Manuel Blum on Password Selection

David Mix Barrington
CMPSCI Theory Seminar
2 December 2014

Manuel Blum

nae.edu

- My grand-advisor, through Mike Sipser
- 30 Ph.D. students, 493 descendants
- Model-independent complexity theory
- Coin-flipping and poker by telephone
- First paper on CAPTCHAs
- 1995 Turing award

Naturally Rehearsing Passwords

- ASIACRYPT 2013 Paper by Jeremiah Blocki (student), Manuel Blum, and Anupam Datta
- Formalizes and evaluates widely-proposed password management schemes
- Quantifies usability in terms of number of rehearsals necessary to keep passwords memorized, given a human memory model
- Proposes new Shared Cues password management scheme

Conflicting Goals for Passwords

- **Usability**: We would like our passwords **easily available to our recollection**, ideally without writing them down.
- **Security**: We want some defense against **online attacks**, **offline attacks** (when a hash of the password is stolen), or **plaintext password leaks** (when one of our passwords is stolen).

Easily Usable Passwords

- Suppose I decide to use the string "axolotl" for all my passwords. No one has any reason to associate me with this Mexican amphibian.
- But that word is one of maybe 20,000 in a large dictionary, and one of about $(26)^8$ = 200 billion words of <= 8 lower-case letters.
- An offline cracker could try all those words, and even an online guesser has some chance of getting it if they are persistent.

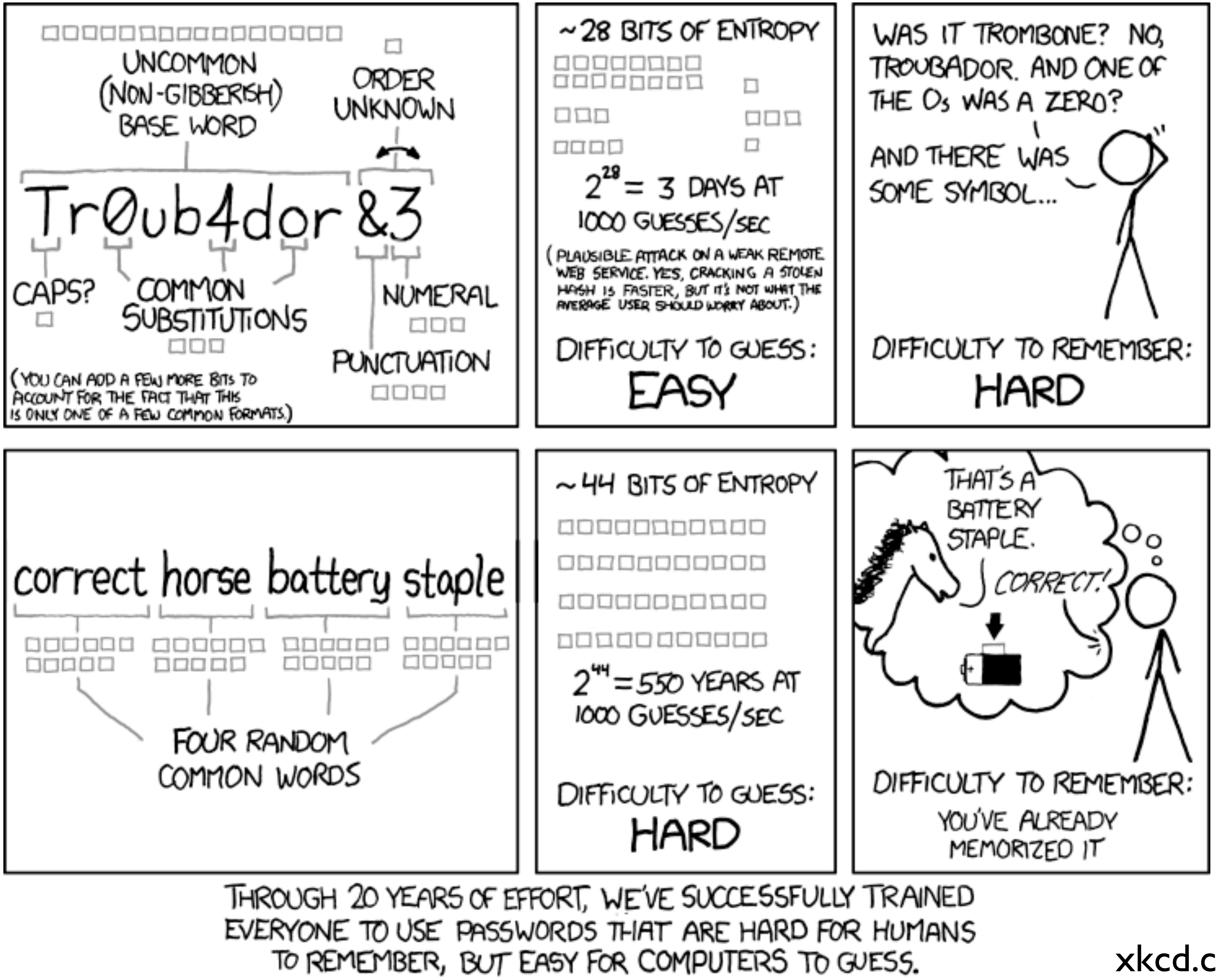
Quantifying Password Strength

- If we choose a password randomly from some set of size n, it has log n bits of **entropy**.
- A random string with digits, upper-case letters, and symbols has much more entropy than a string of the same length with just lower-case letters.
- Thus systems encourage, and often require, passwords with a variety of individual characters in them.

Quantifying Password Strength

- But a random string of such characters is very difficult to memorize.
- The passwords people actually use tend to be familiar words, with slight changes inserted to satisfy a system’s demand for a “stronger” password.
- I tried “axolotl” on passwordmeter.com and got a score of 7%, but “Axolotl7@@@@” got 100%.

You Knew This Was Coming


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

xkcd.com/936

Debate About xkcd Scheme

- Of course “correcthorsebatterystaple” is a bad password now, because it’s on every cracker’s list.
- But the idea is to choose four words randomly and independently yourself from a short dictionary, then memorize them yourself.
- Five words would be better against offline attacks.
- Many systems won’t take such a password, because it is too long, or because it doesn’t have digits or uppercase or symbols.
- Maybe “A*3corhorbatsta”?

Reusing Passwords

- So you’ve chosen and memorized a single strong password, and you use it for 100 different accounts.
- Then some department-store chain gets hacked and their stupid IT department was storing your great password in the clear.
- All of your accounts are now compromised.
- You can change passwords, but this makes the memorization a burden.

The Lifehacker System

- In 2006, Gina Trapani of lifehacker.com proposed a system for generating a distinct password for each account you use.
- If I used “dhg” for the Daily Hampshire Gazette, “fsb” for Florence Savings Bank, etc., I could easily remember these passwords, but they would be lousy passwords.
- Trapani proposes choosing a single base password, and combining it in some fixed way with the special string for each account.
- So I might use, say, “Axolotl7@@@@dhg”.

Assumptions for This Paper

- BBD assume that any paper or digital record of your passwords might be viewed or stolen.
- And as usual in cryptography, they assume that your enemy is completely aware of the overall system that you are using.
- This makes Lifehacker terrible against a plaintext password leak, because if they have one of your passwords and know you are using Lifehacker, they can easily guess others.

The Basic Idea

- Choose easily memorable, or even public, passwords for your different accounts.
- Then personally memorize a scheme to encrypt them, that you can carry out in your head when you need a password. This encryption needs to be chosen at random from a large family.
- Example: memorize a sequence of 26 digits, then encode the lower-case alphabet into it.

Human Memory Capability

- Could you memorize “12165621944567688583658559”?
- I know people who have memorized hundreds of digits of π, for reasons best known to themselves. But that’s unusual.
- Franklin Foer’s book *Moonwalking with Einstein* details some of the tricks used by human memory “athletes”, such as **memory palaces** and **person-action-object (PAO) stories**.

The Shared Cues Scheme

- Choose n people, n actions, and n objects.
- Randomly choose n PAO stories by choosing two permutations of {1,...,n}.
- Memorize these stories.
- Have a computer password manager assign a few stories to each account.
- When you want to use the account, the computer shows you a few people. You remember what they are doing to what. The answers to this provide the password.

Usability of Shared Cues

- As you continue to use passwords, you are also rehearsing your memory of the n stories.
- BBD’s system also makes sure that you also rehearse other random stories at the same time.
- You may need additional rehearsals as well, depending on how often you use the passwords.
- The number of additional rehearsals needed is BBD’s measure of usability.

Security of Shared Cues

- You are using one of $(n!)^2$ possible sets of PAO stories, chosen at random. They consider n = 9 (weak), 43 (strong), and 60 (really strong). These are not written down!
- If someone learns one of your passwords, they can reverse-engineer a few of your stories.
- But if the story sets are chosen from a **sharing set family**, knowing a few stories does not make it easy to get many other passwords.

Comparison of Schemes

- Their Shared Cues scheme SC-0, which uses only nine stories, is as easy to use as Lifehacker and can survive the loss of one plaintext password. But its low entropy makes it vulnerable to offline attack.
- SC-1 and SC-2, with n = 43 and 60 respectively, require a plausible number of extra rehearsals -- they are far easier to use than strong random independent passwords. But they remain secure against offline attacks and against theft of two *chosen* passwords.

What’s New Here?

- The particular scheme may or may not work for any particular person. Like CAPTCHAs, it may get incorporated into widely used software. Or not.
- What impresses me is the methodology. The assumptions are laid out as in theoretical crypto research. The BBD scheme and its competitors are precisely defined.

What's New Here?

- The measure of relative usability is based on a plausible model of human memory.
- The measures of relative security are based on specific attacks that model those used in the real world.
- It looks to me, from a relatively uninformed perspective, that this is the *start* of rigorous research on password selection.