- Main secret (Password)
  - S_1=((1,1),CATCHINGME), S_2=((5,5),BEATINGHIM)
- Secret used for login (Predicate)

	SP = Static Passwords	OTP = One-Time Password	PAS = SP + OTP?	
Login #1	Honolulu2009	561072	$Pred_1$=((1,1),A) $Pred_2$=((5,5),E)	I=2
Login #2	Honolulu2009	860241	$Pred_1$=((1,1),E) $Pred_2$=((5,5),M)	I=10
Login #3	Honolulu2009	752081	$Pred_1$=((1,1),G) $Pred_2$=((5,5),H)	I=7
…	…	…	…	…

Challenges in PAS

- *l* challenge tables (CTs) and one response table (RT)

Responses in PAS

- p predicates $\Rightarrow$ p hidden responses
  - $\text{Pred}_i = ((u_i, v_i), h_i) \Rightarrow B_i = b_i[1] \ldots b_i[l]$, where
  - $b_i[j]$="Yes" if h_i in Cell (u_i, v_i) of the jth CT, else $b_i[j]$="No".

- Final response
  - $(B_1, \ldots, B_p) \Rightarrow$ A cell in RT $\Rightarrow$ A CAPTCHA string

Let's see a simple example…

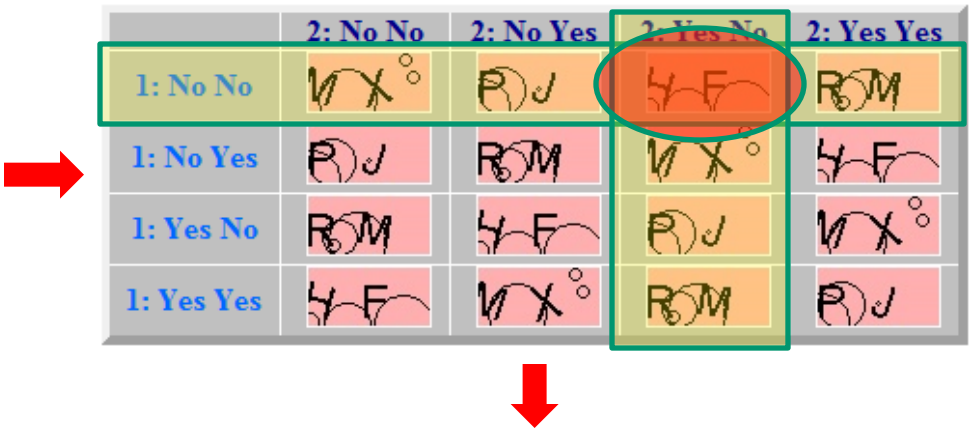
- Password
  - S_1=((1,1), CATCHINGME), S_2=((5,5), BEATINGHIM)
- Predicates
  - $Pred_1$=((1,1), A), $Pred_2$=((5,5), E)

Sub-response	Value
$b_1[1]$	No
$b_1[2]$	No
$b_2[1]$	Yes
$b_2[2]$	No

Let's see a simple example…

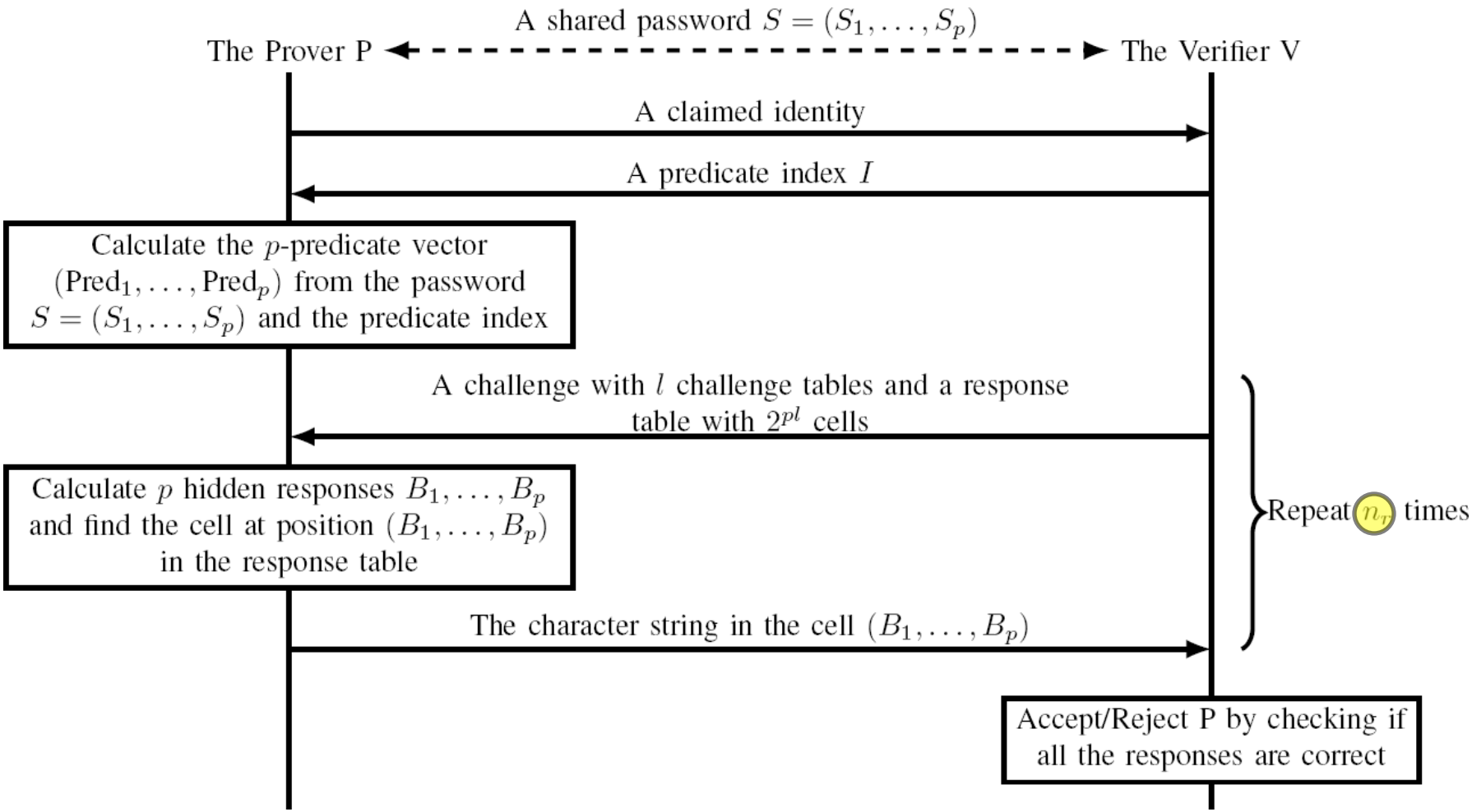
- Password
  - S_1=((1,1), CATCHINGME), S_2=((5,5), BEATINGHIM)
- Predicates
  - $Pred_1$=((1,1), A), $Pred_2$=((5,5), E)

Sub-response	Value
$b_1[1]$	No
$b_1[2]$	No
$b_2[1]$	Yes
$b_2[2]$	No



Final response = HF

The big picture of PAS: A challenge-response protocol



Extended PAS

- One cell index $\Rightarrow$ k ones: $S_i=((u_{i,1},v_{i,1}),\ldots,(u_{i,k},v_{i,k}), W_i)$
  - *k=2:*
    S_1=((1,1),(3,2), CATCHINGME)
    S_2=((5,5),(4,1), BEATINGHIM)
- One predicate index $I \Rightarrow k$ predicate indices $I_1,\ldots,I_k$
- For each sub-password S_i:
  - k cell indices $\Rightarrow$ k hidden sub-responses $\Rightarrow$ A final hidden response B_i
- …

Claimed security

- Security against three attacks
  - Each predicate is used for *t* login sessions, *M=mn*

Attack Type	***Password***	***Predicate***	***Response***
Brute Force	$M^{pk} H^{p \cdot len}$	NA	NA
Random Guess	$M^{pk} H^{p \cdot len}$	$M^{pk} H^{p \cdot len} / (k!)^p$	2^{ln_r}
SAT	$\left(M\left(1-(1-\frac{1}{M})^N\right)^{len/k}\right)^{pk} H^{p \cdot len}$ $N = pk(MH)^{pk} / (2^{ln_r} t (k!)^p)$	$\left(M\left(1-(1-\frac{1}{M})^N\right)^{len/k} H\right)^{pk} / (k!)^p$	NA

- Number of times a password can be used
  - For the default parameter, at least 10 logins (or, each predicate can be used for at least one login)

Claimed usability

- Basic PAS (k=1) with the default parameters
  - p=l=2, m=n=5, len=10, I=2, $\mathbb{H}$={A,…,Z}, n_r=2~5
- Average time for calculating a predicate from I
  - 35 seconds
- Average time for responding each challenge
  - 8.37 ~ 10.5 seconds
- Average login time when n_r=5: 84 seconds

- $\Rightarrow$ PAS outperforms previous work

Universität Konstanz

MACQUARIE UNIVERSITY

HOCHSCHULE DER MEDIEN

NANYANG TECHNOLOGICAL UNIVERSITY

Our Finding:
PAS ≈ OTP with Lower Usability

Re-evaluating security

- Brute-force attack: Predicate space exists!
  - NA ➡ $1 + \left(\binom{MH+k-1}{k}^{p} - 1\right) \Big/ 2^{ln_r t}$ (413.6)
- Brute-force attack: Password space over-estimated!
  - Influence of *t* is missing
  - Password space = Union of predicate spaces
  - $\underset{2^{103}}{M^{pk} H^{p \cdot len}}$ ➡ $\underset{2^{22}}{\left(1 + \left(\binom{MH+k-1}{k}^{p} - 1\right) \Big/ 2^{ln_r t}\right) \frac{len!}{(len-k)!}}$
- SAT attack: Security over-estimated!
  - Password: $\underset{2^{103}}{\left(M\left(1-\left(1-\frac{1}{M}\right)^{N}\right)^{len/k}\right)^{pk} H^{p \cdot len}}$ ➡ $\underset{2^{22}}{\left(1 + \left(\binom{MH+k-1}{k}^{p} - 1\right) \Big/ 2^{ln_r t}\right) \frac{len!}{(len-k)!}}$
  - Predicate: $\underset{625}{\left(M\left(1-\left(1-\frac{1}{M}\right)^{N}\right)^{len/k} H\right)^{pk} / (k!)^{p}}$ ➡ $\underset{413.6}{1 + \left(\binom{MH+k-1}{k}^{p} - 1\right) \Big/ 2^{ln_r t}}$

Re-evaluating security

- Random-guess attack to password/predicates
  - Handled improperly by Bai et al.
    - Bai et al.'s meaning:
      Similar to brute-force attack ⇒ Number of possible passwords/predicates
    - Our understanding:
      Randomly guessing the password/predicate/response ⇒ What is the success probability?
  - Our results

$$\frac{M^{pk}H^{p\cdot len}}{M^{pk}H^{p\cdot len}/(k!)^p} \rightarrow 1 \Big/ \left(\frac{1}{2^{ln_r}} + \frac{(2^{ln_r}-1)}{2^{ln_r}\binom{MH+k-1}{k}^p}\right) < 2^{ln_r}$$

$$\frac{2^{103}}{} \qquad\qquad < 2^{10} \quad (*)$$

* Shown as reciprocals of the success probability, to make it comparable with the original results.

Re-evaluating security

- Security analysis of Bai et al.

Attack Type	***Password***	***Predicate***	***Response***
Brute Force	$M^{pk} H^{p \cdot len}$	NA	NA
Random Guess	$M^{pk} H^{p \cdot len}$	$M^{pk} H^{p \cdot len} / (k!)^p$	2^{ln_r}
SAT	$\left(M\left(1-(1-\frac{1}{M})^N\right)^{len/k}\right)^{pk} H^{p \cdot len}$ $N = pk(MH)^{pk} / (2^{ln_r t}(k!)^p)$	$\left(M\left(1-(1-\frac{1}{M})^N\right)^{len/k} H\right)^{pk} / (k!)^p$	NA

Re-evaluating security

- Our security analysis

Attack Type	***Password***	***Predicate***	***Response***
Brute Force / SAT	$\left(1+\left(\binom{MH+k-1}{k}^{p}-1\right)\Big/ 2^{ln_r t}\right)\frac{len!}{(len-k)!}$	$1+\left(\binom{MH+k-1}{k}^{p}-1\right)\Big/ 2^{ln_r t}$	NA
Random Guess*	$1\Big/\left(1/2^{ln_r}+(2^{ln_r}-1)\Big/\left(2^{ln_r}\binom{MH+k-1}{k}^{p}\right)\right) < 2^{ln_r}$		2^{ln_r}

Re-evaluating usability

- Basic PAS vs. Low-complexity CAS (S&P'2006)
  - Average login time: 84 seconds vs. 90 seconds
  - Security against random-guess attack: 2^{10} vs. 2^{20}~2^{25}
  - Life span of password: <12 logins vs. <10 logins
  - ⇒ Does basic PAS outperforms CAS or NOT?

- Extended PAS
  - > 2.8 minutes for login
  - …
  - ⇒ NOT usable

A probabilistic attack

- Some observations
  - All predicates include the same cell index (u_i, v_i)
  - Predicates may be exhaustively searched

$$1 + \left(\binom{MH+k-1}{k}^{p} - 1 \right) \Big/ 2^{ln_r t} \approx 413.6$$

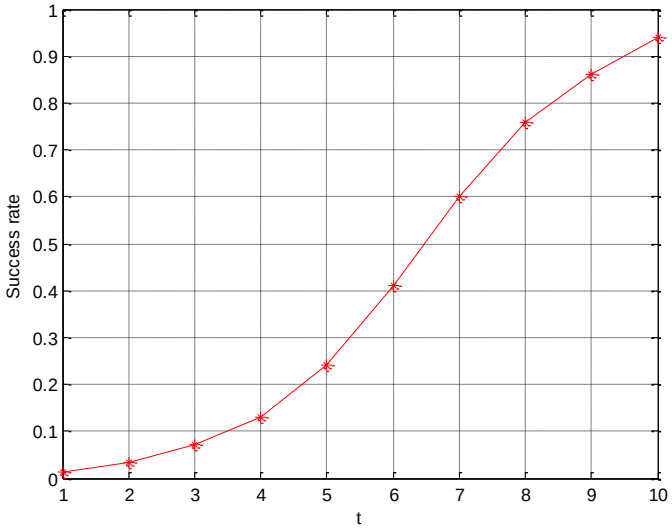
- The basic idea
  - For each login session, search the corresponding predicates $\Rightarrow$ A set of candidate p-predicate vectors
  - Refine all the predicate sets by matching the common cell index
  - $\Rightarrow$ Part of the password can be broken!

A probabilistic attack

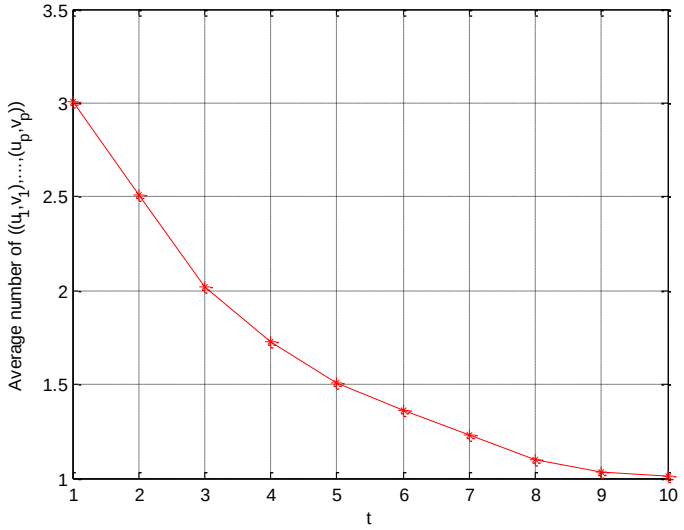
- The attack (given $\hat{t}$ observed login sessions)
  - Step 1: Derive $\hat{t}$ predicate sets $\mathbb{P}_1$, …, $\mathbb{P}_{\hat{t}}$
  - Step 2a: Extract the cell indices from $\mathbb{P}_i$ to get $\mathbb{C}_i$
  - Step 2b: $\mathbb{C}^* = \bigcup_{i=1}^{\hat{t}} \mathbb{C}_i$
  - Step 2c: $\mathbb{P}_i^* = \{x | x \in \mathbb{P}_i \wedge$ cell indices of $x \in \mathbb{C}^*\}$
  - Step 3a: If $|\mathbb{C}^*|=1$, the cell indices is determined and $\mathbb{P}_i^*$ includes candidates of h_i.
  - Step 3b: If $|\mathbb{C}^*|>1$, count the occurrence frequencies of different cell indices in $\{\mathbb{P}_i^*\}$, take the most frequent one as $((u_1,v_1),\ldots,(u_p,v_p))$ and then further refine $\{\mathbb{P}_i^*\}$.

A probabilistic attack: Complexity & experimental results

- Complexity: $O(\hat{t}(MH)^p) = O(2^{20.7})$
- Experimental results
  - A MATLAB implementation of the attack
  - $5\hat{t}$ seconds per attack (on a PC with 2.4GHz Intel Core2 Duo CPU and 2GB memory)



Success rate w.r.t. $\hat{t}$



Number of candidates w.r.t. $\hat{t}$

A probabilistic attack: Let's take a look at a real case

- Password
  - S_1=((1,1), CATCHINGME), S_2=((5,5), BEATINGHIM)

- A real attack with t=6 observed login sessions
  - (u_1,v_1): (1,1) – Completely Broken!
  - W_1: [CI]*[TZ]CH***[MV][EF] – Partially broken!
  - (u_2,v_2): (5,5) – Completely Broken!
  - W_2:[BU]*[AE]TI***[IE][MI] – Partially broken!

A probabilistic attack: What are the consequences?

- The cell indices can be uniquely broken with $t<10$ observed login sessions.
- $\Rightarrow$ Password becomes $\{S_i=W_i=w_i[1]\ldots w_i[len]\}$ ($i=1\sim p$)
- $\Rightarrow$ Equivalently, $\{S_j^*=W_j^*=w_1[j]\ldots w_p[j]\}$ ($j=1\sim len$)
  - Each S_j^* is used for one login session
  - Password is renewed after S_1^*, …, S_{len}^* are all used
  - $\Rightarrow$ PAS becomes an OTP-like system (S_j^* = OTP)
- Usability: PAS < OTP

- $\Rightarrow$ PAS $\approx$ OTP with lower usability

Summary
Take-Home Message

Summary / Take-home message

- Summary
  - PAS is not securer than OTP
  - PAS is less usable than OTP

- Take-home message: It is challenging to find a practical solution to observer attacks.
  - Challenge 1: Security vs. Usability
  - Challenge 2: Weak humans vs. Powerful attackers
  - Do we have to resort to trusted hardware?

Thanks for your attention!

It's time for arguing and criticizing ☺