

HumanAUT

Secure Human Identification Protocols

Adam Bender

Avrim Blum

Manuel Blum

Nick Hopper

The ALADDIN Center

Carnegie Mellon University



Background

- Concerned with designing secure human-executable authentication protocols
 - Authentication protocol: process for a human to prove his unique identity to a computer
 - Examples: typing password at prompt, keying in PIN at ATM



Background

- Need for such schemes comes from fact that all current authentication schemes have flaws
- Protocols need to withstand observation by adversaries, loss or compromise of any hardware (smart cards, biometrics, etc. vulnerable)



Challenge-response

- Computer asks a series of questions
- If answered correctly, user is authenticated, otherwise they are locked out
- Very common in practice as aids to recall forgotten passwords
- To be used for authentication, challenges cannot be repeated



HumanAUT

- HumanAUT is a challenge-response scheme with a shared secret between the human, who answers challenges using the secret, and the computer, which generates a unique random challenge on demand, such that the correct response depends on the shared secret



Previous scheme – Hopper

- Presented in ASIACRYPT '01
- Challenge is a large grid with n digits, shared secret is subset of size k
- Response is sum of these k digits mod 10



Previous scheme – Hopper

- Respond to 7 challenges, with exactly one response *incorrect*
 - Makes learning by observation NP-hard
- Takes $\binom{n}{k/2}$ work to break; for $n = 1000$, $k = 19$, work is $\approx 2^{78}$



Previous scheme – A. Blum

- Challenge is set of n bits, secret is two subsets of size $\log(n)$ ($\log(1000) = 10$)
- Response is mode (more common bit) of first subset \oplus parity (cumulative \oplus) of second subset
- Based on a difficult machine learning problem; hard to determine both sets
 - Predicting response \Leftrightarrow knowing both sets, if the mode bit is “noisy”



Producing potential protocols

- Tried to create protocol that combines security guarantee of Hopper with simple calculations of Blum
- Schemes had at best marginal improvement over Hopper
- No real efficient way to generalize bits to digits, other than to use something similar to Hopper



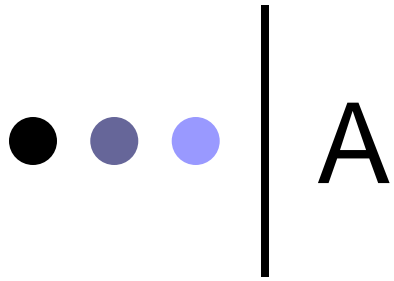
A more practical problem

- Proposed by M. Blum
- Want a protocol that is secure after three authentications are observed
 - Instead of any number
- Allows much simpler protocols, and would still be much more secure in practice than PINs



Letter mappings

- Each person has their own private mapping from letters to single digits
- 10^{26} possible random mappings, ≈ 86 bits of secret
- Possible to learn a random mapping in 5 minutes
 - Rue, “Eighty-six bits of memory magic”, 2002
 - Taught mapping to truly random digits



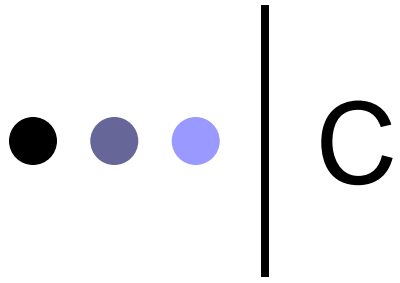
- A maps to 1

- It is the 1st letter of the alphabet

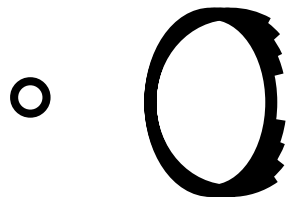
● ● ● | B

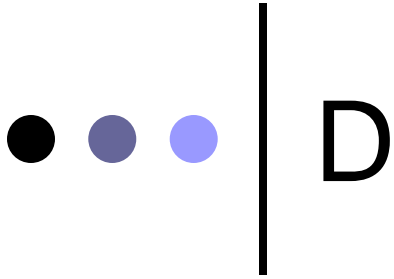
o B maps to 0 BOO!





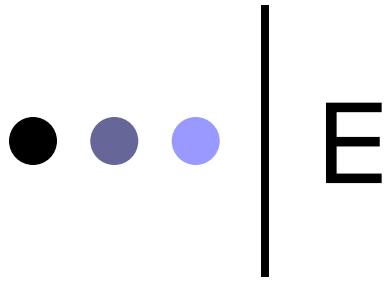
- C maps to 0



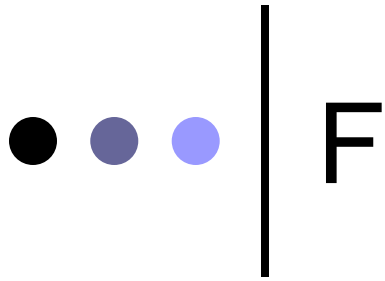


- D maps to 9
- Dressed to the 9's
(there's no D-9'ing it)





- E maps to 7
- sEvEn



- F maps to 3

- F is made of 3 lines - F

- ~~three~~



G

o G maps to 2



o Goody goody 2 shoes





Let's review

- A – 1 (1st letter of alphabet)
- B – 0 (B00)
- C – 0 (C → 0)
- D – 9 (dressed to the 9's, no D-9'ing)
- E – 7 (sEvEn)
- F – 3 (free → three)
- G – 2 (goody goody 2 shoes)



The protocol

- Challenge is a random 3-letter “word”
 - Each letter can only appear once in a challenge (or else information is leaked)
 - Thus there are $\binom{26}{3} = 2600$ challenges
- Response is sum of the digits that those 3 letters map to, taken mod 10
- Repeat 6 times, $1/10^6$ chance of guessing



Example

Challenge:

$$\begin{array}{ccc} F & B & A \\ \downarrow & \downarrow & \downarrow \\ 3 & + & 0 & + & 1 & = & 4 \end{array}$$

Challenge:

$$\begin{array}{ccc} G & E & D \\ \downarrow & \downarrow & \downarrow \\ 2 & + & 7 & + & 9 & = & 8 \end{array}$$



Security analysis

- Each challenge-response represents an equation in 3 of 26 variables
- Every authentication is a set of 6 linear equations, which can be treated as vectors
- These vectors then form a binary matrix, where each row has 3 1's and 23 0's



Security analysis

- Assume an attacker has observed some challenge-response pairs, and thus has formed such a matrix from the equations
- When could he predict the answer to a new challenge?



Independence

- He can determine the solution to a new challenge c if that challenge can be expressed in terms of the vectors of M
- That is, if c is not independent of M
- Goal: given a matrix M with n equations (rows), find how likely is it that a new challenge c is independent



Rank

- The *rank* of a matrix is the number of linearly independent rows in that matrix
- If $\text{rank}(M) < \text{rank}([M \ c])$, then c is independent of M

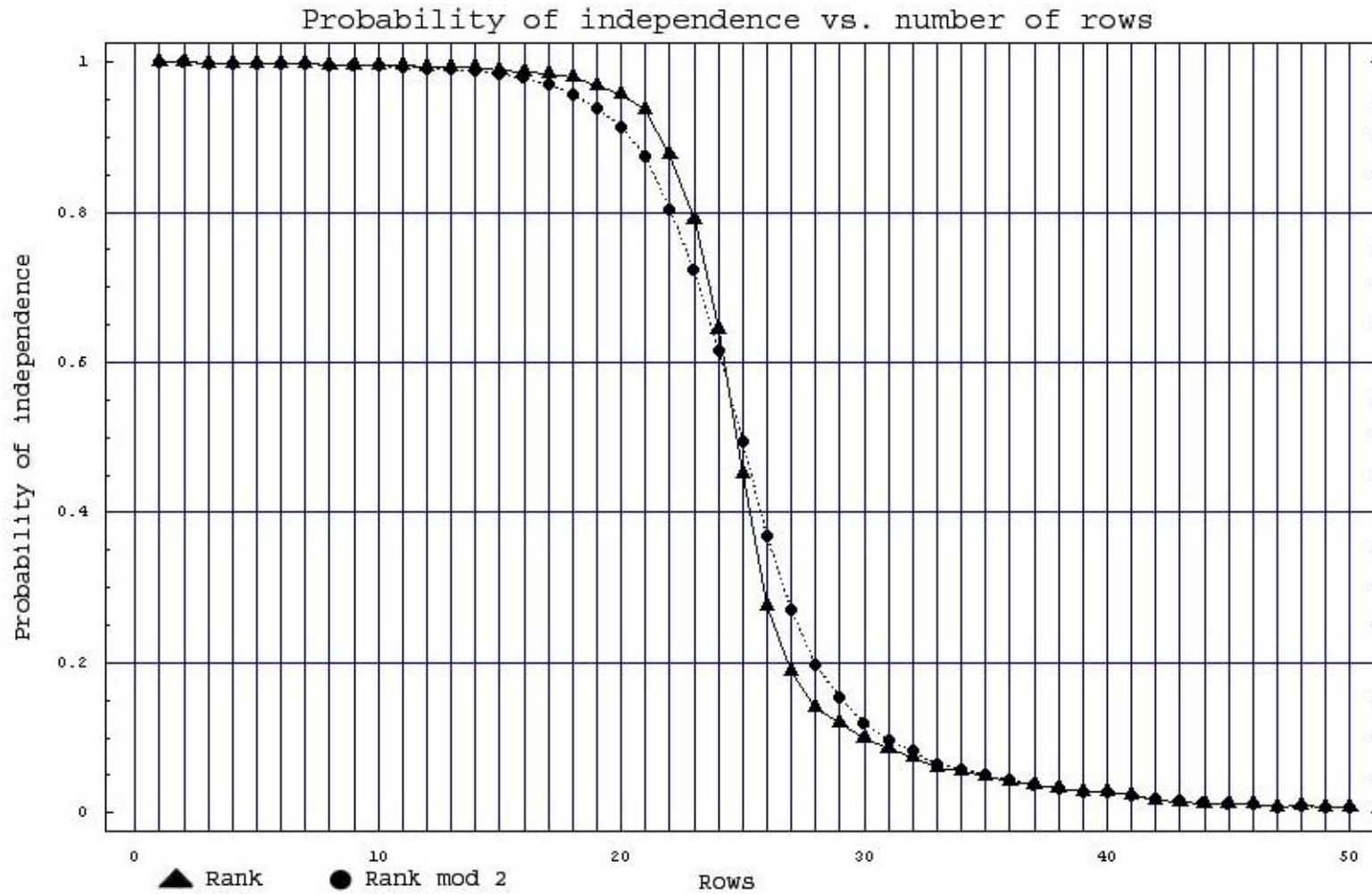


Calculations

- Use Mathematica to calculate $\Pr[\text{rank}(M) < \text{rank}([M \ c])] vs. n$, given random M with n rows and random c
- Get a nice graph
- For $n = 18$ (6 equations * 3 authentications), $\Pr[\text{independence}] \approx 98\%$, so very likely that adversary cannot predict correct response



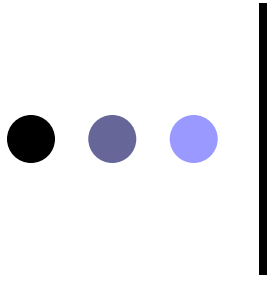
Probability of dependence





What I did on my summer vacation

- Did independent study (same topic) last spring
- Enjoyed the experience and learned a lot about research methods and such
- Convinced me that I should go to grad school





Previous scheme – A. Blum

- Challenge is set of n bits, secret is two subsets of size $\log(n)$
- Response is mode (more common bit) of first subset \oplus parity (cumulative \oplus) of second subset
- Based on a difficult machine learning problem; hard to determine both sets
 - Predicting answer is equivalent to knowing both sets



Proposed scheme

- Based on both of these previous schemes, as well as ideas from Charlie Garrod
- Started as a way to generalize Avrim's scheme from bits to digits
- Improvement is that it is easier and faster to execute



Proposed scheme

- Challenge is set of n digits
- Secret is two subsets, A and B , of size $\log(n)$
- m_A is mode of the parities (even or odd) of the digits in A (0 or 1), m_B is mode of the parities the digits in B
- If $m_A \oplus m_B = 0$, response is sum of digits in $A \bmod 10$, else response is sum of digits in $B \bmod 10$



Security analysis

- Each digit could affect the mode of parities (and thus the choice of which subset to add), and does affect the sum
- Thus learning membership in the secret subset has the same work complexity –

$$\binom{n}{\log(n)}$$

- For $n = 1024$, work $\approx 2^{78}$



Executability analysis

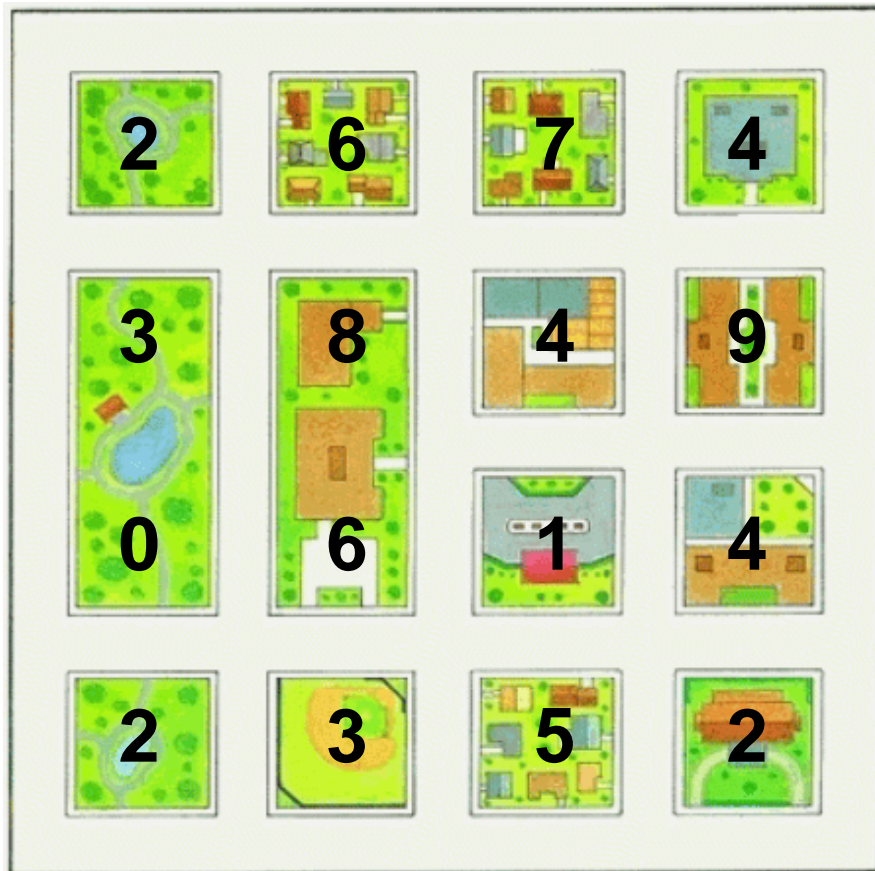
- Hopper's scheme:
 - 18 additions / response
 - 7 responses / authentication
 - 126 total additions
 - Guessing succeeds 1 out of $1.1 \cdot 10^6$ trials
- Proposed scheme:
 - 2 counts to 5, 9 additions / response
 - 6 responses / authentication
 - 54 total additions
 - Guessing succeeds 1 out of 10^6 trials



A consideration

- Subsets must have an odd number of elements or else there is a possibility of having a set where both parities occur the same number of times
- Could use 11 digit subsets (work factor increases to 2^{85}), or 9-digit and 11-digit subsets to have a secret of the same size

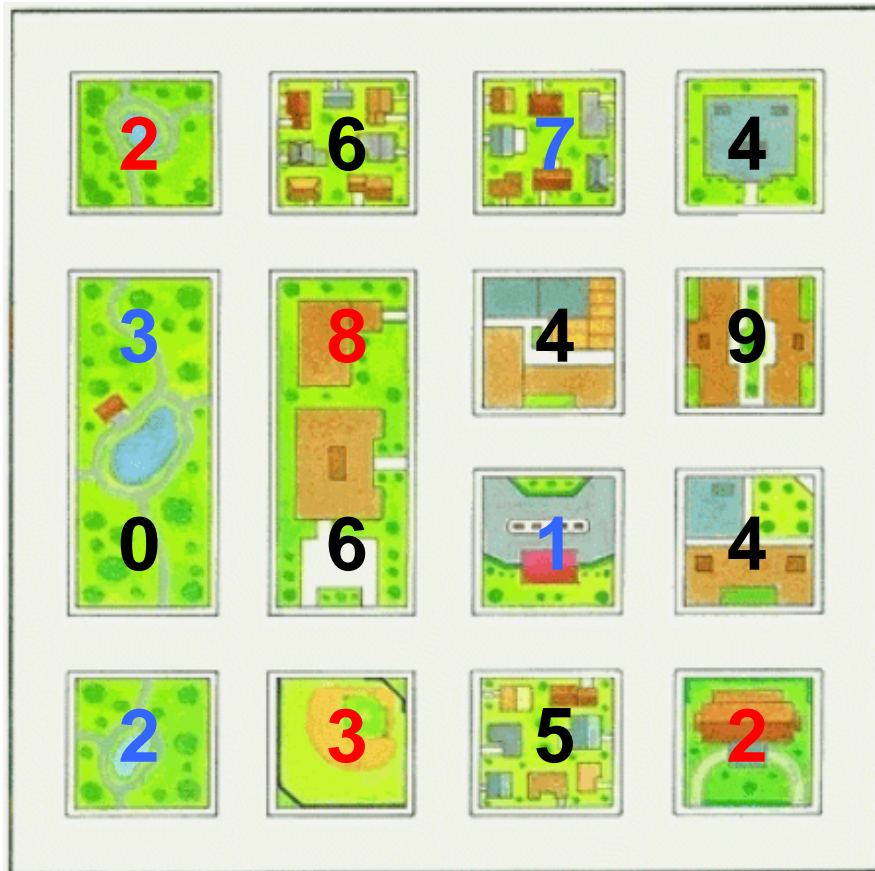
● ● ● | Presentation



Can be presented as a grid of digits on a map –
Easier for people to recall
Locations on a map

Secret subsets (A and B)
are randomly distributed

● ● ● | Presentation



Modes of parities are
even (0) and **odd** (1)

$0 \oplus 1 = 1$, so use **B**

Response is $7+3+1+2=3$



Modeling with equations

