

# STATISTICAL METRICS FOR **INDIVIDUAL** PASSWORD STRENGTH

**Joseph Bonneau**

jcb82@cl.cam.ac.uk



**UNIVERSITY OF  
CAMBRIDGE**

**Computer Laboratory**

SECURITY PROTOCOLS WORKSHOP  
CAMBRIDGE, UK  
APRIL 12, 2012

# How strong is my password?



Password:

Type the password:

Strength score is:  Strength verdict:



Username:

Password:  **Medium**



PHP Password Strength Meter

**Strong**



Company:

Email:

Password:

Test Your Password		Minimum Requirements
Password:	<input type="password" value="....."/>	<ul style="list-style-type: none"><li>• Minimum 8 characters in length</li><li>• Contains 3/4 of the following items:<ul style="list-style-type: none"><li>- Uppercase Letters</li><li>- Lowercase Letters</li><li>- Numbers</li><li>- Symbols</li></ul></li></ul>
Hide:	<input checked="" type="checkbox"/>	
Score:	<input type="text" value="76%"/>	
Complexity:	Strong	

# Approach #1: Assume a model probability distribution

## GRC's Interactive Brute Force Password "Search Space" Calculator

(NOTHING you do here ever leaves your browser. What happens here, stays here.)

No Uppercase     3 Lowercase     7 Digits     No Symbols   

**abc1234567**

Enter and edit your test passwords in the field above while viewing the analysis below.

### Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26+10 = 36
Search Space Length (Characters):	10 characters
Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to this password's length)	3,760,620,109,779,060
Search Space Size (as a power of 10):	$3.76 \times 10^{15}$

### Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	1.20 thousand centuries
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	10.45 hours
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	37.61 seconds

Note that typical attacks will be online password guessing limited to, at most, a few hundred guesses per second.

# Approach #1: Assume a model probability distribution

Length Char.	94 Character Alphabet			10 char. alphabet		94 char alphabet
	No Checks	Dictionary Rule	Dict. & Comp. Rule			
1	4	-	-	3	3.3	6.6
2	6	-	-	5	6.7	13.2
3	8	-	-	7	10.0	19.8
4	10	14	16	9	13.3	26.3
5	12	17	20	10	16.7	32.9
6	14	20	23	11	20.0	39.5
7	16	22	27	12	23.3	46.1
8	18	24	30	13	26.6	52.7
10	21	26	32	15	33.3	65.9
12	24	28	34	17	40.0	79.0
14	27	30	36	19	46.6	92.2
16	30	32	38	21	53.3	105.4
18	33	34	40	23	59.9	118.5
20	36	36	42	25	66.6	131.7
22	38	38	44	27	73.3	144.7
24	40	40	46	29	79.9	158.0
30	46	46	52	35	99.9	197.2
40	56	56	62	45	133.2	263.4

NIST “entropy” formula

# Approach #1: Assume a model probability distribution

Other models:

- 1 Markov models
- 2 Probabilistic context-free grammar
- 3 Edit distance

# Approach #2: Time to crack



## Secure Purchase Accent Password Recovery Software



Purchase Accent software securely **online** and **offline**. To purchase, choose a product title and a type of license you'd like to pay for, then just follow the instructions.

Pay with credit cards, checks, money orders, PayPal, or bank transfers.

Read [Terms and Conditions](#).



### Prices:

- Standard Edition - **£1,399**  
(you save £396)
- Forensic Edition - **£3,995**  
(you save £1,895)
- Business Edition - **£13,995**  
(you save £5,177)

# Massive password data sets available for the first time

290729	123456
79076	12345
76789	123456789
59462	password
49952	iloveyou
33291	princess
21725	1234567
20901	rockyou
20553	12345678
16648	abc123

RockYou leak

# This talk: assume the distribution is known

- Assume a completely-known distribution  $\mathcal{X}$
- $\mathcal{X}$  has  $N$  events (passwords)  $x_1, x_2, \dots$
- Events have probability  $p_1 \geq p_2 \geq \dots \geq p_N \geq 0$

**Question:** How “strong” is a given event  $x$ ?



## 1 Normalisation for uniform distributions:

$$\forall x \in U_N \quad S_{U_N}(x) = \lg N$$

## 2 Monotonicity:

$$\forall x, x' \in \mathcal{X} \quad p_x \geq p_{x'} \iff S_{\mathcal{X}}(x) \leq S_{\mathcal{X}}(x')$$

## 1 Normalisation for uniform distributions:

$$\forall x \in U_N \quad S_{U_N}(x) = \lg N$$

## 2 Monotonicity:

$$\forall x, x' \in \mathcal{X} \quad p_x \geq p_{x'} \iff S_{\mathcal{X}}(x) \leq S_{\mathcal{X}}(x')$$

$$S_{\chi}^P(x) = -\lg p_x$$

## Issues:

- 1 Doesn't correspond to sequential guessing

$$S^l_{\mathcal{X}}(x) = \lg(2 \cdot i_x - 1)$$

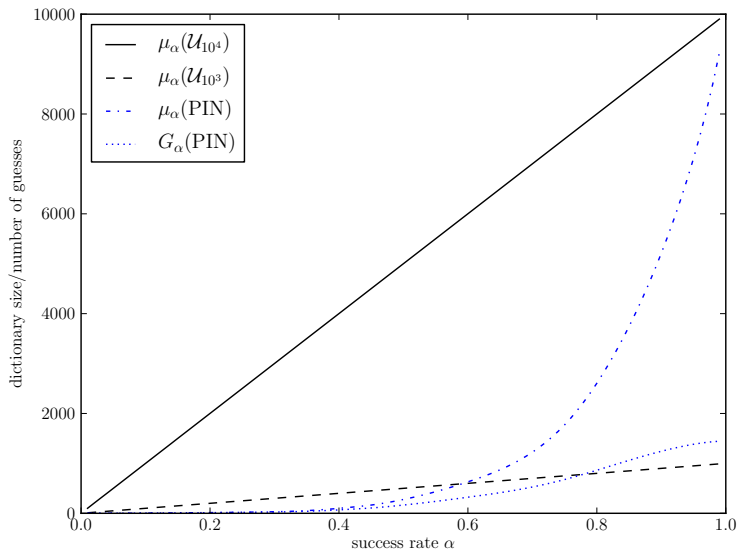
## Issues:

- 1  $S^l_{\mathcal{X}}(x_1) = 0$
- 2 Requires averaging indices for passwords of equal probability
- 3 For  $\mathcal{X} \approx \mathcal{U}_N$ , expected value is  $\approx \lg N - (\lg e - 1)$

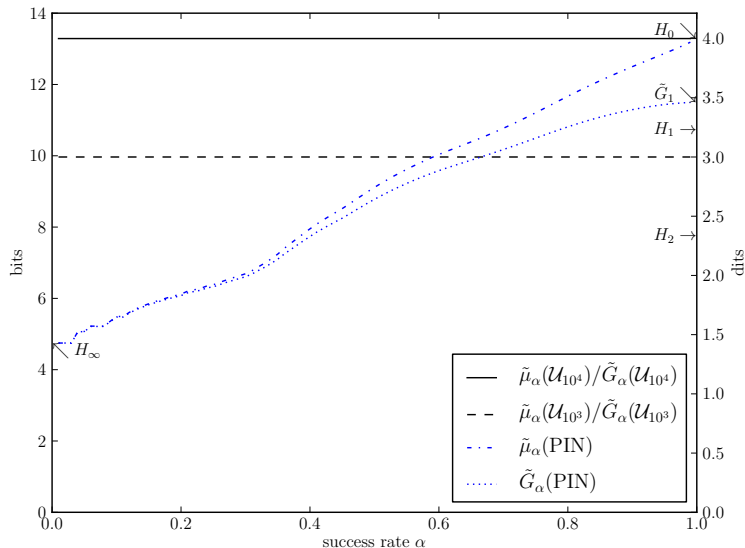
# Adapting distribution-wide metrics

- $\alpha$  is the proportion of accounts broken in a guessing attack
- $\tilde{\mu}_\alpha$  is the optimal dictionary size needed (bits)
- $\tilde{G}_\alpha$  is the actual amount of work per account (bits)

# Adapting distribution-wide metrics



# Adapting distribution-wide metrics



$$S_{\mathcal{X}}^G(x') = \tilde{G}_{\alpha_x}(\mathcal{X})$$

$$\text{where } \alpha_x = \sum_{i=1}^{i_x} p_i$$

## Advantages:

- 1 Normal & monotonic due to definition of  $\tilde{G}_\alpha$
- 2  $S_{\mathcal{X}}^G(x_1) = H_\infty(\mathcal{X})$



# Example estimates for RockYou passwords

$x$	$\lg(i_x)$	$f_x$	$S_{RY}^P$	$S_{RY}^I$	$S_{RY}^G$	$S^{NIST}$
123456	0	290729	6.81	0.00	6.81	14.0
12345	1	79076	8.69	1.58	7.46	12.0
password	2	59462	9.10	2.81	8.01	18.0
rockyou	3	20901	10.61	3.91	8.68	16.0
jessica	4	14103	11.17	4.95	9.42	16.0
butterfly	5	10560	11.59	5.98	10.08	19.5
charlie	6	7735	12.04	6.99	10.71	16.0
diamond	7	5167	12.62	7.99	11.30	16.0
freedom	8	3505	13.18	9.00	11.88	16.0
letmein	9	2134	13.90	10.00	12.48	16.0
bethany	10	1321	14.59	11.00	13.09	16.0
lovers1	11	739	15.43	12.00	13.74	22.0
samanta	12	389	16.35	13.00	14.42	16.0
123456p	13	207	17.27	14.00	15.13	22.0
diving	14	111	18.16	15.00	15.87	14.0
flower23	15	63	18.98	16.00	16.62	24.0
scotty2hotty	16	34	19.87	17.02	17.38	30.0
lilballa	17	18	20.79	18.01	18.13	18.0
robbies	18	9	21.79	19.06	18.93	16.0
DANELLE	19	5	22.64	19.96	19.62	22.0
antanddeck06	20	3	23.37	20.84	20.30	30.0
babies8	21	2	23.96	21.78	21.00	22.0
sapo26	22	1	24.96	24.00	22.44	20.0
jcb82	23	0	24.96	24.00	22.65	18.0

# Example estimates for small distributions

Dataset	$M$	% seen	$S_{RY}^P$	$S_{RY}^I$	$S_{RY}^G$	$S^{NIST}$
RockYou (baseline)	—	100.0%	21.15	18.79	18.75	19.82
small password sets						
Chinese	1000	34.0%	22.28	21.24	21.52	20.21
Fox-Admin	369	68.8%	20.95	18.99	19.33	19.28
Hebrew	1307	50.3%	21.25	19.63	20.14	17.46
Hotmail	11576	57.6%	21.82	20.29	20.43	18.21
myBart	2007	19.0%	22.93	22.37	22.54	23.53
MySpace	50546	59.5%	21.64	20.02	20.19	22.53
NATO-books	11822	50.9%	21.66	20.17	20.47	19.35
Sony-BMG	41024	61.3%	20.93	19.10	19.53	19.87
malware dictionaries						
Conficker	190	96.8%	16.99	13.60	15.07	16.51
Morris	445	94.4%	18.62	15.68	16.56	15.27
blacklists						
Twitter-2010	404	7.9%	23.16	22.86	23.02	15.30
Twitter-2011	429	99.8%	15.11	11.31	13.46	15.27

Thank you

jcb82@cl.cam.ac.uk

Simple solution: add-one smoothing

$$1 \quad S_{\mathcal{X}}^P(x) = \lg(N + 1)$$

$$2 \quad S_{\mathcal{X}}^I(x') = \lg 2N + 1$$

$$3 \quad S_{\mathcal{X}}^G(x') \approx \tilde{G}_1(\mathcal{X})$$

# Stability of metrics

If an event's probability changes from  $p \rightarrow p'$

$$\textcircled{1} \max [\Delta S_{\mathcal{X}}^{\text{P}}(x)] = \text{abs} \left( \lg \frac{p'}{p} \right)$$

$$\textcircled{2} \max [\Delta S_{\mathcal{X}}^{\text{I}}(x)] = \lg \frac{2}{\min(p, p')}$$

$$\textcircled{3} \max [\Delta S_{\mathcal{X}}^{\text{G}}(x)] = \text{abs} \left( \lg \frac{p'}{p} \right)$$

For a Zipf distribution,  $\Delta S_{\mathcal{X}}^{\text{P}}(x) = \Delta S_{\mathcal{X}}^{\text{I}}(x)$