

OF CONTRASEÑAS, סיסמאות AND 密码: CHARACTER ENCODING ISSUES FOR WEB PASSWORDS

Joseph Bonneau

Rubin Xu

jcb82@cl.cam.ac.uk



UNIVERSITY OF
CAMBRIDGE

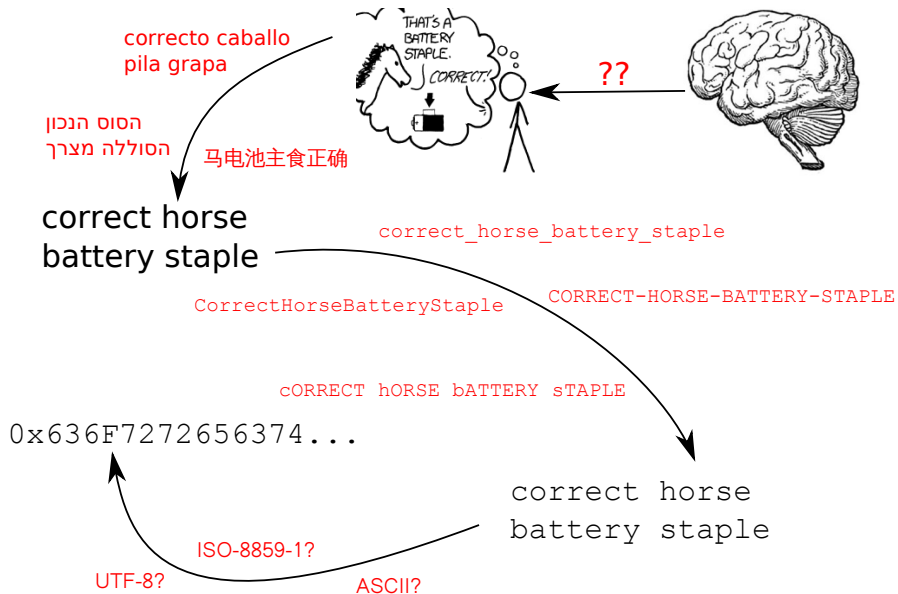
Computer Laboratory

WEB 2.0 SECURITY & PRIVACY

SAN FRANCISCO, CA, USA

MAY 24, 2012

How passwords get created



Writing systems around the world



Surprisingly little variation in (weak) passwords!

		dictionary										global
		de	en	es	fr	id	it	ko	pt	zh	vi	
target	de	6.5%	3.3%	2.6%	2.9%	2.2%	2.8%	1.6%	2.1%	2.0%	1.6%	3.5%
	en	4.6%	8.0%	4.2%	4.3%	4.5%	4.3%	3.4%	3.5%	4.4%	3.5%	7.9%
	es	5.0%	5.6%	12.1%	4.6%	4.1%	6.1%	3.1%	6.3%	3.6%	2.9%	6.9%
	fr	4.0%	4.2%	3.4%	10.0%	2.9%	3.2%	2.2%	3.1%	2.7%	2.1%	5.0%
	id	6.3%	8.7%	6.2%	6.3%	14.9%	6.2%	5.8%	6.0%	6.7%	5.9%	9.3%
	it	6.0%	6.3%	6.8%	5.3%	4.6%	14.6%	3.3%	5.7%	4.0%	3.2%	7.2%
	ko	2.0%	2.6%	1.9%	1.8%	2.3%	2.0%	5.8%	2.4%	3.7%	2.2%	2.8%
	pt	3.9%	4.3%	5.8%	3.8%	3.9%	4.4%	3.5%	11.1%	3.9%	2.9%	5.1%
	zh	1.9%	2.4%	1.7%	1.7%	2.0%	2.0%	2.9%	1.8%	4.4%	2.0%	2.9%
	vi	5.7%	7.7%	5.5%	5.8%	6.3%	5.7%	6.0%	5.8%	7.0%	14.3%	7.8%

for top 1000 passwords, greatest efficiency loss is only 4.8 (fr/vi)

Research questions

- why is there so little language variation?
- how do non-English speakers choose passwords?
- how do websites fail for non-English chraracters?
- how do users cope with an English-dominated world?

Character encoding: a mercifully brief history

- ASCII (ca 1960)
 - English subset of Latin alphabet only
 - ≈ 128 code points defined
 - high-order bit preserved for parity checking
- ASCII extensions
 - use high-order bits for extra characters
 - proprietary schemes (Windows code sheets)
 - 1988: ISO 8859 series (16 subsets)
- multi-byte encoding schemes
 - defined for Chinese, Japanese, Korean, and others
 - most use 2 bytes per character
- the dawn of the Internet
 - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
 - DNS: ASCII subset

Character encoding: a mercifully brief history

- ASCII (ca 1960)
 - English subset of Latin alphabet only
 - ≈ 128 code points defined
 - high-order bit preserved for parity checking
- ASCII extensions
 - use high-order bits for extra characters
 - proprietary schemes (Windows code sheets)
 - 1988: ISO 8859 series (16 subsets)
- multi-byte encoding schemes
 - defined for Chinese, Japanese, Korean, and others
 - most use 2 bytes per character
- the dawn of the Internet
 - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
 - DNS: ASCII subset

Character encoding: a mercifully brief history

- ASCII (ca 1960)
 - English subset of Latin alphabet only
 - ≈ 128 code points defined
 - high-order bit preserved for parity checking
- ASCII extensions
 - use high-order bits for extra characters
 - proprietary schemes (Windows code sheets)
 - 1988: ISO 8859 series (16 subsets)
- multi-byte encoding schemes
 - defined for Chinese, Japanese, Korean, and others
 - most use 2 bytes per character
- the dawn of the Internet
 - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
 - DNS: ASCII subset

Character encoding: a mercifully brief history

- ASCII (ca 1960)
 - English subset of Latin alphabet only
 - ≈ 128 code points defined
 - high-order bit preserved for parity checking
- ASCII extensions
 - use high-order bits for extra characters
 - proprietary schemes (Windows code sheets)
 - 1988: ISO 8859 series (16 subsets)
- multi-byte encoding schemes
 - defined for Chinese, Japanese, Korean, and others
 - most use 2 bytes per character
- the dawn of the Internet
 - HTML, HTTP: ISO-8859-1 (Western Latin/Latin-1)
 - DNS: ASCII subset

Unicode and UTF-8

- Unicode

- assigns a *code point* to every character in human writing systems
- e.g. ñ → 241
- **many** other features
- over 1 M code points defined

- UTF-8

- assigns code point to a variable number of bytes
- e.g. 241 (ñ) → 0xc3b1
- never allows 0x00 to appear outside code point 0

Unicode and UTF-8

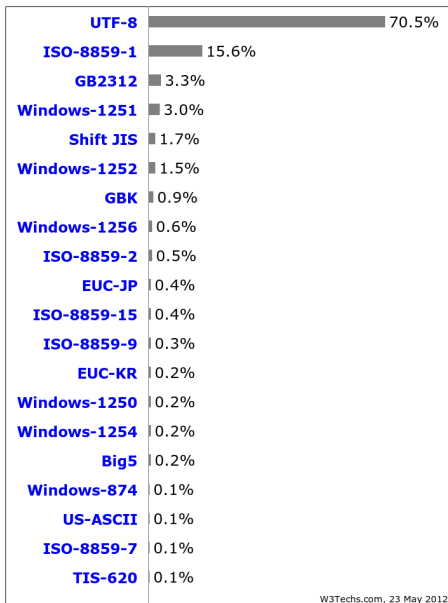
- Unicode

- assigns a *code point* to every character in human writing systems
- e.g. ñ → 241
- **many** other features
- over 1 M code points defined

- UTF-8

- assigns code point to a variable number of bytes
- e.g. 241 (ñ) → 0xc3b1
- never allows 0x00 to appear outside code point 0

Frequency of character encoding schemes today



The password submission process-step 1



user types password

- managed by OS/browser
- code point and encoding known

The password submission process-step 1



mima



1.密码 2.米玛 3.米 4.迷 5.密



user types password

- managed by OS/browser
- code point and encoding known

The password submission process-step 2

Password

Retype Password

browser transcodes password to page encoding

- many places for page to specify
 - HTTP header, HTML header, form attribute
- replace with HTML numeric character reference
- undefined behavior if character entity reference also available!
 - IE: ñ → ñ
 - FF/Chrome: ñ → ñ

The password submission process-step 3

- all characters outside of limited ASCII range are URL-encoded
 - also called percent encoding
- double encoding possible if characters already transcoded
- direct encoding possible for `multipart/formdata` form action

The password submission process-step 3

- all characters outside of limited ASCII range are URL-encoded
 - also called percent encoding
- double encoding possible if characters already transcoded
- direct encoding possible for `multipart/formdata` form action

encoding of 爱 (love)

encoding	submission	length
GB2312	%B0%AE	6
UTF-8	%E7%88%B1	9
ISO 8859-1	%26%2329233%3B	14

What sites need to do to support UTF-8 passwords

What sites need to do to support UTF-8 passwords

NOTHING

Part 1: what can go wrong

Test of 22 sites:

- **English**/UTF-8: Google, Facebook, Microsoft Live, Twitter, Wikipedia, Yahoo!
- **English**/ISO-8859-1: Amazon, DeviantArt, Gawker, IMDB, Walmart
- **Chinese**/UTF-8: CSDN, Renren, Kaixin001, Sina Weibo, Tianya, Mop, Gamer.com.tw
- **Chinese**/GB2312: QQ, Taobao, Baidu, Youku

Correctly supporting sites

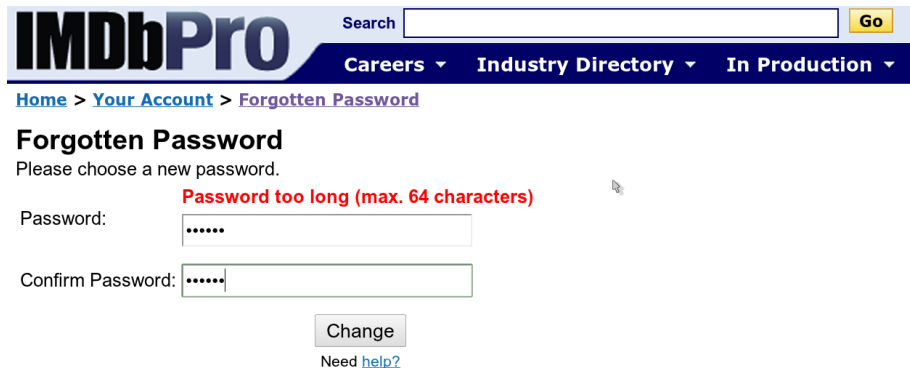
Facebook, Twitter, Wikipedia, DeviantArt¹, CSDN, Renren, Kaixin001

¹Only non-UTF-8 site

Explicit ban on non-ASCII passwords

UTF-8: Google, Microsoft Live, Yahoo!, Sina Weibo, Tianya
other: Amazon, Taobao, Baidu, Youku

Counting encoded bytes instead of logical characters



IMDbPro Search [Go](#)

[Careers](#) ▾ [Industry Directory](#) ▾ [In Production](#) ▾

[Home](#) > [Your Account](#) > [Forgotten Password](#)

Forgotten Password

Please choose a new password.

Password: Password too long (max. 64 characters)

Confirm Password:

[Change](#)

Need [help?](#)

IMDB, Walmart

Code point truncation

Weibo, QQ call `charcodeat()` in JavaScript

Code point truncation

Weibo, QQ call `charcodeat()` in JavaScript

aaaaaaaa
=
LLLLLLLLLLLL
=
CCCCCCCC
=
~~~~~  
=  
屁屁屁屁屁屁屁屁

# DES-crypt() truncation

- Truncation to 8 characters per specification
- [Gamer.com.tw](#): 我的中 accepted for 我的中文得很好
- underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À → 192 → 0xC380
- present in BSD, PHP, PostgreSQL...

# DES-crypt() truncation

- Truncation to 8 characters per specification
- **Gamer.com.tw**: 我的中 accepted for 我的中文得很好
- underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À → 192 → 0xC380
- present in BSD, PHP, PostgreSQL...

# DES-crypt() truncation

- Truncation to 8 characters per specification
- **Gamer.com.tw**: 我的中 accepted for 我的中文得很好
- underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À → 192 → 0xC380
- present in BSD, PHP, PostgreSQL...

# DES-crypt() truncation

- Truncation to 8 characters per specification
- **Gamer.com.tw**: 我的中 accepted for 我的中文得很好
- underlying bug discovered: ÀCEMOMENT accepted for ÀLAPLAGE
  - À → 192 → 0xC380
- present in BSD, PHP, PostgreSQL...

# Down-conversion in `jcrypt()`

- buggy version of Java implementation of `bcrypt()`
- Gawker, Mop: ?????????? accepted for 我的中文得很好

# Down-conversion in `bcrypt()`

- buggy version of Java implementation of `bcrypt()`
- **Gawker**, **Mop**: ?????????? accepted for 我的中文得很好

# Down-conversion in `jcrypt()`

- majority of sites don't support UTF-8 passwords correctly
- many bugs left to find...



## Part 2: how users cope

# Case study: Chinese



mima

1.密码 2.米玛 3.米 4.迷 5.密

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

# Case study: Chinese



mima

1. 密码 2. 米玛 3. 米 4. 迷 5. 密

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

# Case study: Chinese



mima

1. 密码 2. 米玛 3. 米 4. 迷 5. 密

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

# Case study: Chinese



mima

1. 密码 2. 米玛 3. 米 4. 迷 5. 密

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

# Case study: Chinese



mima

1. 密码 2. 米玛 3. 米 4. 迷 5. 密

- Large leaked data sets now available
  - 70yx-gaming site, 10 M users
  - CSDN-forum site, 6 M users
- (nearly) all data in ASCII
  - graphical Pinyin input disabled for password field
- <15% of users enter valid Pinyin passwords
- 45% numeric only, 90% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 11% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

## סיסמאות

- Small leaked data set used
  - Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

## סיסמאות

- Small leaked data set used
  - Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords



## סימאות

- Small leaked data set used
  - Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

## סיסמאות

- Small leaked data set used
  - Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

## סיסמאות

- Small leaked data set used
  - Wondertree-spiritual site, 1K users
- 2.5% of passwords included Hebrew characters
  - over 90% of usernames did...
- 40% numeric only, 65% contain some digits
  - compare to 15%, 45% for RockYou passwords
- 8% adjacent keyboard patterns
  - compare to 3% for RockYou passwords

# Hebrew transliteration strategies



- Phonetic transliteration
  - אהבה → ahava (love)
- Keyboard transliteration
  - אין עוד מלבדו → thigusnksu (There is no one else but him)

# Case study: Spanish



- Spanish alphabet: mostly English/Latin
  - ñ considered a letter proper
  - á, é, í, ó, ú used to indicate stress
- Tens or hundreds of thousands of Spanish passwords at RockYou
  - impossible to compute due to cognates

# Spanish transliteration strategies

| password   | meaning         | proper | transliterated | ratio |
|------------|-----------------|--------|----------------|-------|
| ñ → n      |                 |        |                |       |
| contraseña | password        | 408    | 218            | 34.8% |
| muñeca     | doll            | 197    | 354            | 64.2% |
| cariño     | affection, dear | 104    | 153            | 59.5% |
| pequeña    | little (girl)   | 87     | 72             | 45.2% |
| teextraño  | I miss you      | 65     | 27             | 29.3% |
| á → a      |                 |        |                |       |
| teamomamá  | I love you mom  | 2      | 151            | 98.7% |
| ó → o      |                 |        |                |       |
| código     | code            | 5      | 110            | 95.7% |
| ú → u      |                 |        |                |       |
| música     | music           | 2      | 1447           | 99.9% |

# Spanish transliteration strategies



- ñ transliterated about half of the time
  - varies by password-strongly significant!
- stress accents almost always dropped
  - likely greater than 99% including examples like pájaro (bird)

# Summary



- multilingual passwords are poorly supported
- users rarely make use when they are
- evidence that security is being harmed



# Future directions



- can users enter Chinese passwords securely?
- how will we cope with mobile devices?
- more data needed to study linguistic trends
  - Russian, Arabic, Japanese, Korean, Greek, Hindi, Bengali, etc.

# Thank you

jcb82@cl.cam.ac.uk

major thanks to:

- Noam Szpiro
- Elsa Monica Treviño Ramírez
- Claudia Diaz (Claudia Maria Díaz Martínez)