Usability: low tech, high security

Utilisabilité: haute sécurité en basse technologie

Nikola K. Blanchard, Institut de Recherche en Informatique Fondamentale, Université Paris Diderot

PhD defense before the following jury:

Adrian KOSOWSKI	Université Paris Diderot, INRIA	*Examinateur*
Michelle MAZUREK	University of Maryland, College Park	*Rapporteuse*
Marine MINIER	Université de Lorraine	*Examinatrice*
David NACCACHE	Ecole Normale Supérieure de Paris	*Rapporteur*
Peter Y.A. RYAN	Université du Luxembourg	*Rapporteur*
Nicolas SCHABANEL	CNRS, ENS de Lyon	*Co-directeur de thèse*
Ted SELKER	University of Maryland, Baltimore County	*Co-directeur de thèse*

PhD Defense at Université Paris Diderot

June 21st, 2019

Introduction: a voting experiment

Voting experiments in Strasbourg and San-Sebastian

Random-Sample Voting Ballot

QUESTION: Should voting in national elections be compulsory?

VOTING TIME: 12:00PM CET Thursday 17 November 2016 through 9:30PM CET Friday 18 November 2016

INSTRUCTIONS:

1 Choose either half of this sheet randomly (ballot number and password are the same for both halves).

2 Use a web browser to visit the webpage: `https://vbb.rsvoting.org/rsv/vbb/gfmdd2016-q1/`

Your ballot number is your **login ❶**: **`001`**

Your **password ❷** is: **`vhbe-buhb-mrda-fwpz`**

3 When prompted, enter the vote code that is printed adjacent your vote.

4 You should discard or destroy at least the half of this sheet that you used to vote; it is recommended, however, that you keep the other half of this sheet and write down on it in the space provided your vote code for later use in the audit.

Choice	Vote-Code ❸
Yes	4457-1444-2131
No	6975-7435-2625

---✂---✂---

Random-Sample Voting Ballot

QUESTION: Should voting in national elections be compulsory?

VOTING TIME: 12:00PM CET Thursday 17 November 2016 through 9:30PM CET Friday 18 November 2016

INSTRUCTIONS:

1 Choose either half of this sheet randomly (ballot number and password are the same for both halves).

2 Use a web browser to visit the webpage: `https://vbb.rsvoting.org/rsv/vbb/gfmdd2016-q1/`

Your ballot number is your **login ❶**: **`001`**

Your **password ❷** is: **`vhbe-buhb-mrda-fwpz`**

3 When prompted, enter the vote code that is printed adjacent your vote.

4 You should discard or destroy at least the half of this sheet that you used to vote; it is recommended, however, that you keep the other half of this sheet and write down on it in the space provided your vote code for later use in the audit.

Choice	Vote-Code ❸
Yes	4134-9733-6914
No	1855-4750-4118

Random-Sample Voting Ballot

The problem of authentication

Authentication methods

Something you know: passwords

- Low usability with many passwords
- Often badly implemented server-side
- Password managers create a single point of failure

Something you have: devices

- Vulnerable to denial-of-service
- Third-party authentication introduces trust issues

Something you are: biometrics

- Introduces permanent vulnerabilities, security outcome unsure today

Passwords today

State of password use [Wash *et al.*, 2016, Das *et al.*, 2014, Centrify report, 2014]:

- × Average user has $\sim$ 100 accounts
- → ”123456” still the most frequent password [Doel, 2018]
- → High rate of re-use (75% of users)
- → Lots of sharing (40% of users)

- × Creates 50 passwords per year on average
- × No general method, ad-hoc creation due to arbitrary constraints
- → Frequent loss of passwords (40% to 60% reinitialised every 3 months)

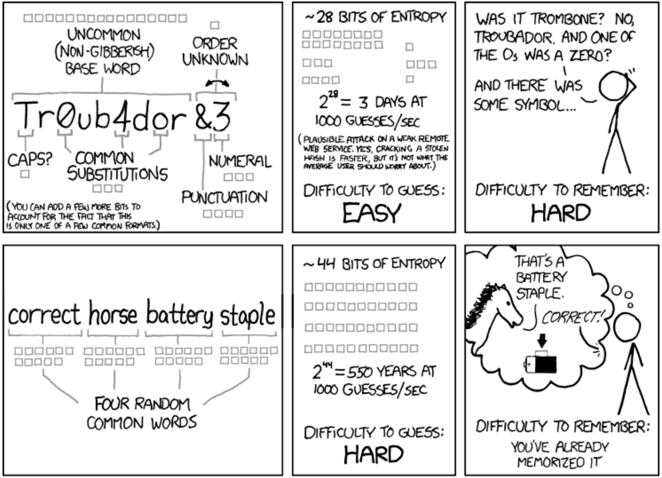


Image from XKCD, also shown in [Shay *et al.*, 2012]

Attacking the password:

- Constraints are counter-productive [Cranor, 2016, Ur *et al.*, 2015, Florêncio *et al.*, 2014]
- Length trumps complexity [Shay *et al.*, 2014]

Attacking the server [Florêncio *et al.*, 2014]:

- Passwords should be salted and hashed (Facebook, march 2019)
- The hash function has to be specifically chosen (SHA-256 is not enough)
- It should all happen client-side

Methodology

Methodology 1: user studies

How to observe real effects on population samples:

- Control the probability of the effect being a fluke
- Have large sample sizes
- Set hypotheses in advance:
  1. Refer to bibliography
  2. Use simulations
  3. Organise a pilot study
- Limit the impact of priming:
  1. Use neutral wording
  2. When priming unavoidable, make it go against the hypothesised effect

Methodology 2: p-values

Is an effect real?

- Set a hypothesis
- Estimate the p-value ≈ probability of observing the data if the hypothesis is false
- Hypothesis is considered statistically significant if $p<0.05$

However:

- $p<0.05$ is not equivalent to 95% probability of being true!
- Testing n hypotheses simultaneously increases the probability of a false positive. This needs to be controlled for:
  1. Bonferroni: divide the threshold for statistical significance by n
  2. Holm: sort p-values and reject all the ones for which $p_k > \frac{0.05}{n+1-k}$

Main results

Our results, part 1: Authentication

→ Analysis of code transcription `hK8iLK!6z vs BOC MIP POD`

Consonant-Vowel-Consonant for Error-Free Code Entry, Blanchard N.K., Gabasova L., Selker T., *in HCI International, 2019*

→ Typo correction in passwords Passwo~~o~~rd

Comment corriger efficacement les typos dans les mots de passe, Blanchard N.K. *in ALGOTEL 2019*

→ Mental password manager 🧠 → password

Créer de tête de nombreux mots de passe inviolables et inoubliables, Blanchard N.K., Gabasova L., Selker T., Sennesh, E. *in ALGOTEL 2018*

→ Passphrase generator `Furry grills minidesk newsdesk deletes internet`

Improving security and usability with guided word choice, Blanchard N.K., Malaingre C., Selker T., *in ACSAC 2018*

Mots de passe : le choix humain plus sécurisé que la génération aléatoire, Blanchard N.K., Malaingre C., Selker T., *in ALGOTEL 2018*

→ Models of mental computing 🧠 + 🧠 = 2 🧠 ?

Our results, part 2: Voting

→ Usability experiments on voting

Vote par sondage uniforme incorruptible, Blanchard N.K, *in ALGOTEL 2017*

Building Trust for Sample Voting, Blanchard N.K., *in TeSS 2018* and *International Journal of Decision Support System Technology 2018*

Improving voting technology is hard: the trust-legitimacy-participation loop and related problems, Blanchard N.K., Selker T., *in STAST 2018*

→ Usable physical implementations of Three-ballot

→ Primitives and protocols for Boardroom voting

Other results and papers

Dynamic clustering

Dynamic Sum-Radii Clustering, Blanchard N.K., Schabanel N., *in WALCOM 2017*

Institution design

CIVICS: Changing Incentives for Voters in International Cooperation through Sampling, Blanchard N.K., *in 2019 Smolny Conference*

Metaheuristics for planetary science

Progressive metaheuristics for high-dimensional radiative transfer model inversion, Gabasova L., Blanchard N.K., Schmitt B., Grundy W., New Horizons COMP team, *in EPSC 2018*

Pluto surface composition from spectral model inversion with metaheuristics, Gabasova L., Blanchard N.K., Olkin, C.B., Spencer, J.R., Young, L.A., Smith, K.E. Weaver, H.A. Stern, A., New Horizons COMP team, *in EPSC 2019*

Plan of the talk

Analysis of code transcription

Consonant-Vowel-Consonant for Error-Free Code Entry, Blanchard N.K., Gabasova L., Selker T., *in HCI International, 2019*

Typo correction in passwords

Comment corriger efficacement les typos dans les mots de passe, Blanchard N.K. *in ALGOTEL 2019*

Mental password manager

Créer de tête de nombreux mots de passe inviolables et inoubliables, Blanchard N.K., Gabasova L., Selker T., Sennesh, E. *in ALGOTEL 2018*

Passphrase generator

Improving security and usability with guided word choice, Blanchard N.K., Malaingre C., Selker T., *in ACSAC 2018*

Mots de passe : le choix humain plus sécurisé que la génération aléatoire, Blanchard N.K., Malaingre C., Selker T., *in ALGOTEL 2018*

Models of mental computing

Plan of the talk

Analysis of code transcription

Consonant-Vowel-Consonant for Error-Free Code Entry, Blanchard N.K., Gabasova L., Selker T., *in HCI International, 2019*

Typo correction in passwords

Comment corriger efficacement les typos dans les mots de passe, Blanchard N.K. *in ALGOTEL 2019*

Mental password manager

Créer de tête de nombreux mots de passe inviolables et inoubliables, Blanchard N.K., Gabasova L., Selker T., Sennesh, E. *in ALGOTEL 2018*

Passphrase generator

Improving security and usability with guided word choice, Blanchard N.K., Malaingre C., Selker T., *in ACSAC 2018*

Mots de passe : le choix humain plus sécurisé que la génération aléatoire, Blanchard N.K., Malaingre C., Selker T., *in ALGOTEL 2018*

Models of mental computing

Password typo correction

Why typo correction

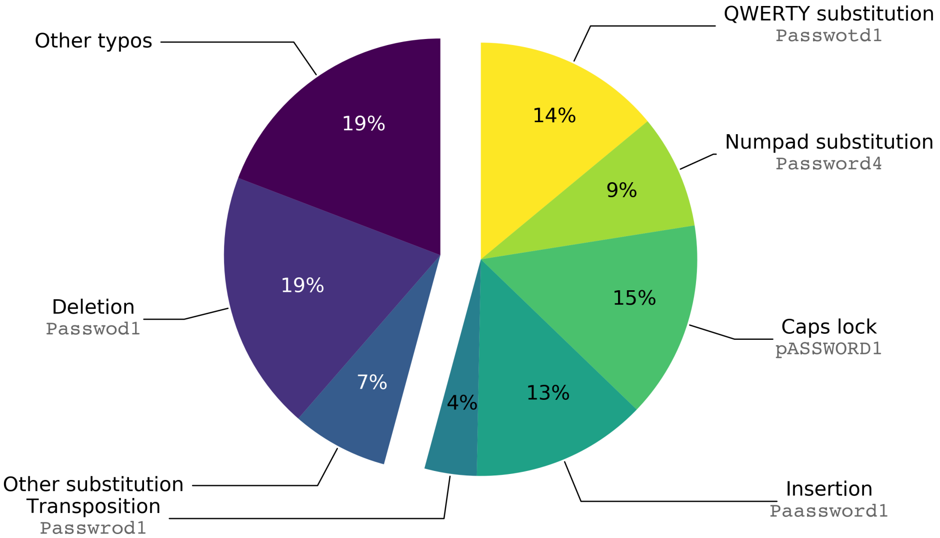
Typos lower usability [Chatterjee *et al.*, 2016,2017, Woodage *et al.*, 2017]:

- Very frustrating
- Frequent (3% error rate)
- More prevalent with longer passwords/passphrases

Correcting typos does not lower security:

- No effect on offline attacks
- Most frequent passwords are far from each other
- Stricter rate limiting than without typo correction

Types of typos (recomputed from [Chatterjee *et al.*, 2016])



Constraints

Secure: no new vulnerabilities beyond the accepted typos

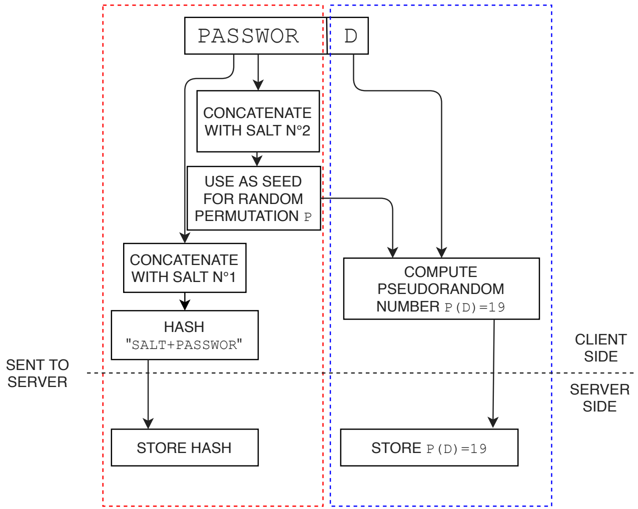
Low cost:

- No expensive computation on the server
- Simple to implement/backwards compatible
- Compatible with hashing

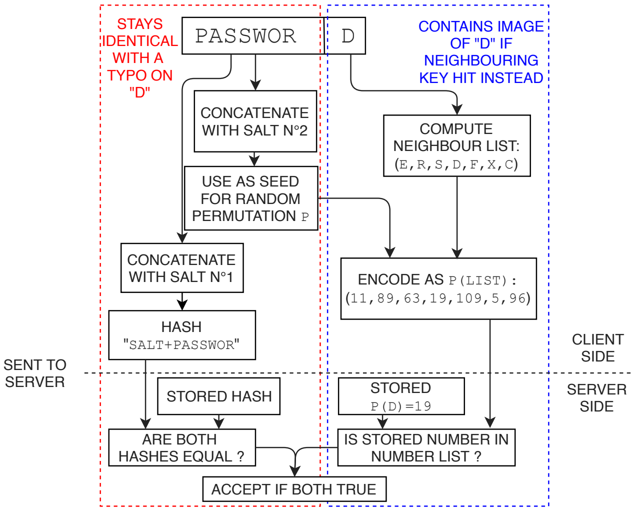
Correct as many *acceptable* typos as possible (32% in [Chatterjee *et al.*, 2016])

Correcting substitutions

Correcting substitutions: Registration



Correcting substitutions: Login



Extending to other typos

Transposition:

- Remove two letters before hashing
- Encode each letter with two different random permutations

Insertion:

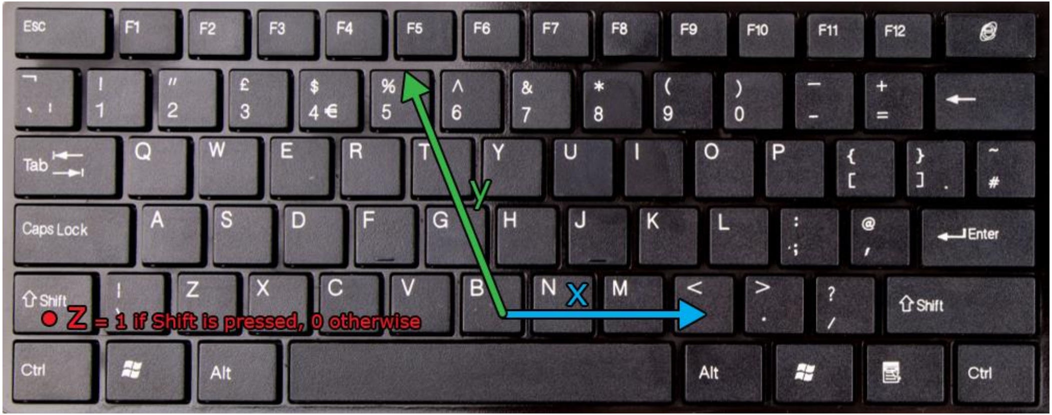
- Combine both previous methods
- Removing two letters from an insertion can be found using the substitution hash

Comparison of the frameworks

Algorithm	Substitution	Transposition	Insertion	Complete
Computation in # of				
Permutations	n	$4n-4$	$4n-4$	$\max(4(n-1), 60)$
Hashes	$n+1$	n	n	$\max(n+1, 17)$
Numbers	$n \times k$	$(n-1) \times 4k$	$(n-1) \times 4k$	$\max(4(n-1)k, 60k)$
Storage in # of				
Hashes	$n+1$	n	$2n$	$\max(2n+1, 33)$
Numbers	n	$4n$	$5n$	$\max(5n, 80)$
Typos handled				
Conservative	24.2 %	28.4 %	34.5 %	50.2 %
Tolerant	24.2 %	28.4 %	42.2 %	57.7 %

A simpler theoretical algorithm

Create a coordinate system on the keyboard such that legitimate typos are at distance 1.



Generic algorithm based on the discrete logarithm

For small primes p_i, encode password as

$$X(P) = \prod_{1 \leq i \leq n} p_i^{x_i} \times p_{i+n}^{y_i} \times p_{i+2n}^{z_i}$$

Send $g^{X(P)}$ for a random g in a given large group.

If $P' \approx P$: $\quad g^{X(P')} = (g^{X(P)})^{p_i} \quad$ OR $\quad (g^{X(P')})^{p_i} = g^{X(P)}$

Summary of typo correction

Secure:

- Similar online resistance as [Chatterjee *et al.*, 2017]
- Offline attack speed-up < 1.5 on real-world data.

Low cost:

- No extra computation on the server in expectation
- All communications still fit in a single normal-size packet
- Compatible with previous systems

Corrects 57% of all typos, 91% of *acceptable* typos.

Plan of the talk

Analysis of code transcription

Consonant-Vowel-Consonant for Error-Free Code Entry, Blanchard N.K., Gabasova L., Selker T., *in HCI International, 2019*

Typo correction in passwords

Comment corriger efficacement les typos dans les mots de passe, Blanchard N.K. *in ALGOTEL 2019*

Mental password manager

Créer de tête de nombreux mots de passe inviolables et inoubliables, Blanchard N.K., Gabasova L., Selker T., Sennesh, E. *in ALGOTEL 2018*

Passphrase generator

Improving security and usability with guided word choice, Blanchard N.K., Malaingre C., Selker T., *in ACSAC 2018*

Mots de passe : le choix humain plus sécurisé que la génération aléatoire, Blanchard N.K., Malaingre C., Selker T., *in ALGOTEL 2018*

Models of mental computing

Cue-Pin-Select: a mental password manager

joint work with Leila Gabasova, Ted Selker and Eli Sennesh

Constraints for a good password management algorithm

Security:

- High entropy for each password
- High residual entropy against stolen clear-text passwords

Usability:

- Memorable even without frequent use (hence deterministic)
- Easy to understand by laypeople

Adaptability:

- Compatible with frequent constraints

Idea: mentally extract entropy from a large secret

Cue-Pin-Select

High-level view:

- Create one high-entropy passphrase and a 4-digit *PIN*

```
parallel major domain disastrous divergent waterways
```

6908

- Create a 4-letter *cue* for each service

AMZN

→ Deterministically extract 4 *trigrams* from the passphrase using the *PIN* and the *cue*

```
parallel major domain disastrous divergent waterways
```

Example run

1

amzn

parallel major domain disastrous divergent waterways

+6

6908 maj_____________

2

Security analysis

Bruteforcing Cue-Pin-Select

Today's standard for web services: 36-42 bits (30 years at 1000 tries/s).

Brute-force against Cue-Pin-Select:

- Naive against a password $\rightarrow$ 56 bits
- Optimised dictionary against a password $\rightarrow$ 52 bits
- Naive against passphrase $\rightarrow$ 210 bits
- Dictionary against passphrase $\rightarrow$ 111 bits

Statistical studies of clear-text attacks

To simplify analysis, we assume a very strong adversary who knows:

- 1+ revealed passwords
- Length of the passphrase
- Position of each revealed trigram in the passphrase

We uniformly randomly generate 10 000 passphrases, cues and corresponding passwords and test the entropy left

Simulated cleartext attack

Passphrase:

PARALLELMAJORDOMAINDISASTROUSDIVERGENTWATERWAYS

Adversary knows just the length:

____

One clear-text:

________MAJ______________ROUSD__________TER____

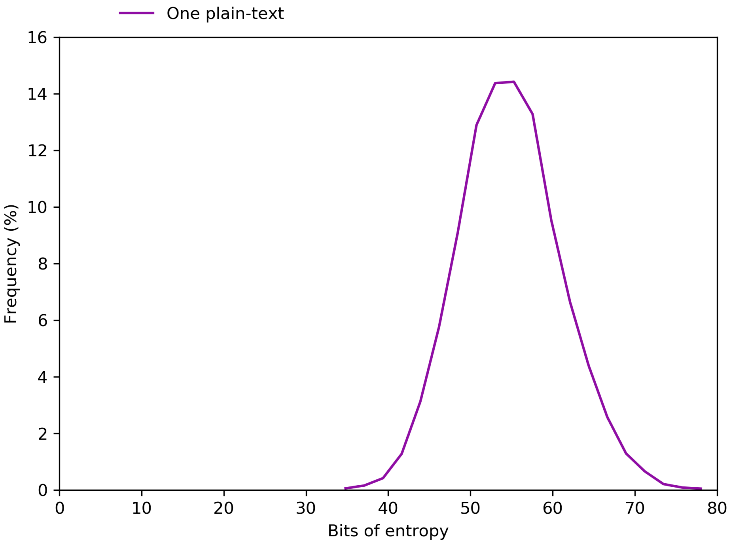
Two clear-texts:

__RAL__LMAJ______IND_____ROUSD______NTW_TER____

Three clear-texts:

P_RAL__LMAJ______IND_____ROUSDIV___ENTW_TER__YS

Residual entropy for 1-3 clear-texts (10 000 random passphrase/cue couples)



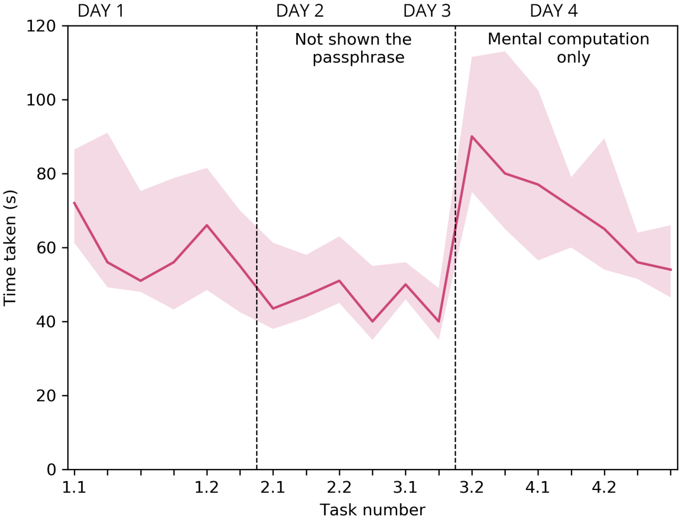
Testing it on users

User experiment

4-day experiment:

- Day 1: high cost, some errors
- Day 2: quick speed-up with pen and paper
- Day 3: increase when shift to mental computation
- Day 4: speed-up over the last day, no errors

At the end, large variability, 24-71s



Adaptability

Algorithm can be extended to handle:

- Number and special characters
- Length constraints
- Frequent changes

Cue-Pin-Select Summary

Cue-Pin-Select:

- 52 bits security per password
- Guaranteed resistance to single clear-text attack, probable resistance to 2-3 clear-texts
- Can create 500+ passwords without high risk of strong partial collision
- Quick learning process to get under 1 min
- According to models, strongly memorable
- Natural extension to handle frequent constraints
- Other extension to improve security

Plan of the talk

Analysis of code transcription

Consonant-Vowel-Consonant for Error-Free Code Entry, Blanchard N.K., Gabasova L., Selker T., *in HCI International, 2019*

Typo correction in passwords

Comment corriger efficacement les typos dans les mots de passe, Blanchard N.K. *in ALGOTEL 2019*

Mental password manager

Créer de tête de nombreux mots de passe inviolables et inoubliables, Blanchard N.K., Gabasova L., Selker T., Sennesh, E. *in ALGOTEL 2018*

Passphrase generator

Improving security and usability with guided word choice, Blanchard N.K., Malaingre C., Selker T., *in ACSAC 2018*

Mots de passe : le choix humain plus sécurisé que la génération aléatoire, Blanchard N.K., Malaingre C., Selker T., *in ALGOTEL 2018*

Models of mental computing

Empirically testing mental computing models

joint work with Ted Selker and Florentin Waligorski

Why mental computing models

It has immediate effects:

- It allows systematic comparison of mental algorithms
- Replaces some user experiments
- Large savings in time/money

It is a fundamental question:

- Old question in cognitive science [Dehaene, 1992], [Ashcraft, 1992], [Butterworth *et al.*, 2001], [Rodic *et al.*, 2015]
- Brought to CPSci by [Blocki, Blum *et al.*, 2013, 2015, 2017]
- It can guide the development of new methods (e.g. in education)

Model by Blocki and Blum

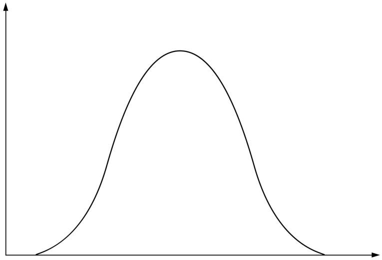
Summary of the original model:

Operation	Input digits	Proposed cost
Equality	1	1
	2	2
Addition + modulo	1	# output digits
	2	1 + # output digits
Multiplication + modulo	1	# output digits
	2	1 + # output digits
Character-to-digit map	N/A	1

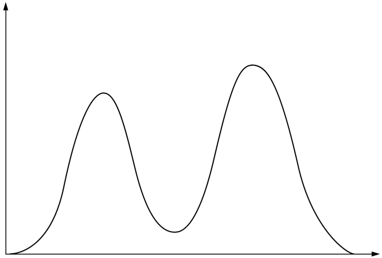
What we want

Three objectives:

- Distribution instead of single cost



- Cluster analysis of users



- Empirical validation

User study

81 different users, speaking mainly English and French

9 sections in the experiment to answer the following:

- Get baseline costs for operations
- Access time to the i-th element
- Do costs commute?
- Are abilities are clustered?

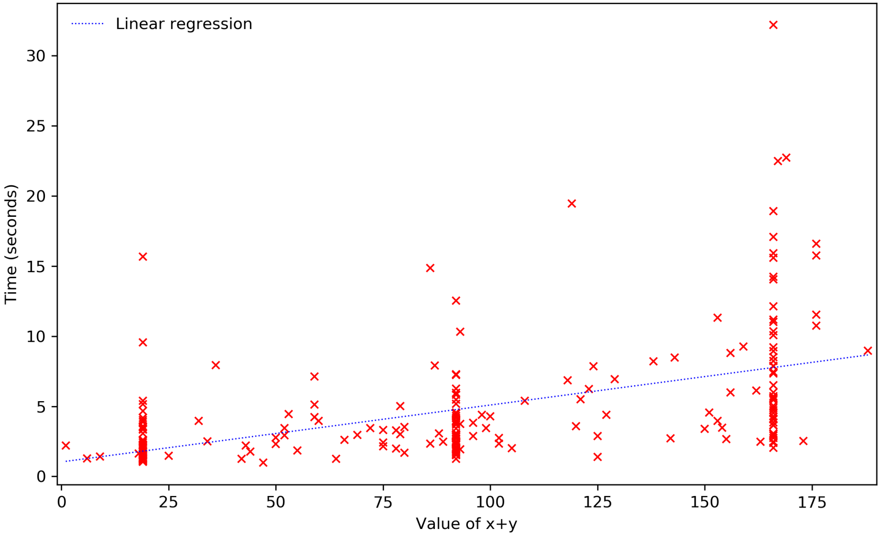
Preliminary results

Access time in a letter/number map is not constant:

- Times between 1.6s and 13.9s
- Getting the next element is 2-3 times faster than the previous
- Only partial re-use of previously computed maps
- Validated with month/number map

Arithmetic operations are not linear (in # of digits). They seem linear in output value (consistent with [Dehaene, 1992] but more work is needed.

Arithmetic operations: times



Conclusion

Summary of research questions

How to improve password usability:

- Use better codes
- Generate more memorable secrets
- Correct typos to allow longer passwords
- Find methods to create many passwords

Using similar ideas in voting:

- Investigate what people can do and start from that
- Propose paper-based solutions to improve trust and understanding
- Work on the pipeline from research to real implementation

Future research

Many questions on the mental computing models:

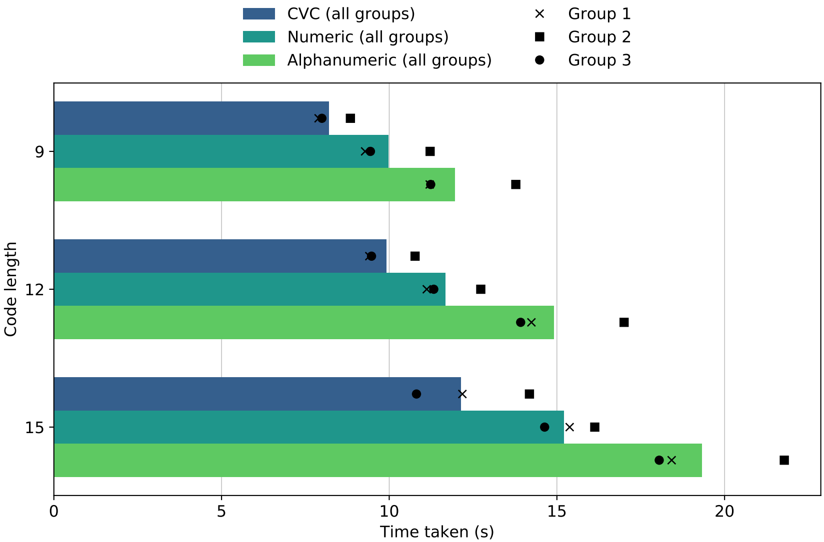
- Are abilities clustered? Do we need tailored mental algorithms?
- How do costs interact inside a mental algorithm?
- Can we develop a realistic cost function?
- Can we prove lower bounds for Cue-Pin-Select or find better mental algorithms?

Second direction, usable voting:

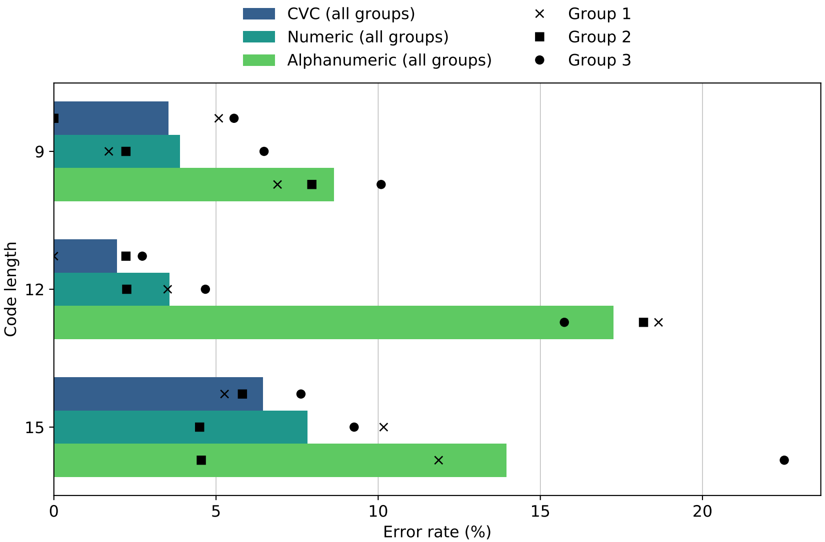
- How usable and secure are the paper voting protocols proposed in practice?
- Can we make a relevant model to prove security bounds?

Thank you for your attention

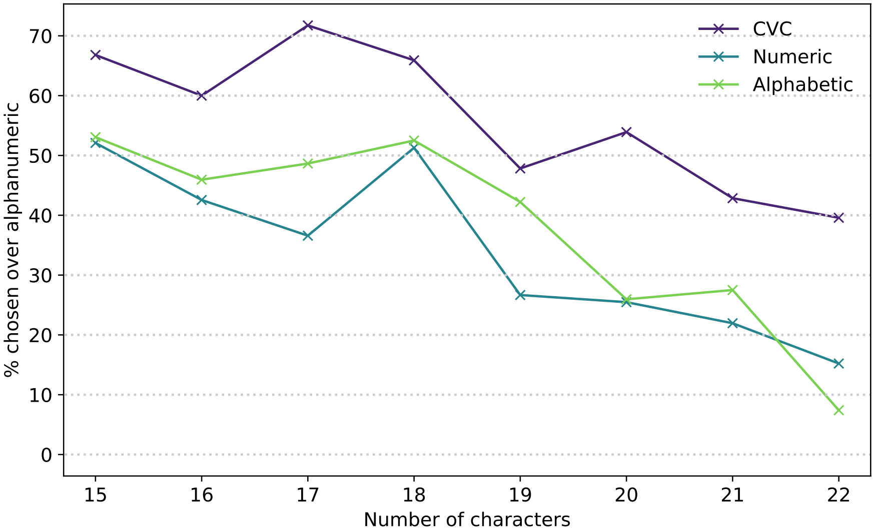
CVC: speed by structure and length



CVC: error rates by structure and length



CVC: code preference against alphanumeric

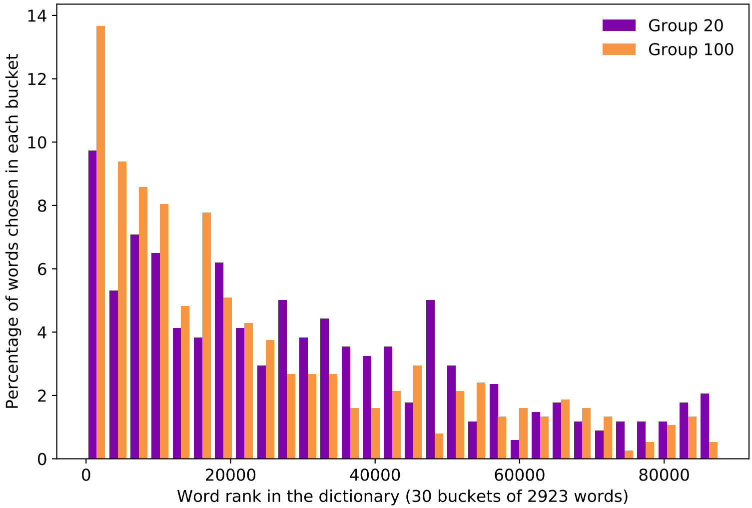


Typo: Key-setting transposition-tolerant algorithm

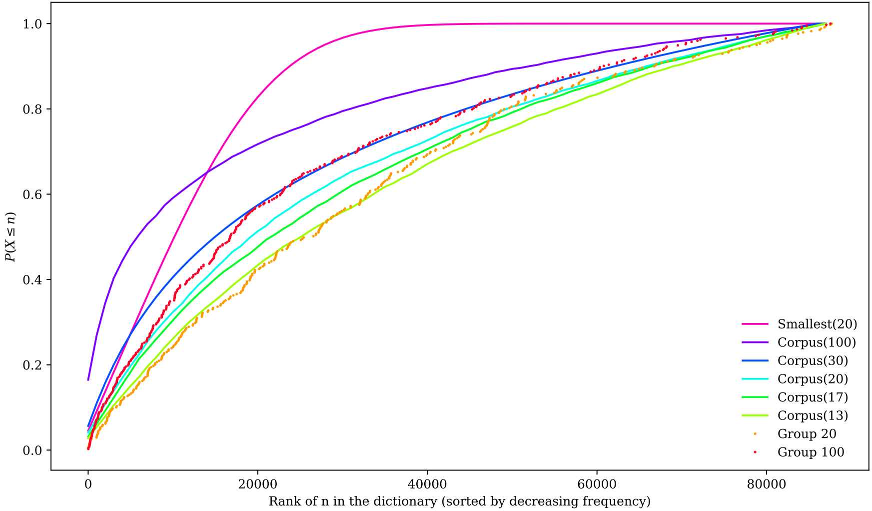
Data: Salts $S_0, S_1, \ldots S_5$, Password P of length n, Keyboard map M: Keys $\rightarrow [0; 255]$
Result: Main hash and list of $n-1$ (hash / integer list) pairs

begin
  $H_0 \longleftarrow \text{HASH}(\mathrm{Concatenate}(S_0, P))$
  for i from 1 to $n-1$ do
    $P_i \longleftarrow P \setminus \{P[i] \bigcup P[i+1]\}$
    $H_i \longleftarrow \text{HASH}(\mathrm{Concatenate}((S_1, P_i))$
    for j from 1 to 4 do
      $\text{Random\_bits[j]} \longleftarrow \text{PRNG}(\mathrm{Concatenate}(S_2, P_i))$
      $\pi_{i,j} \longleftarrow \text{Brassard(Random\_bits[j])}$
    $KA_i \longleftarrow [\pi_{i,1}(M(P[i]))]$
    $KB_i \longleftarrow [\pi_{i,2}(M(P[i+1]))]$
    $KC_i \longleftarrow [\pi_{i,3}(M(P[i]))]$
    $KD_i \longleftarrow [\pi_{i,4}(M(P[i+1]))]$
  return $(H_0, (H_i, KA_i, KB_i, KC_i, KD_i)_{1 \leq i \leq n-1})$

Passphrases: semantic effects



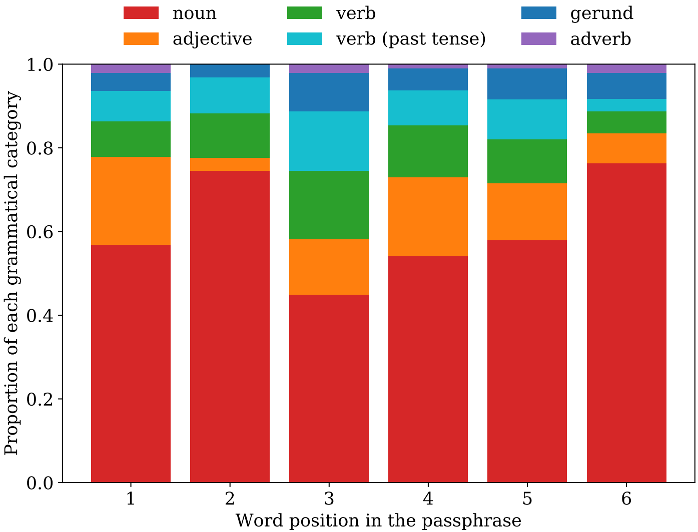
Passphrases: distribution of words chosen



Passphrases: entropy comparison

Strategy	Entropy (bits)	Strategy	Entropy
Uniform(87,691)	16.42	*Smallest*(20)	12.55
Corpus(13)	16.25	*Uniform*(5,000)	12.29
Corpus(17)	16.15	*Uniform*(2,000)	10.97
Corpus(20)	16.10	*Smallest*(100)	10.69
Corpus(30)	15.92	*Corpus*(300,000)	8.94
Corpus(100)	15.32	*Corpus*(87,691)	8.20
Uniform(10,000)	13.29		

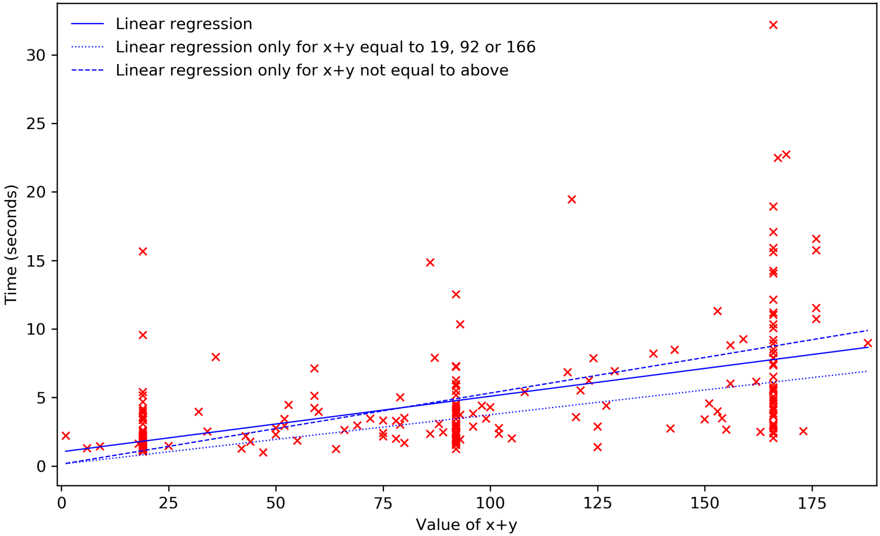
Passphrases: syntactic effects



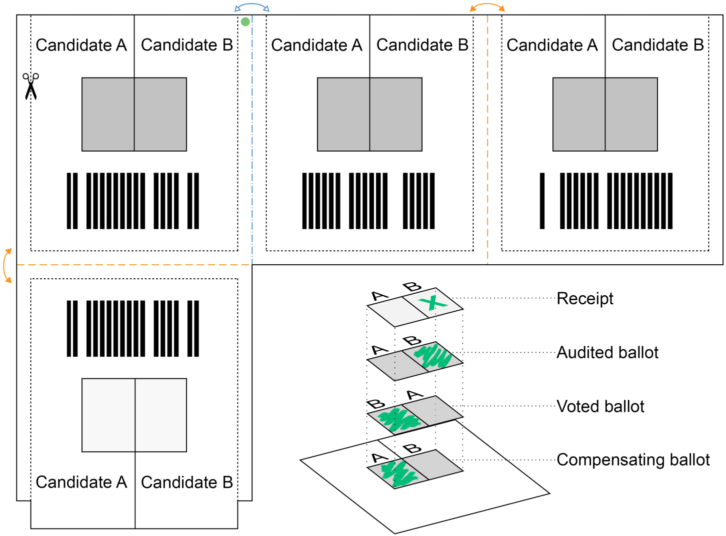
Passphrases: error rates

Section	Correct	Typo	Variant	Order	Miss	Wrong
Control	23% (6/26)	0.42 (11)	0.42 (11)	0.38 (10)	1.19 (31)	0.46 (12)
1:20	40% (19/47)	0.13 (6)	0.17 (8)	0.13 (6)	0.55 (26)	0.11 (5)
1:100	51% (26/51)	0.20 (10)	0.10 (5)	0.06 (3)	0.31 (16)	0.08 (4)
2:20	48% (14/29)	0.03 (1)	0.07 (2)	0.28 (8)	0	0.10 (3)
2:100	58% (15/26)	0.15 (4)	0.08 (2)	0.11 (3)	0.04 (1)	0.15 (4)

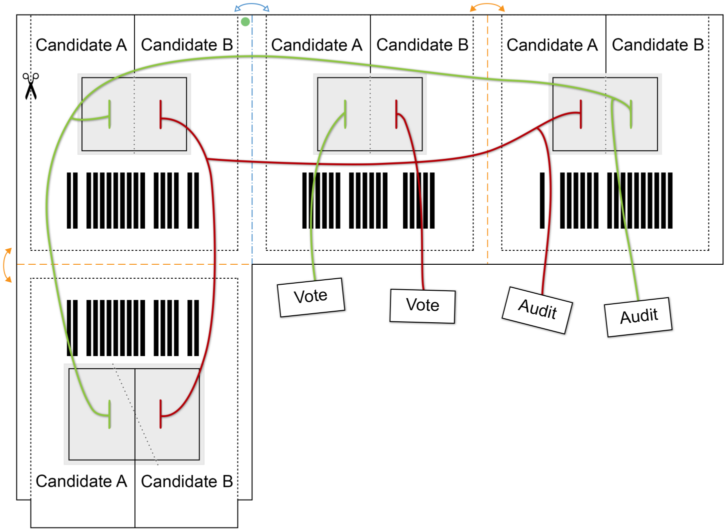
Mental arithmetic operations: different regressions



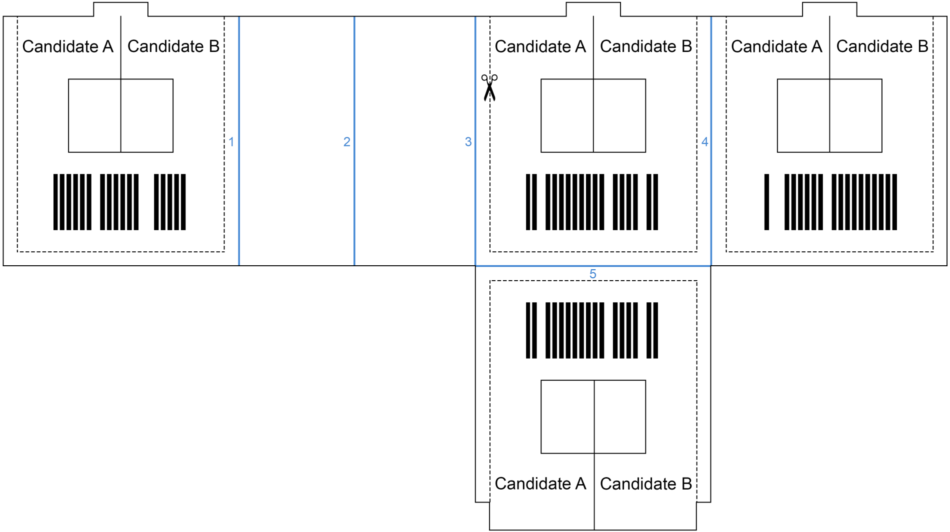
Ballot designs



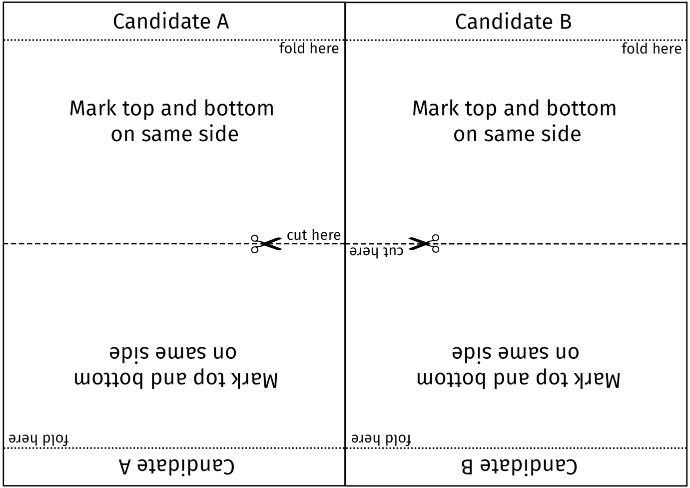
Ballot designs



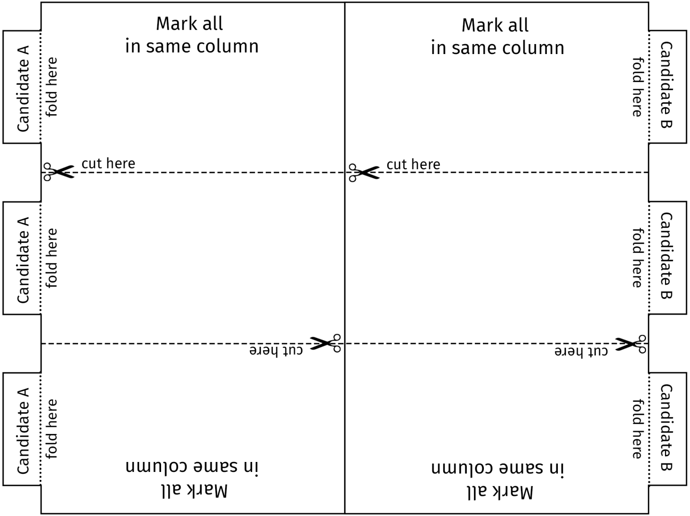
Ballot designs



Boardroom ballot designs



Boardroom ballot designs



Boardroom ballot designs