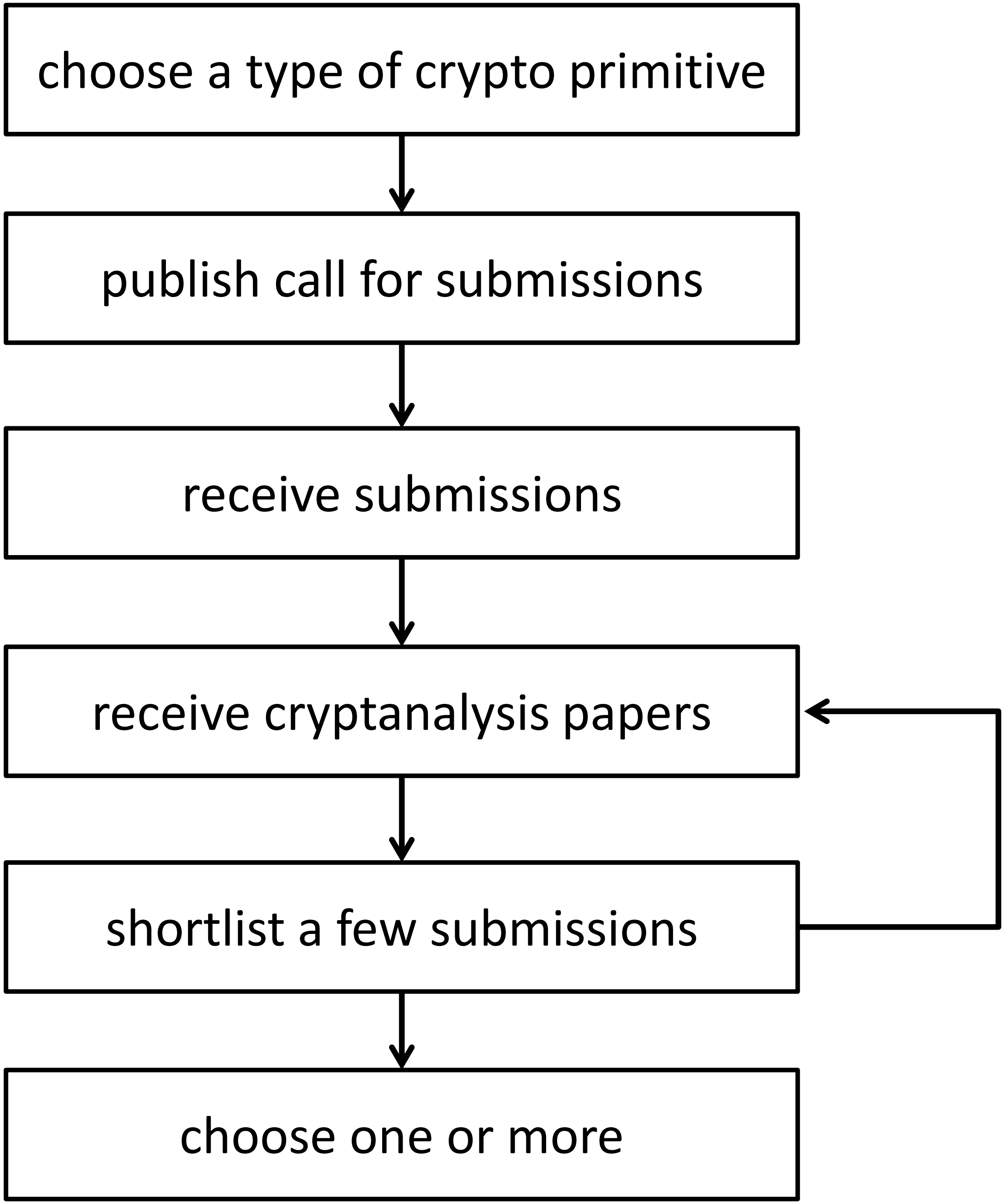
Crypto Competitions

Jean-Philippe Aumasson — Kudelski Security
https://131002.net | @veorq



choose a type of crypto primitive

publish call for submissions

receive submissions

receive cryptanalysis papers

shortlist a few submissions

choose one or more

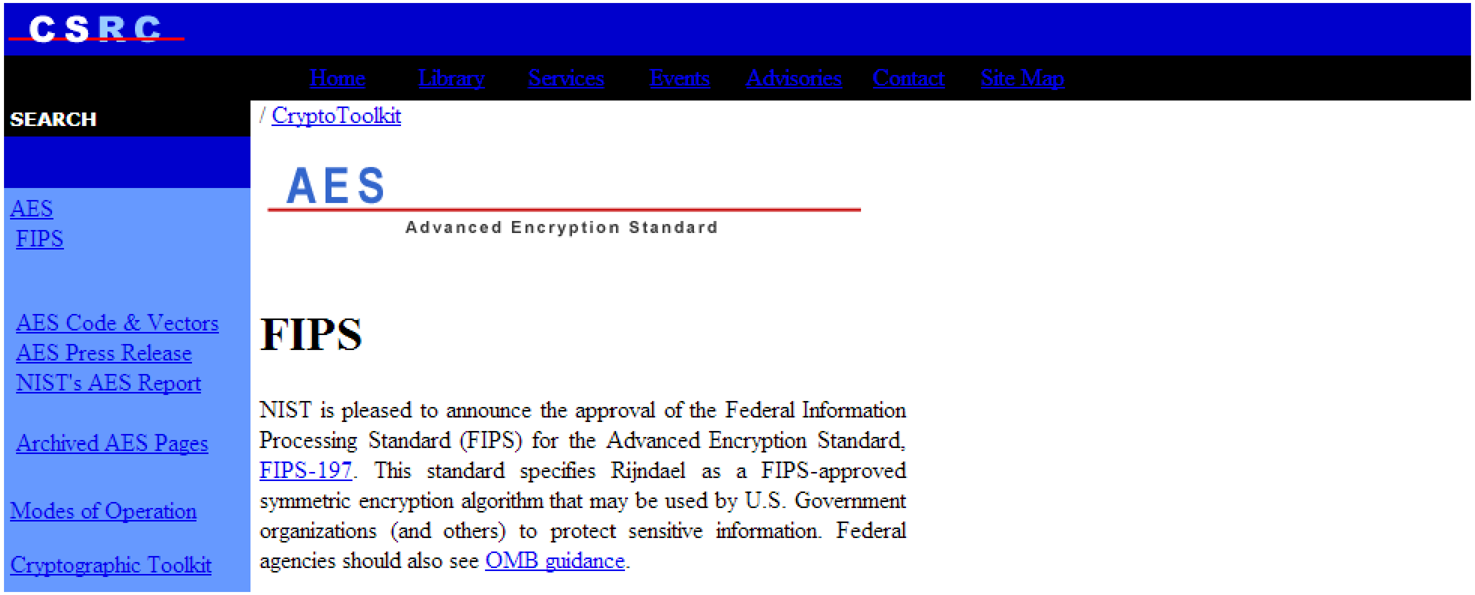
1997-2000: **AES** (NIST)

block ciphers

15 submissions

5 ‘finalists’

1 winner: Rijndael



FIPS

NIST is pleased to announce the approval of the Federal Information Processing Standard (FIPS) for the Advanced Encryption Standard, FIPS-197. This standard specifies Rijndael as a FIPS-approved symmetric encryption algorithm that may be used by U.S. Government organizations (and others) to protect sensitive information. Federal agencies should also see OMB guidance.

2004-2008: **eSTREAM** (ECRYPT)

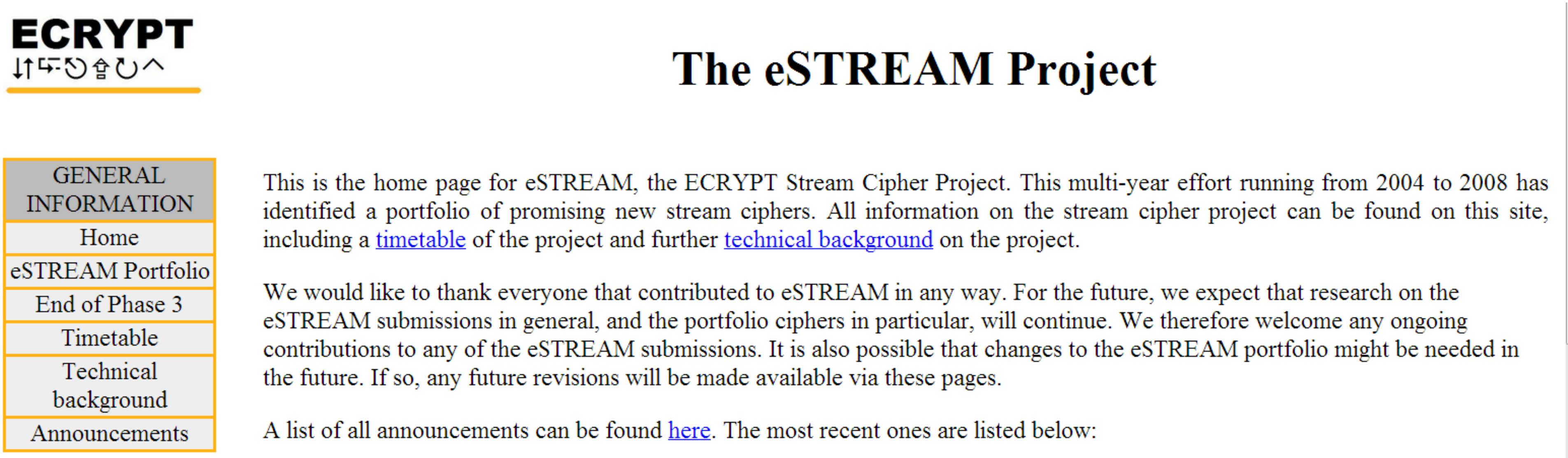
stream ciphers

34 submissions

27 ‘second-round’ candidates

16 ‘finalists’

portfolio of 8 (-1) winners



The eSTREAM Project

This is the home page for eSTREAM, the ECRYPT Stream Cipher Project. This multi-year effort running from 2004 to 2008 has identified a portfolio of promising new stream ciphers. All information on the stream cipher project can be found on this site, including a timetable of the project and further technical background on the project.

We would like to thank everyone that contributed to eSTREAM in any way. For the future, we expect that research on the eSTREAM submissions in general, and the portfolio ciphers in particular, will continue. We therefore welcome any ongoing contributions to any of the eSTREAM submissions. It is also possible that changes to the eSTREAM portfolio might be needed in the future. If so, any future revisions will be made available via these pages.

A list of all announcements can be found here. The most recent ones are listed below:

2007-2012: **SHA-3** (NIST)

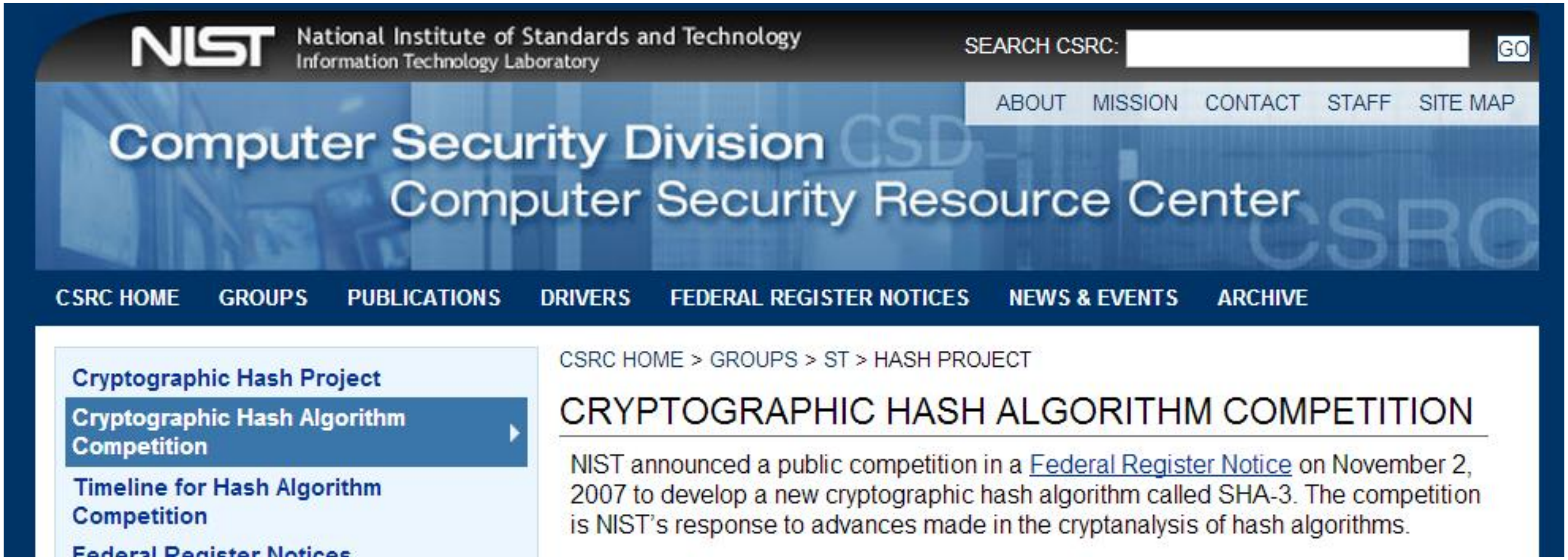
hash function

51 submissions

14 'second-round' candidates

5 'finalists'

1 winner: Keccak


NIST National Institute of Standards and Technology
Information Technology Laboratory
SEARCH CSRC:
GO
ABOUT
MISSION
CONTACT
STAFF
SITE MAP
Computer Security Division CSD
Computer Security Resource Center CSRC
CSRC HOME
GROUPS
PUBLICATIONS
DRIVERS
FEDERAL REGISTER NOTICES
NEWS & EVENTS
ARCHIVE
Cryptographic Hash Project
Cryptographic Hash Algorithm Competition
Timeline for Hash Algorithm Competition
Federal Register Notices

CSRC HOME > GROUPS > ST > HASH PROJECT

CRYPTOGRAPHIC HASH ALGORITHM COMPETITION

NIST announced a public competition in a Federal Register Notice on November 2, 2007 to develop a new cryptographic hash algorithm called SHA-3. The competition is NIST's response to advances made in the cryptanalysis of hash algorithms.

Cryptographic demolition derbies

(metaphor © Schneier)

survival of the ***fittest***

≈ balance strength/performance/etc.

incentive model

design great ciphers → **reputation++**

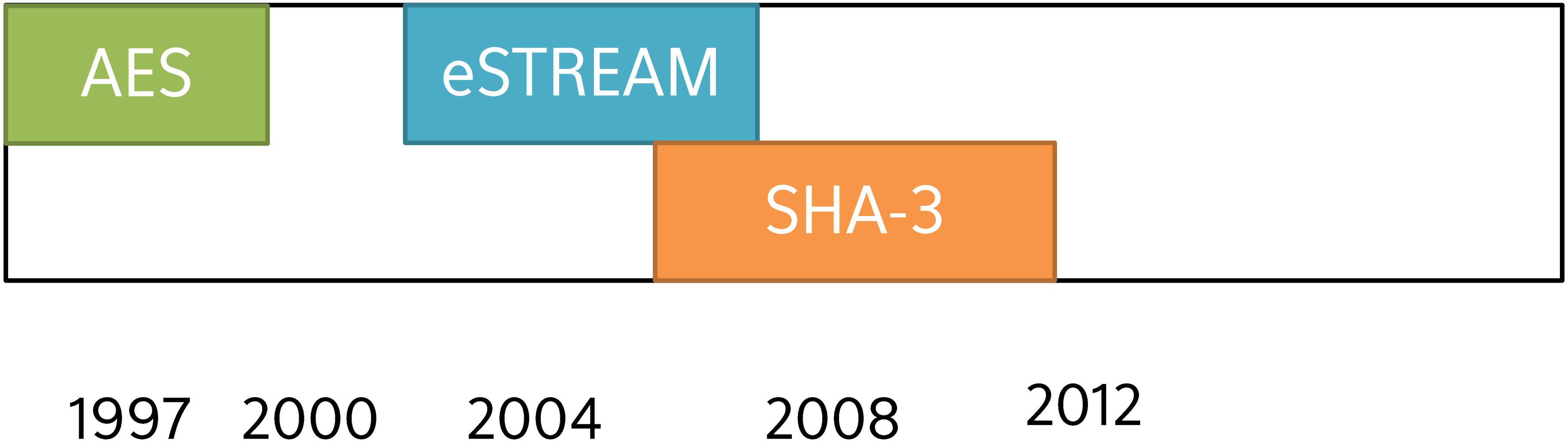
break candidate ciphers → **papers++**

design|analyze|implement → **grants++**

competition and conferences → **fun++**

->

free work for the organizers

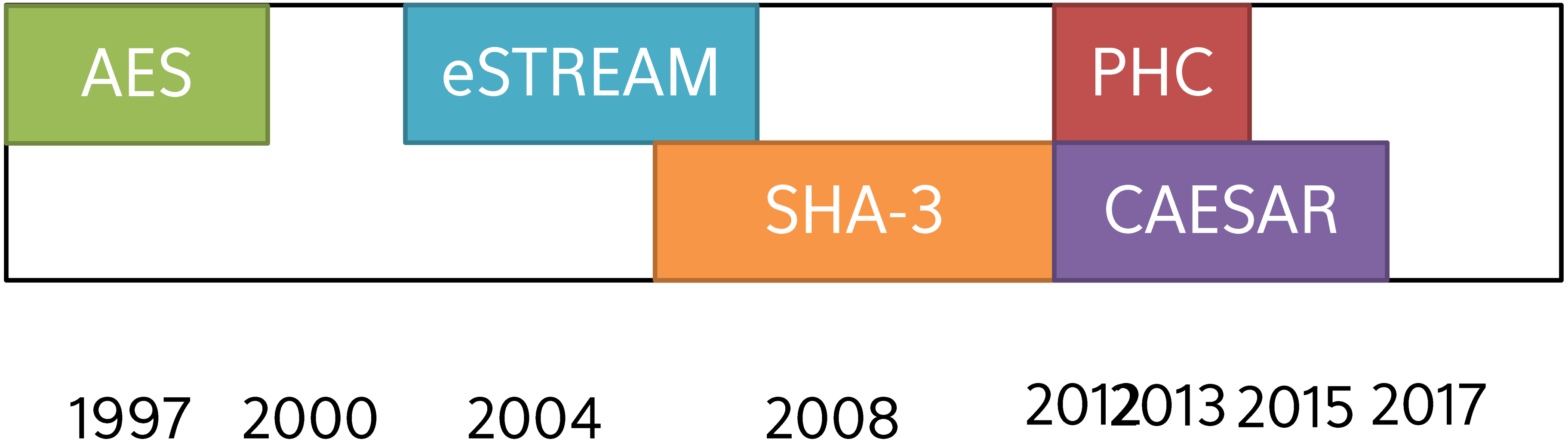

AES
eSTREAM
SHA-3
1997
2000
2004
2008
2012

block ciphers ✓

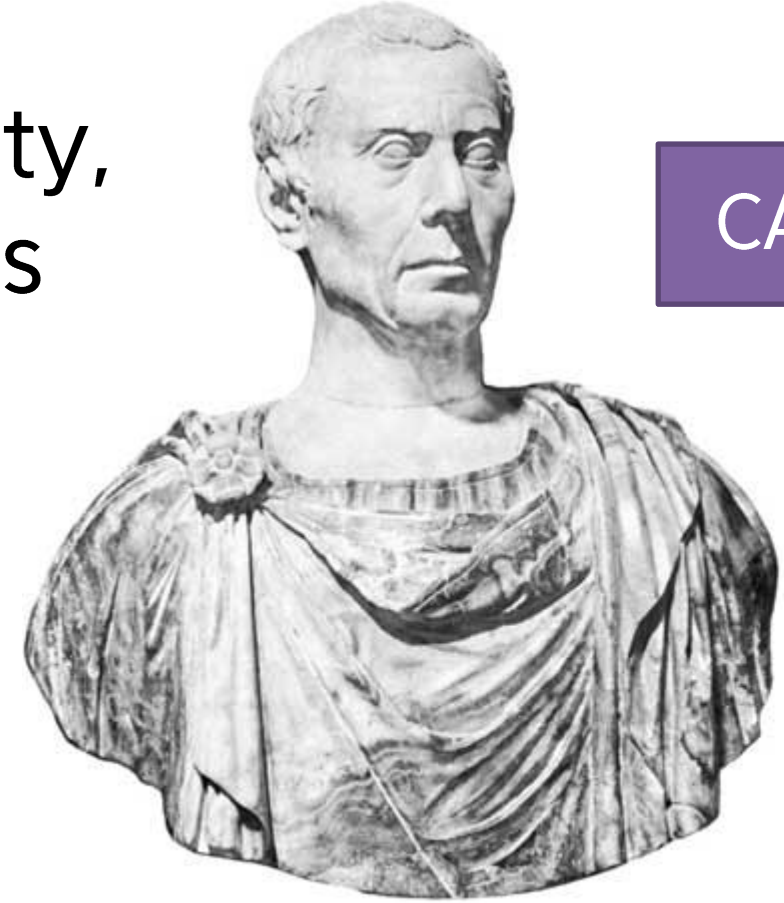
stream ciphers ✓

hash functions ✓

?


AES
eSTREAM
PHC
SHA-3
CAESAR
1997
2000
2004
2008
2012
2013
2015
2017

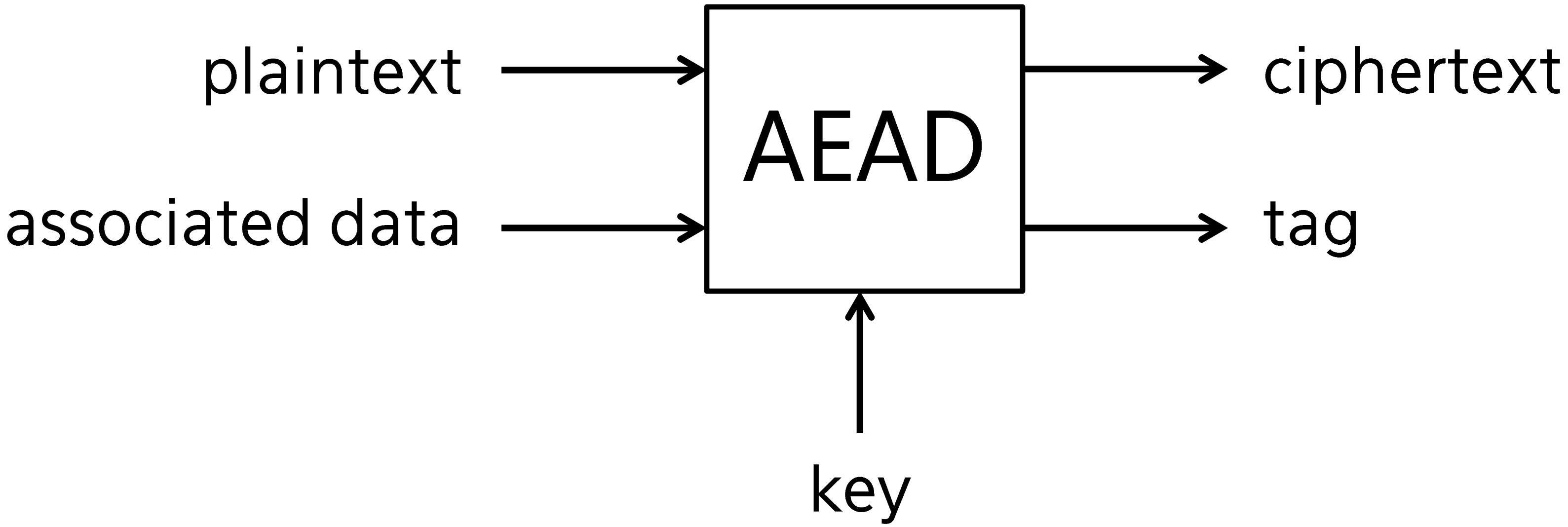
Competition for
Authenticated
Encryption:
Security,
Applicability,
Robustness



CAESAR

“CAESAR will identify a portfolio of **authenticated ciphers** that offer advantages over AES-GCM and that are suitable for widespread adoption.”

http://competitions.cr.yp.to/caesar-call-2.html



AES-GCM: the “most standard” AEAD

(NIST SP 800-38D)

CTR-like parallelism and pipelining

1-pass streaming (“online”) mode

static/constant AD can be preprocessed

no need for decryption function

poor bounds for short tags or long messages

not that efficient with no AES-NI

requires GF(2^{128}) arithmetic

complicated/bug-prone

CAESAR design choices

target platform(s)/application(s)

- native **64-bit** desktop/server software
  - can exploit AES-NI, SSE*, AVX2, etc. instruction sets
  - memory generally not an issue
  - baseline AES-GCM or AES-OCB
- **32- to 64-bit** software
  - avoid dependency on a technology (e.g. instruction)
  - memory generally not an issue
- **low-end**
  - software (AVR, PIC, etc., e.g. for SCADA)
  - hardware (RFID tags, cheap chips)
  - many lightweight designs available (also from NSA)

CAESAR design choices

stream cipher or block cipher based?

– AES reuse has pros and cons...

nonce-based or not? if yes:

– what happens if nonces are repeated?

– how long should be the nonce?

associated data flexibility

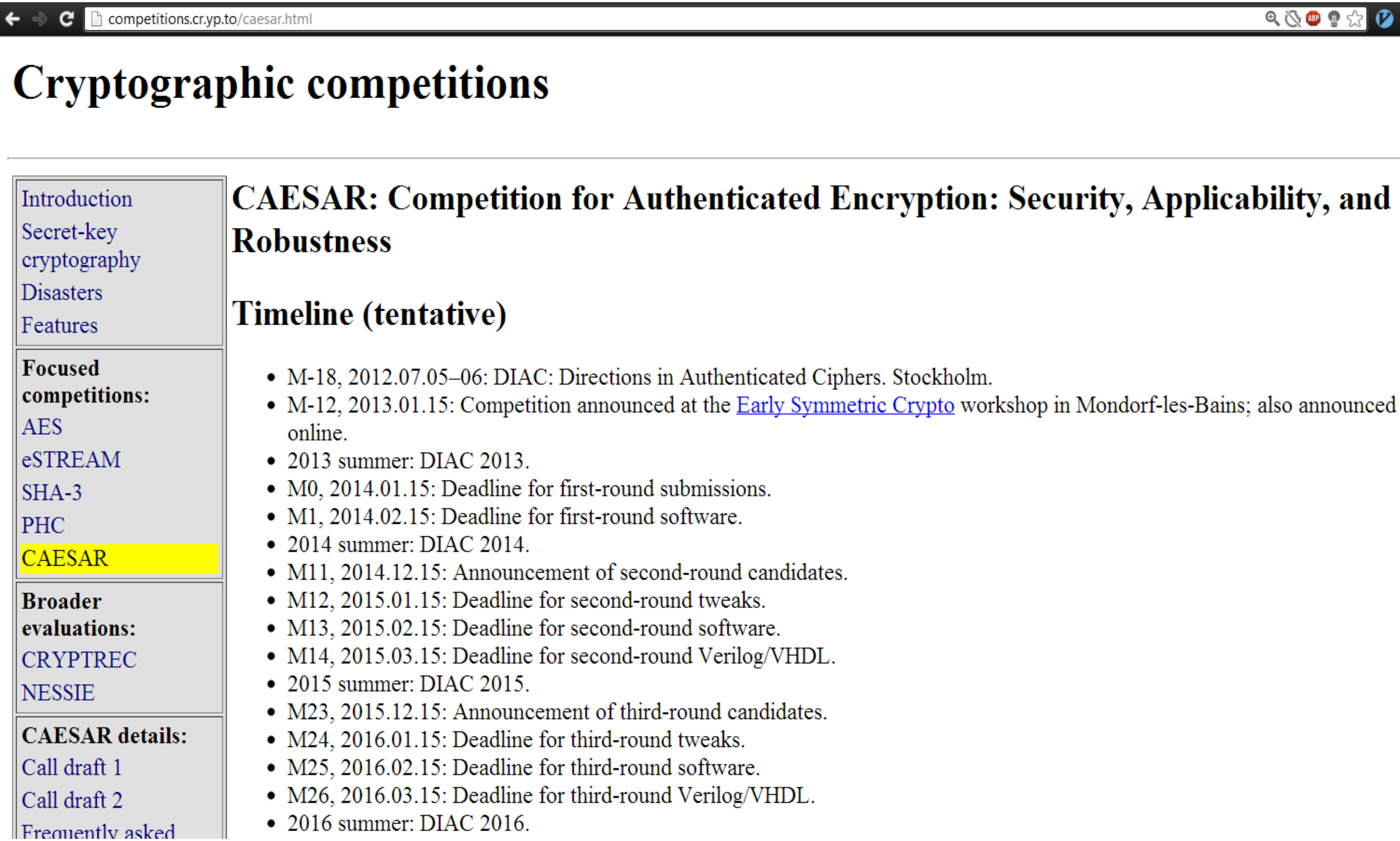
– at the beginning, end, or anywhere?

provable security

– rely on existing proved mode, or new one?

– relax provsec features for better performance?

CAESAR submission deadline: **Jan 15, 2014**

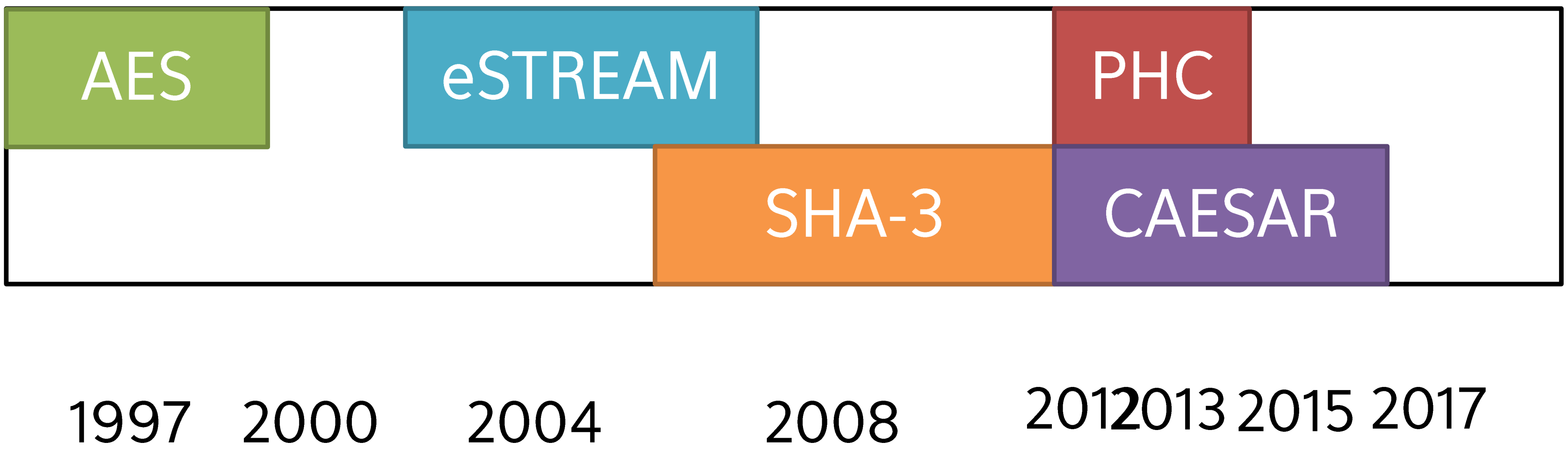


Cryptographic competitions

CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness

Timeline (tentative)

- M-18, 2012.07.05–06: DIAC: Directions in Authenticated Ciphers. Stockholm.
- M-12, 2013.01.15: Competition announced at the Early Symmetric Crypto workshop in Mondorf-les-Bains; also announced online.
- 2013 summer: DIAC 2013.
- M0, 2014.01.15: Deadline for first-round submissions.
- M1, 2014.02.15: Deadline for first-round software.
- 2014 summer: DIAC 2014.
- M11, 2014.12.15: Announcement of second-round candidates.
- M12, 2015.01.15: Deadline for second-round tweaks.
- M13, 2015.02.15: Deadline for second-round software.
- M14, 2015.03.15: Deadline for second-round Verilog/VHDL.
- 2015 summer: DIAC 2015.
- M23, 2015.12.15: Announcement of third-round candidates.
- M24, 2016.01.15: Deadline for third-round tweaks.
- M25, 2016.02.15: Deadline for third-round software.
- M26, 2016.03.15: Deadline for third-round Verilog/VHDL.
- 2016 summer: DIAC 2016.



AES

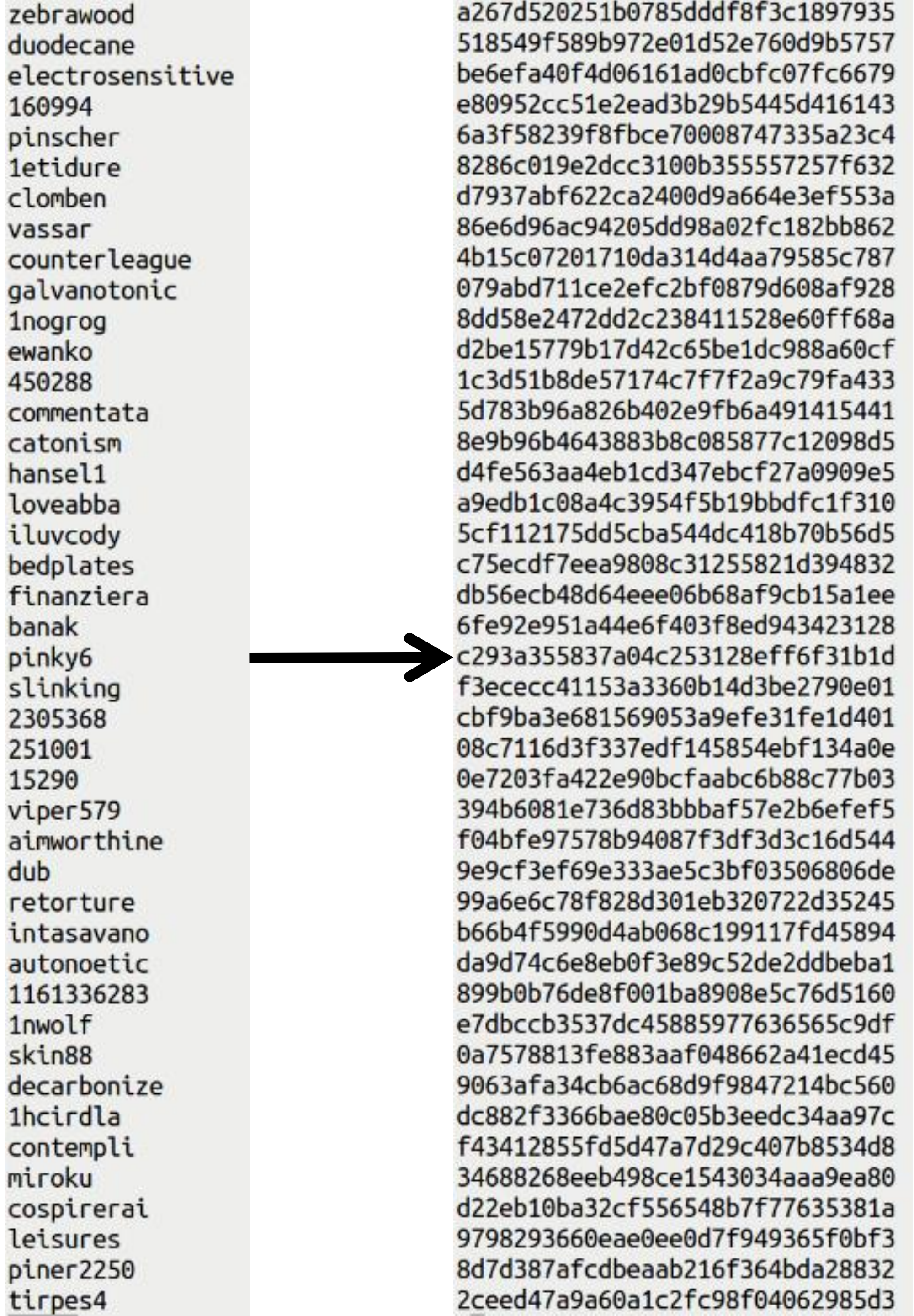
eSTREAM

PHC

SHA-3

CAESAR

1997 2000 2004 2008 2012 2013 2015 2017



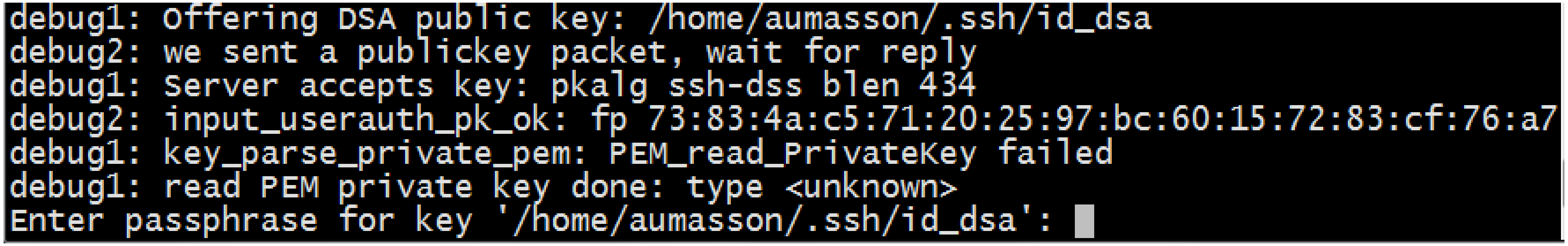
```
zebrawood           a267d520251b0785dddf8f3c1897935
duodecane           518549f589b972e01d52e760d9b5757
electrosensitive    be6efa40f4d06161ad0cbfc07fc6679
160994              e80952cc51e2ead3b29b5445d416143
pinscher            6a3f58239f8fbce70008747335a23c4
1etidure            8286c019e2dcc3100b355557257f632
clomben             d7937abf622ca2400d9a664e3ef553a
vassar              86e6d96ac94205dd98a02fc182bb862
counterleague       4b15c07201710da314d4aa79585c787
galvanotonic        079abd711ce2efc2bf0879d608af928
1nogrog             8dd58e2472dd2c238411528e60ff68a
ewanko              d2be15779b17d42c65be1dc988a60cf
450288              1c3d51b8de57174c7f7f2a9c79fa433
commentata          5d783b96a826b402e9fb6a491415441
catonism            8e9b96b4643883b8c085877c12098d5
hansel1             d4fe563aa4eb1cd347ebcf27a0909e5
loveabba            a9edb1c08a4c3954f5b19bbdfc1f310
iluvcody            5cf112175dd5cba544dc418b70b56d5
bedplates           c75ecdf7eea9808c31255821d394832
finanziera          db56ecb48d64eee06b68af9cb15a1ee
banak               6fe92e951a44e6f403f8ed943423128
pinky6              c293a355837a04c253128eff6f31b1d
slinking            f3ececc41153a3360b14d3be2790e01
2305368             cbf9ba3e681569053a9efe31fe1d401
251001              08c7116d3f337edf145854ebf134a0e
15290               0e7203fa422e90bcfaabc6b88c77b03
viper579            394b6081e736d83bbbaf57e2b6efef5
aimworthine         f04bfe97578b94087f3df3d3c16d544
dub                 9e9cf3ef69e333ae5c3bf03506806de
retorture           99a6e6c78f828d301eb320722d35245
intasavano          b66b4f5990d4ab068c199117fd45894
autonoetic          da9d74c6e8eb0f3e89c52de2ddbeba1
1161336283          899b0b76de8f001ba8908e5c76d5160
1nwolf              e7dbccb3537dc45885977636565c9df
skin88              0a7578813fe883aaf048662a41ecd45
decarbonize         9063afa34cb6ac68d9f9847214bc560
1hcirdla            dc882f3366bae80c05b3eedc34aa97c
contempli           f43412855fd5d47a7d29c407b8534d8
miroku              34688268eeb498ce1543034aaa9ea80
cospirerai          d22eb10ba32cf556548b7f77635381a
leisures            9798293660eae0ee0d7f949365f0bf3
piner2250           8d7d387afcdbeaab216f364bda28832
tirpes4             2ceed47a9a60a1c2fc98f04062985d3
```

PHC

Password
Hashing
Competition

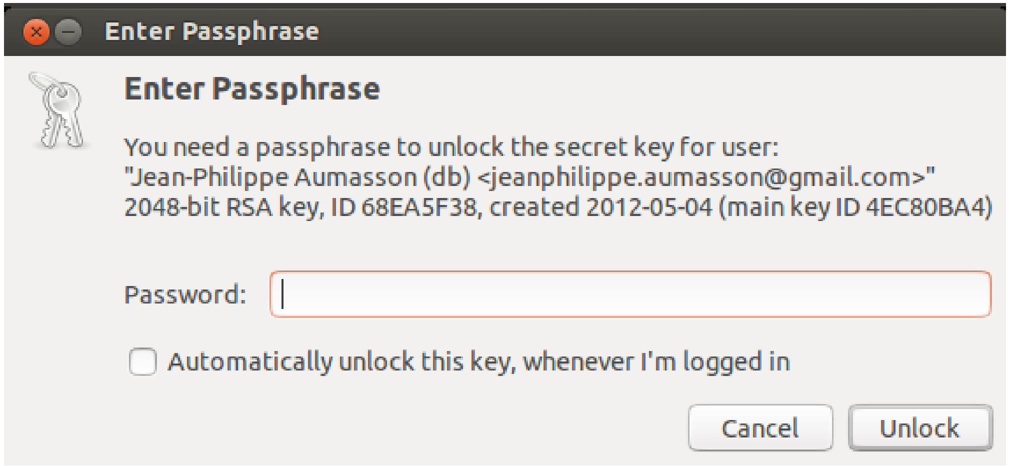
Password-based key derivation

produces an encryption key from a password



```
debug1: Offering DSA public key: /home/aumasson/.ssh/id_dsa
debug2: we sent a publickey packet, wait for reply
debug1: Server accepts key: pkalg ssh-dss blen 434
debug2: input_userauth_pk_ok: fp 73:83:4a:c5:71:20:25:97:bc:60:15:72:83:cf:76:a7
debug1: key_parse_private_pem: PEM_read_PrivateKey failed
debug1: read PEM private key done: type <unknown>
Enter passphrase for key '/home/aumasson/.ssh/id_dsa': 
```

for protection of SSH private keys, password-based encryption, full-disk encryption, etc.

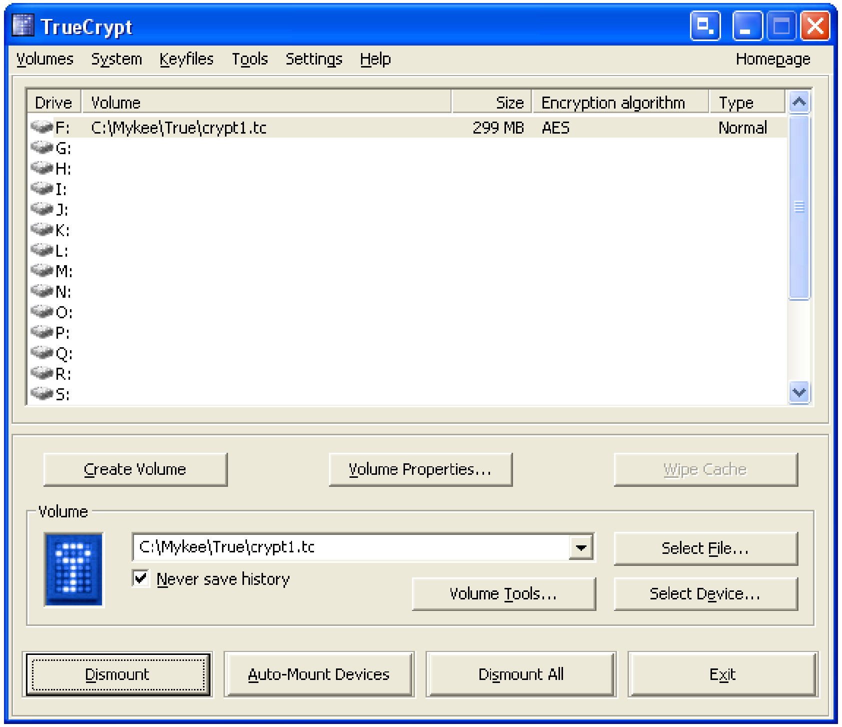


Enter Passphrase

You need a passphrase to unlock the secret key for user:
"Jean-Philippe Aumasson (db) <jeanphilippe.aumasson@gmail.com>"
2048-bit RSA key, ID 68EA5F38, created 2012-05-04 (main key ID 4EC80BA4)

Password: |

□ Automatically unlock this key, whenever I'm logged in



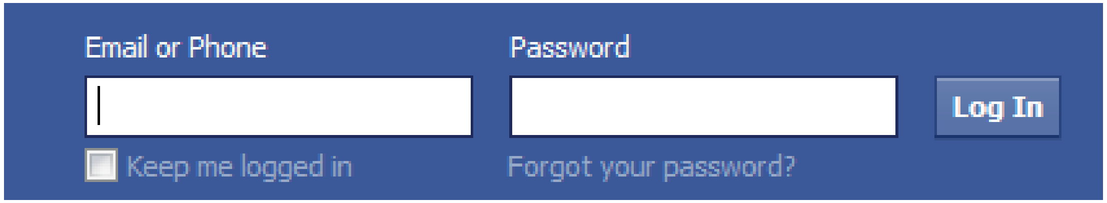
Drive	Volume	Size	Encryption algorithm	Type
F:	C:\Mykee\True\crypt1.tc	299 MB	AES	Normal
G:				
H:				
I:				
J:				
K:				
L:				
M:				
N:				
O:				
P:				
Q:				
R:				
S:				

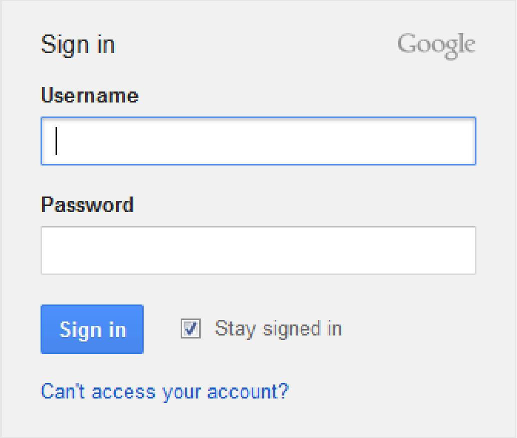
Volume: C:\Mykee\True\crypt1.tc

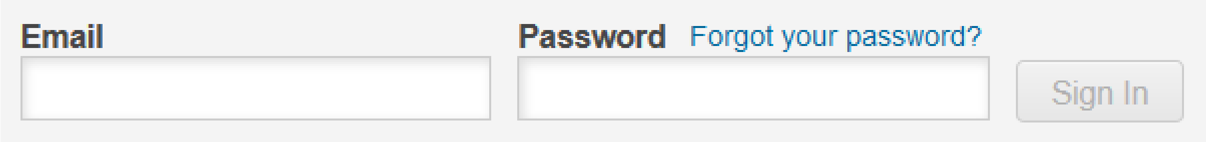
☑ Never save history

Password-based authentication

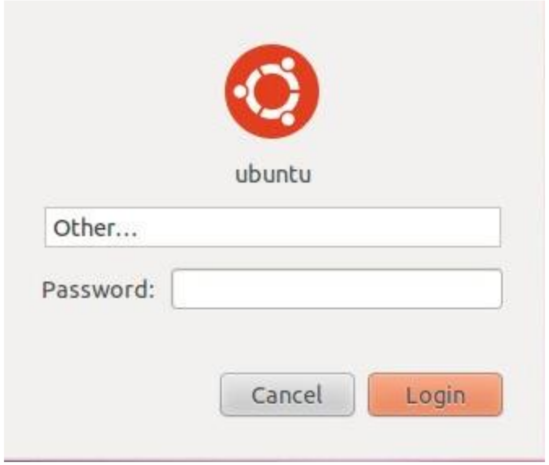
server checks submitted pwd against its DB's

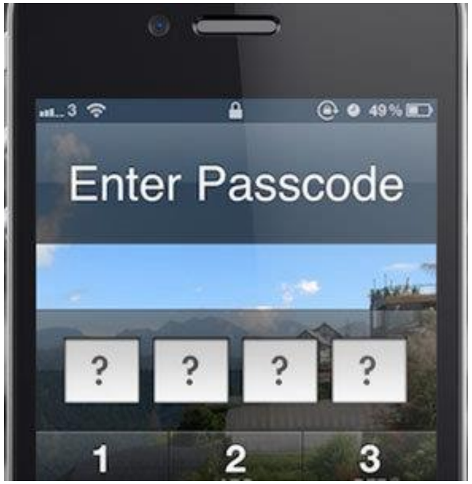


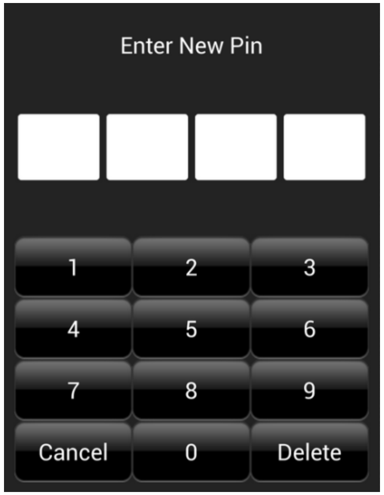




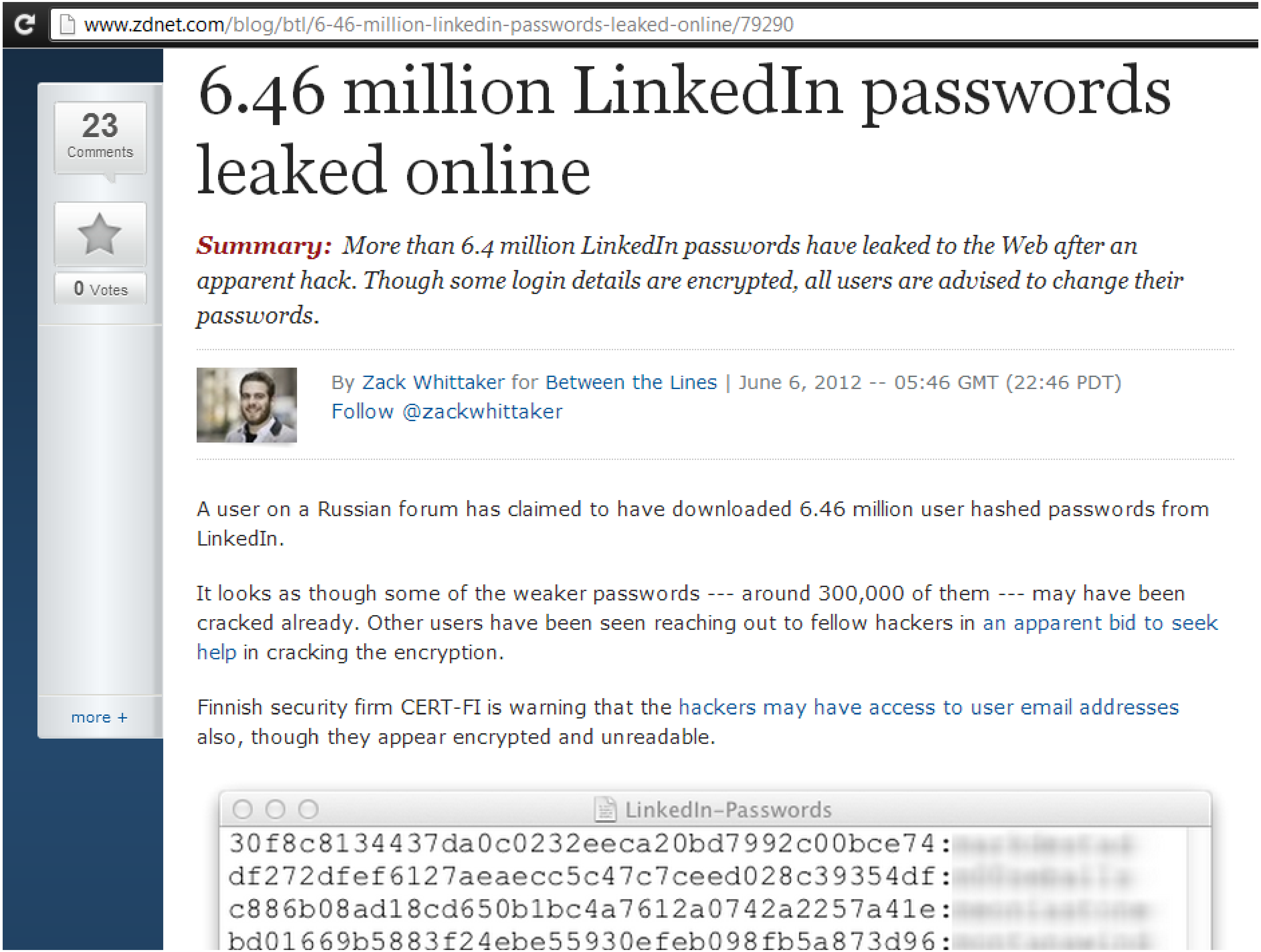
web services, OS user auth, mobile PINs, etc.







So what’s the problem?


www.zdnet.com/blog/btl/6-46-million-linkedin-passwords-leaked-online/79290
23 Comments
0 Votes
more +

6.46 million LinkedIn passwords leaked online

Summary: *More than 6.4 million LinkedIn passwords have leaked to the Web after an apparent hack. Though some login details are encrypted, all users are advised to change their passwords.*

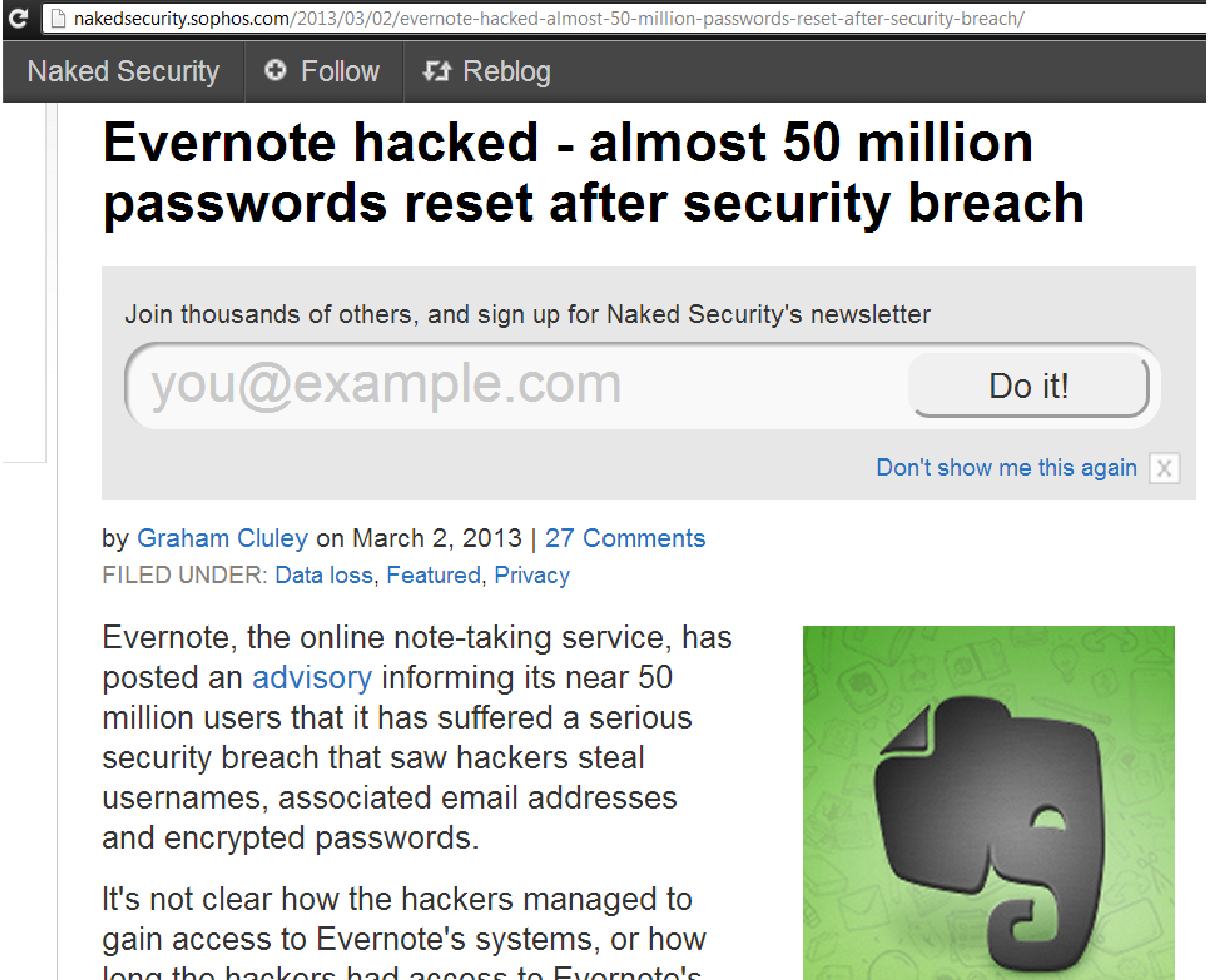
By Zack Whittaker for Between the Lines | June 6, 2012 -- 05:46 GMT (22:46 PDT)
Follow @zackwhittaker

A user on a Russian forum has claimed to have downloaded 6.46 million user hashed passwords from LinkedIn.

It looks as though some of the weaker passwords --- around 300,000 of them --- may have been cracked already. Other users have been seen reaching out to fellow hackers in an apparent bid to seek help in cracking the encryption.

Finnish security firm CERT-FI is warning that the hackers may have access to user email addresses also, though they appear encrypted and unreadable.

```
30f8c8134437da0c0232eeca20bd7992c00bce74:[redacted]
df272dfef6127aeaecc5c47c7ceed028c39354df:[redacted]
c886b08ad18cd650b1bc4a7612a0742a2257a41e:[redacted]
bd01669b5883f24ebe55930efeb098fb5a873d96:[redacted]
```



nakedsecurity.sophos.com/2013/03/02/evernote-hacked-almost-50-million-passwords-reset-after-security-breach/

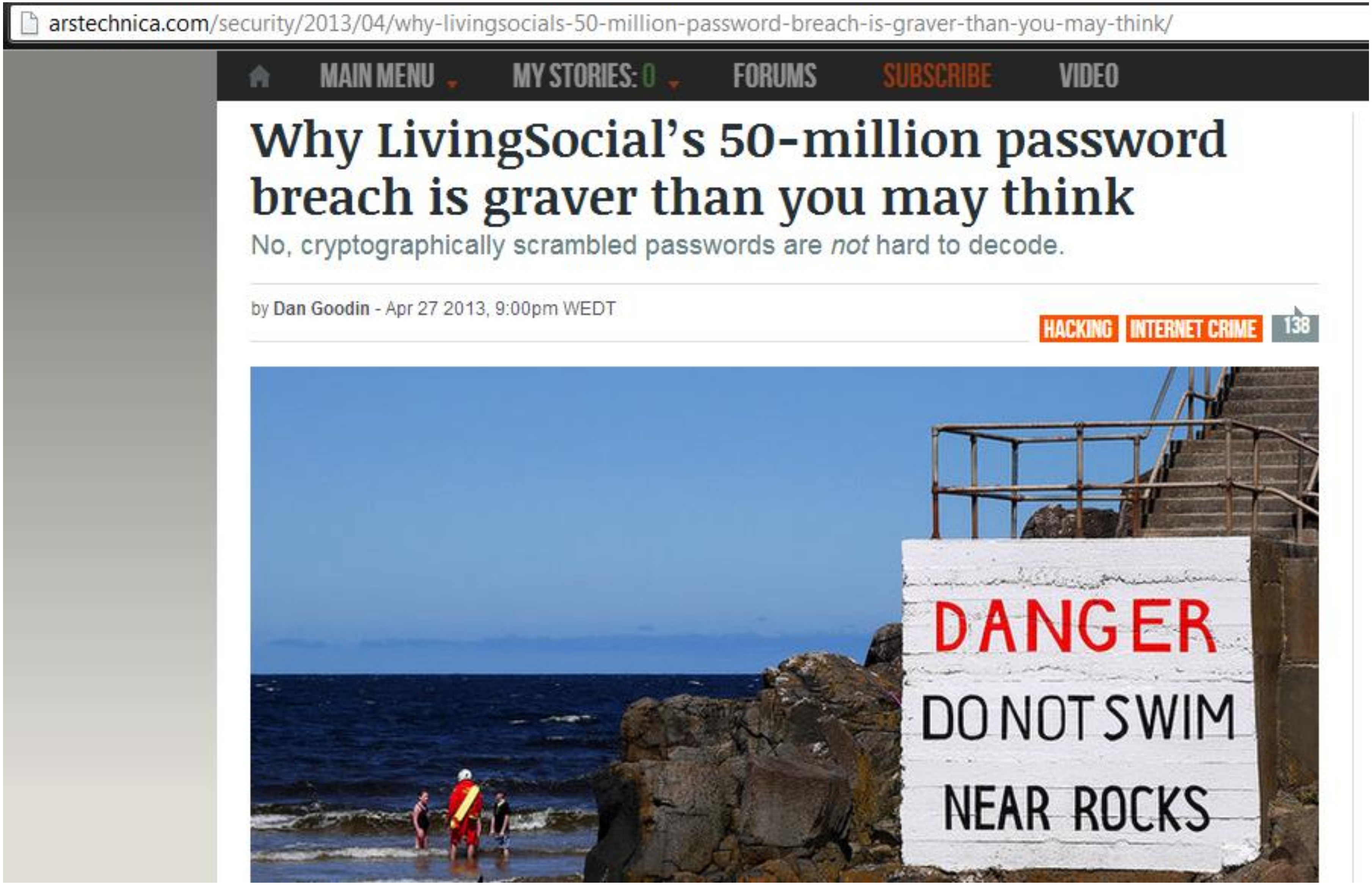
Evernote hacked - almost 50 million passwords reset after security breach

by Graham Cluley on March 2, 2013 | 27 Comments

FILED UNDER: Data loss, Featured, Privacy

Evernote, the online note-taking service, has posted an advisory informing its near 50 million users that it has suffered a serious security breach that saw hackers steal usernames, associated email addresses and encrypted passwords.

It's not clear how the hackers managed to gain access to Evernote's systems, or how long the hackers had access to Evernote's

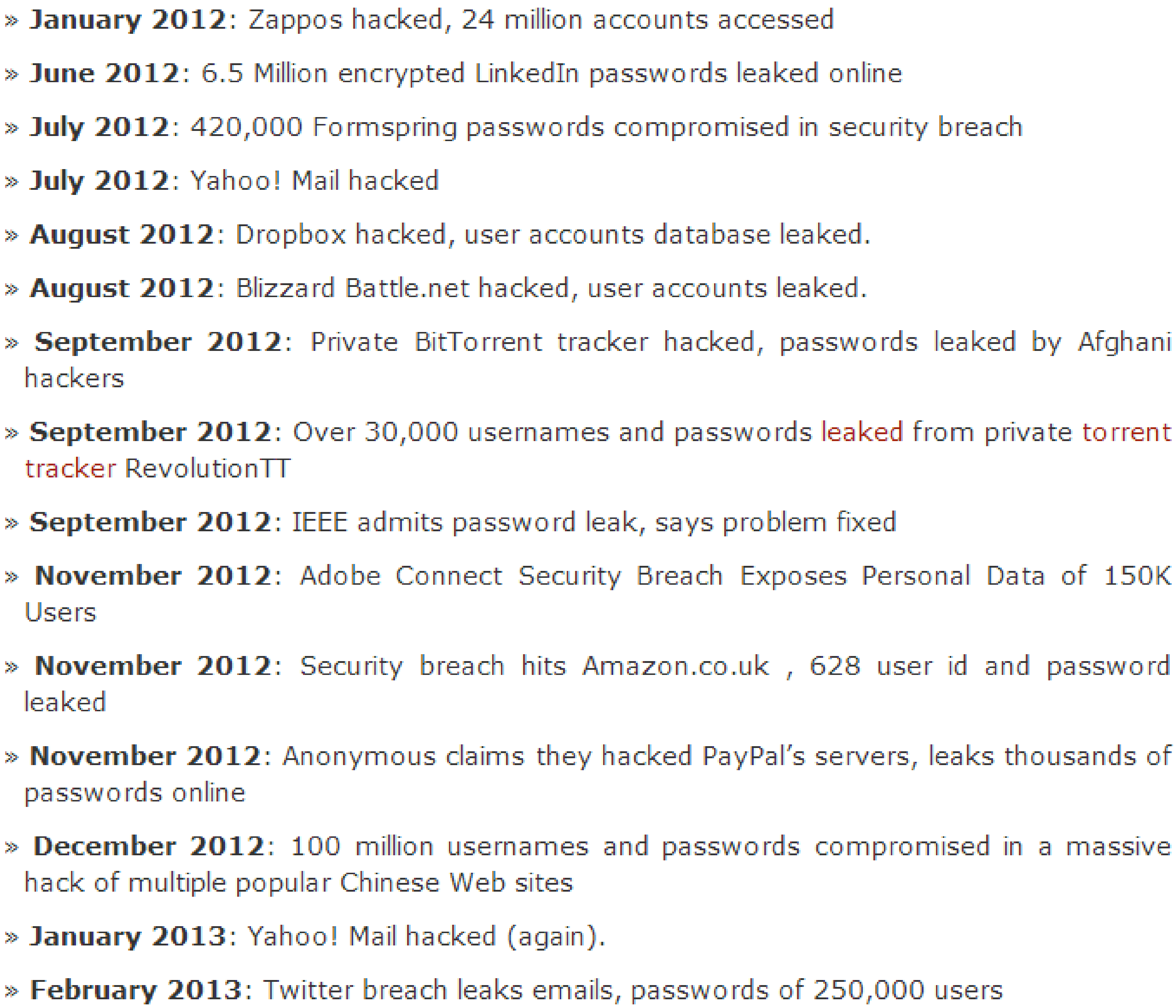


arstechnica.com/security/2013/04/why-livingsocials-50-million-password-breach-is-graver-than-you-may-think/

Why LivingSocial's 50-million password breach is graver than you may think

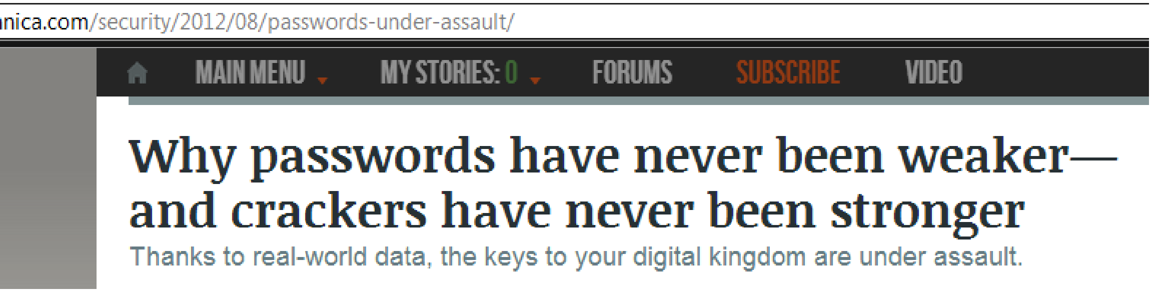
No, cryptographically scrambled passwords are *not* hard to decode.

by Dan Goodin - Apr 27 2013, 9:00pm WEDT



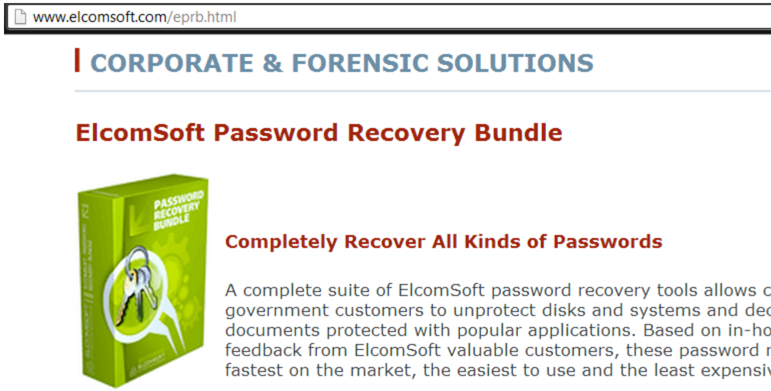
- » **January 2012**: Zappos hacked, 24 million accounts accessed
- » **June 2012**: 6.5 Million encrypted LinkedIn passwords leaked online
- » **July 2012**: 420,000 Formspring passwords compromised in security breach
- » **July 2012**: Yahoo! Mail hacked
- » **August 2012**: Dropbox hacked, user accounts database leaked.
- » **August 2012**: Blizzard Battle.net hacked, user accounts leaked.
- » **September 2012**: Private BitTorrent tracker hacked, passwords leaked by Afghani hackers
- » **September 2012**: Over 30,000 usernames and passwords leaked from private torrent tracker RevolutionTT
- » **September 2012**: IEEE admits password leak, says problem fixed
- » **November 2012**: Adobe Connect Security Breach Exposes Personal Data of 150K Users
- » **November 2012**: Security breach hits Amazon.co.uk , 628 user id and password leaked
- » **November 2012**: Anonymous claims they hacked PayPal's servers, leaks thousands of passwords online
- » **December 2012**: 100 million usernames and passwords compromised in a massive hack of multiple popular Chinese Web sites
- » **January 2013**: Yahoo! Mail hacked (again).
- » **February 2013**: Twitter breach leaks emails, passwords of 250,000 users

it’s just **hashes**, my password is safe!



Why passwords have never been weaker—and crackers have never been stronger

Thanks to real-world data, the keys to your digital kingdom are under assault.

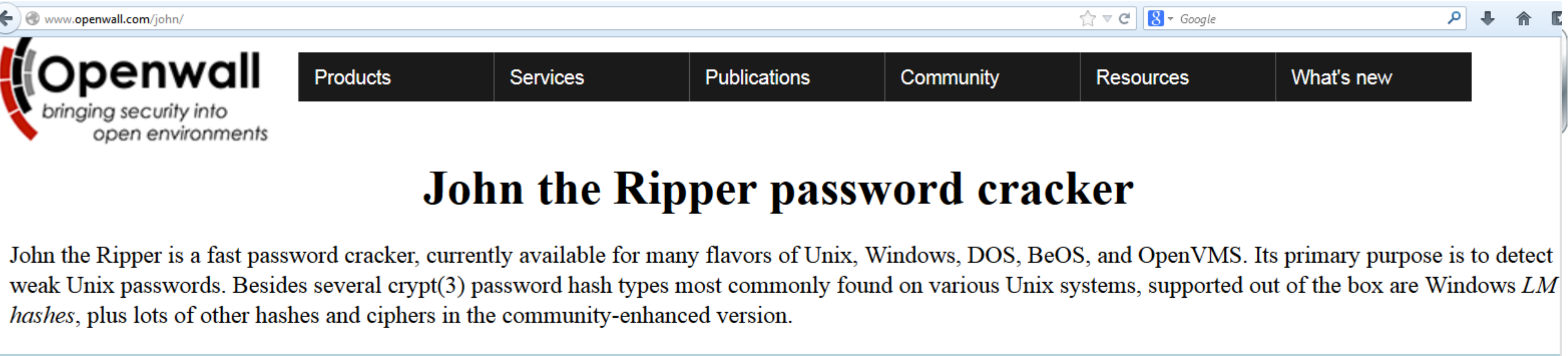


CORPORATE & FORENSIC SOLUTIONS

ElcomSoft Password Recovery Bundle

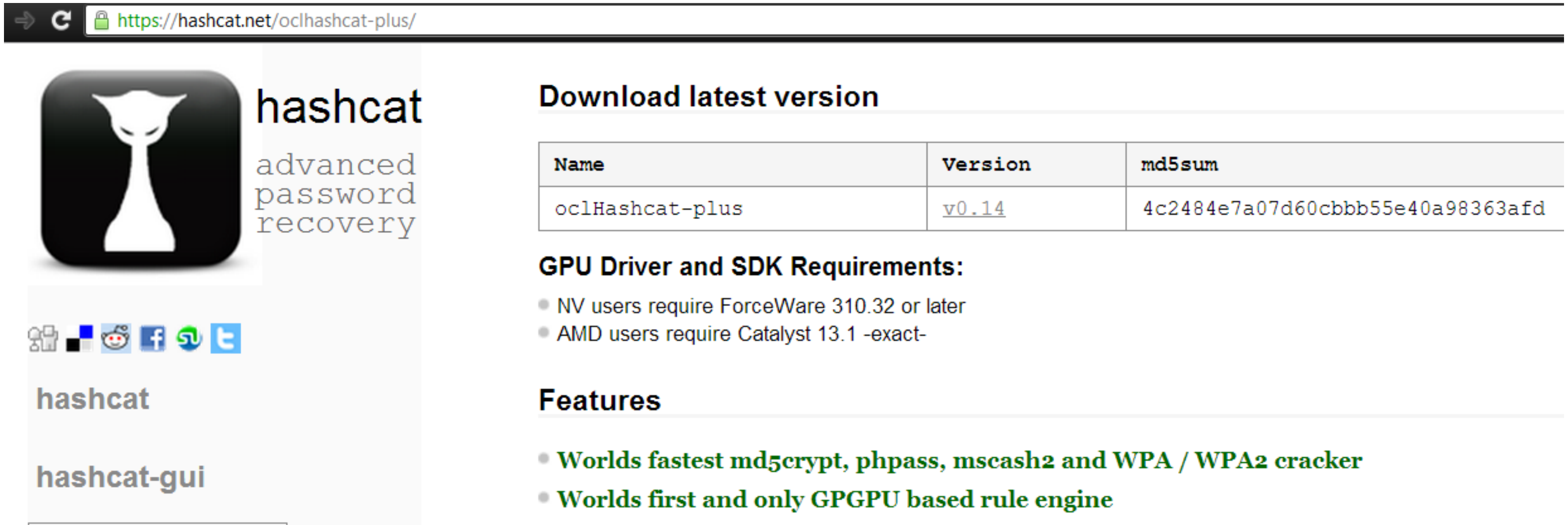
Completely Recover All Kinds of Passwords

A complete suite of ElcomSoft password recovery tools allows c
government customers to unprotect disks and systems and dec
documents protected with popular applications. Based on in-ho
feedback from ElcomSoft valuable customers, these password r
fastest on the market, the easiest to use and the least expensiv



John the Ripper password cracker

John the Ripper is a fast password cracker, currently available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix systems, supported out of the box are Windows *LM hashes*, plus lots of other hashes and ciphers in the community-enhanced version.


https://hashcat.net/oclhashcat-plus/
hashcat
advanced password recovery
hashcat
hashcat-gui

Download latest version

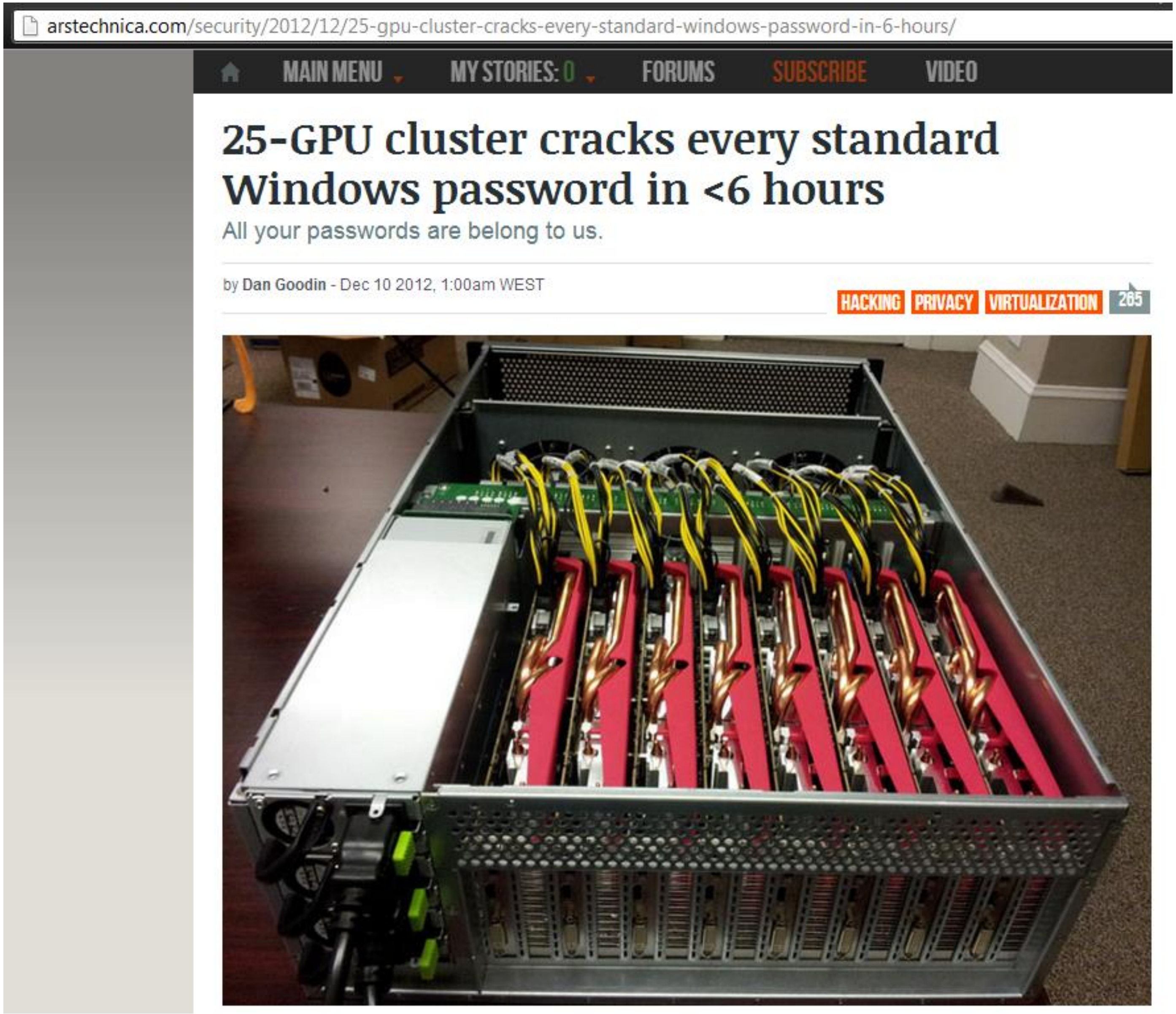
Name	Version	md5sum
oclHashcat-plus	v0.14	4c2484e7a07d60cbbb55e40a98363afd

GPU Driver and SDK Requirements:

- NV users require ForceWare 310.32 or later
- AMD users require Catalyst 13.1 -exact-

Features

- **Worlds fastest md5crypt, phpass, mscash2 and WPA / WPA2 cracker**
- **Worlds first and only GPGPU based rule engine**


arstechnica.com/security/2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours/
MAIN MENU
MY STORIES: 0
FORUMS
SUBSCRIBE
VIDEO

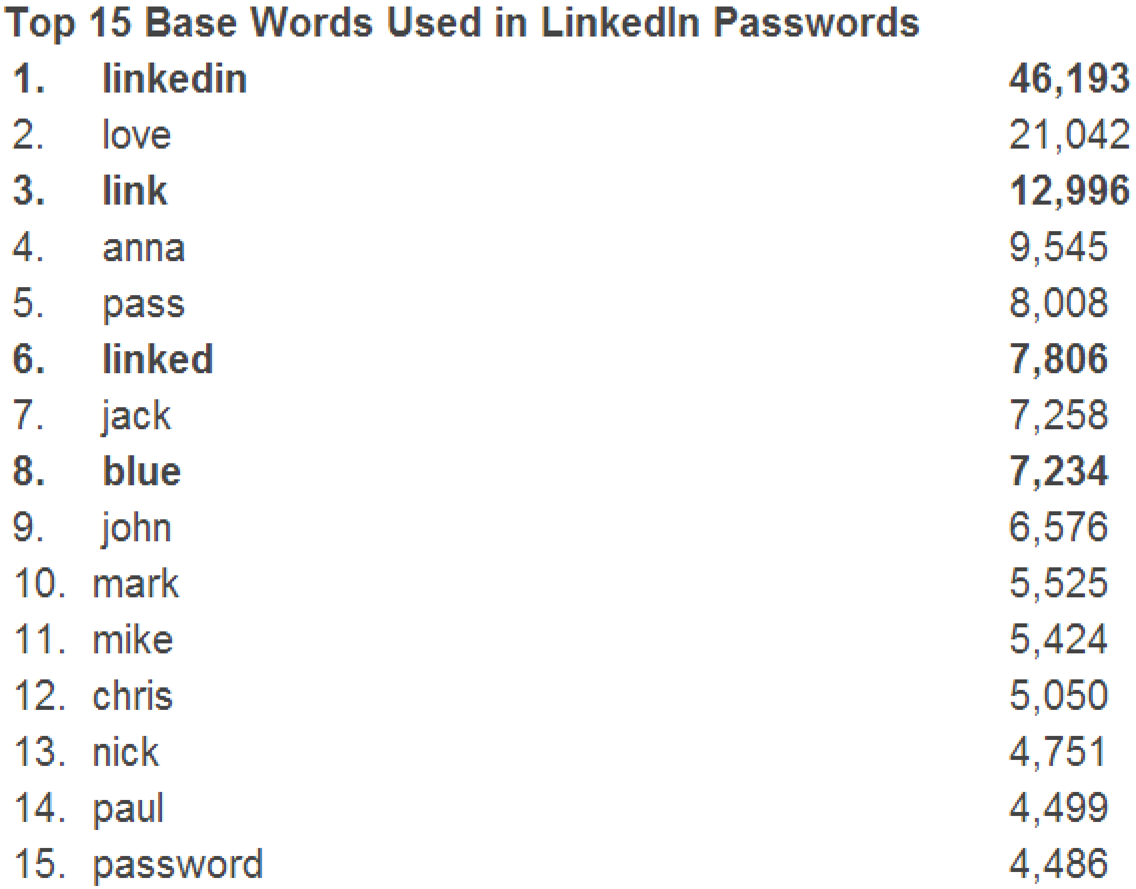
25-GPU cluster cracks every standard Windows password in <6 hours

All your passwords are belong to us.

by Dan Goodin - Dec 10 2012, 1:00am WEST

HACKING
PRIVACY
VIRTUALIZATION
265

90+% of LinkedIn passwords cracked

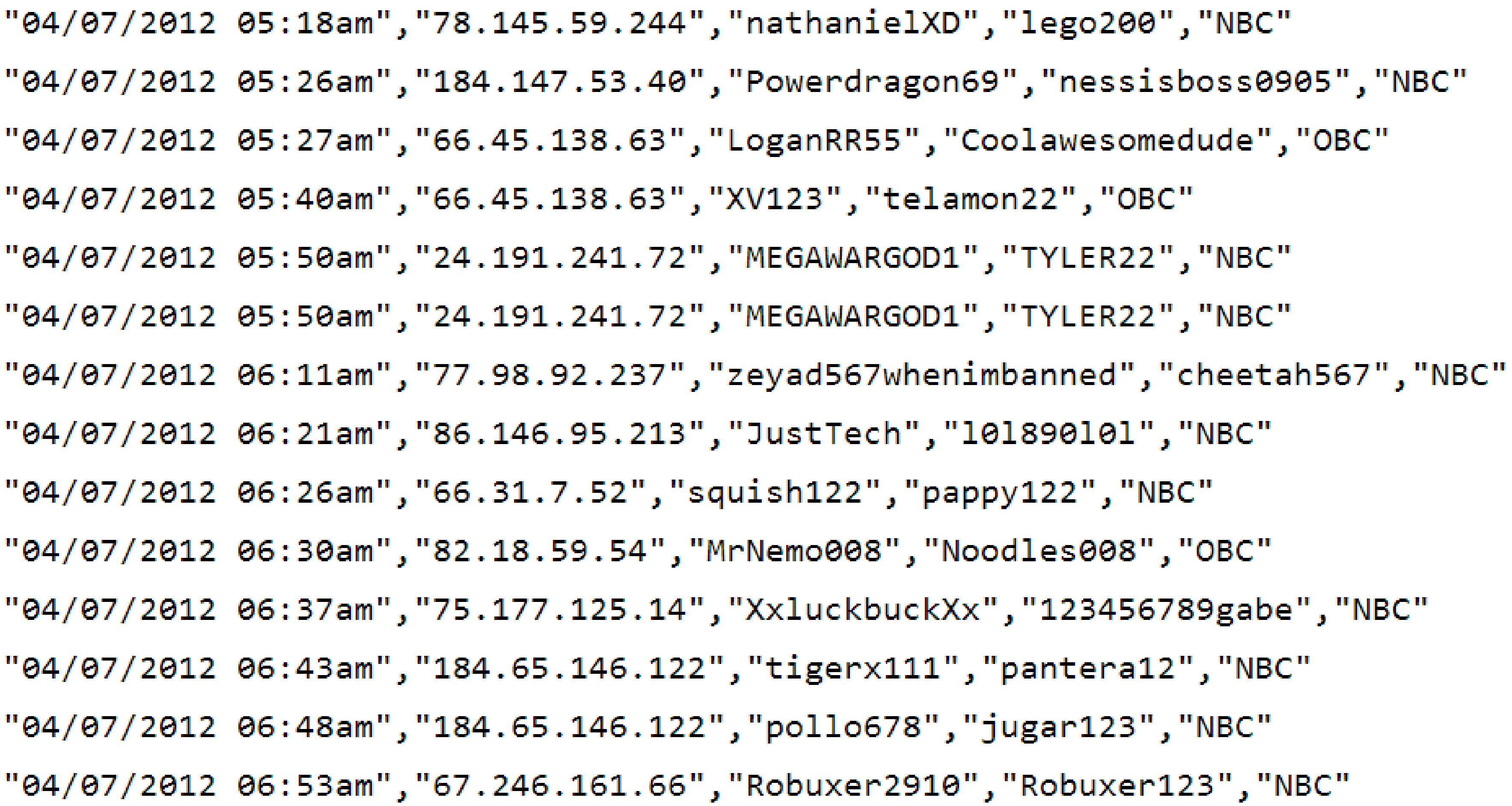


Top 15 Base Words Used in LinkedIn Passwords

1.	**linkedin**	**46,193**
2.	love	21,042
3.	**link**	**12,996**
4.	anna	9,545
5.	pass	8,008
6.	**linked**	**7,806**
7.	jack	7,258
8.	**blue**	**7,234**
9.	john	6,576
10.	mark	5,525
11.	mike	5,424
12.	chris	5,050
13.	nick	4,751
14.	paul	4,499
15.	password	4,486

How (not) to store passwords

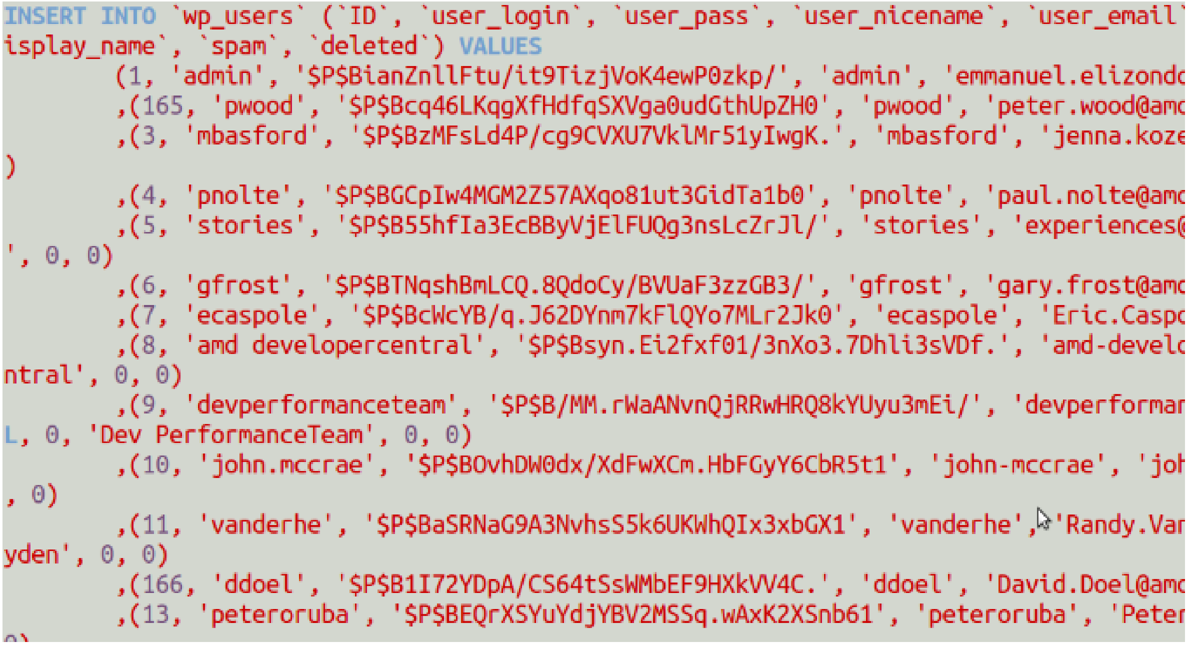
D- cleartext



```
"04/07/2012 05:18am","78.145.59.244","nathanielXD","lego200","NBC"
"04/07/2012 05:26am","184.147.53.40","Powerdragon69","nessisboss0905","NBC"
"04/07/2012 05:27am","66.45.138.63","LoganRR55","Coolawesomedude","OBC"
"04/07/2012 05:40am","66.45.138.63","XV123","telamon22","OBC"
"04/07/2012 05:50am","24.191.241.72","MEGAWARGOD1","TYLER22","NBC"
"04/07/2012 05:50am","24.191.241.72","MEGAWARGOD1","TYLER22","NBC"
"04/07/2012 06:11am","77.98.92.237","zeyad567whenimbanned","cheetah567","NBC"
"04/07/2012 06:21am","86.146.95.213","JustTech","101890101","NBC"
"04/07/2012 06:26am","66.31.7.52","squish122","pappy122","NBC"
"04/07/2012 06:30am","82.18.59.54","MrNemo008","Noodles008","OBC"
"04/07/2012 06:37am","75.177.125.14","XxluckbuckXx","123456789gabe","NBC"
"04/07/2012 06:43am","184.65.146.122","tigerx111","pantera12","NBC"
"04/07/2012 06:48am","184.65.146.122","pollo678","jugar123","NBC"
"04/07/2012 06:53am","67.246.161.66","Robuxer2910","Robuxer123","NBC"
```

any idea why it’s bad?

C crypto hash



```
INSERT INTO `wp_users` (`ID`, `user_login`, `user_pass`, `user_nicename`, `user_email`
isplay_name`, `spam`, `deleted`) VALUES
        (1, 'admin', '$P$BianZnllFtu/it9TizjVoK4ewP0zkp/', 'admin', 'emmanuel.elizondc
        ,(165, 'pwood', '$P$Bcq46LKqgXfHdfqSXVga0udGthUpZH0', 'pwood', 'peter.wood@amc
        ,(3, 'mbasford', '$P$BzMFsLd4P/cg9CVXU7VklMr51yIwgK.', 'mbasford', 'jenna.koze
)
        ,(4, 'pnolte', '$P$BGCpIw4MGM2Z57AXqo81ut3GidTa1b0', 'pnolte', 'paul.nolte@amc
        ,(5, 'stories', '$P$B55hfIa3EcBByVjElFUQg3nsLcZrJl/', 'stories', 'experiences@
', 0, 0)
        ,(6, 'gfrost', '$P$BTNqshBmLCQ.8QdoCy/BVUaF3zzGB3/', 'gfrost', 'gary.frost@amc
        ,(7, 'ecaspole', '$P$BcWcYB/q.J62DYnm7kFlQYo7MLr2Jk0', 'ecaspole', 'Eric.Caspc
        ,(8, 'amd developercentral', '$P$Bsyn.Ei2fxf01/3nXo3.7Dhli3sVDf.', 'amd-develc
ntral', 0, 0)
        ,(9, 'devperformanceteam', '$P$B/MM.rWaANvnQjRRwHRQ8kYUyu3mEi/', 'devperformar
L, 0, 'Dev PerformanceTeam', 0, 0)
        ,(10, 'john.mccrae', '$P$BOvhDW0dx/XdFwXCm.HbFGyY6CbR5t1', 'john-mccrae', 'joh
, 0)
        ,(11, 'vanderhe', '$P$BaSRNaG9A3NvhsS5k6UKWhQIx3xbGX1', 'vanderhe','Randy.Var
yden', 0, 0)
        ,(166, 'ddoel', '$P$B1I72YDpA/CS64tSsWMbEF9HXkVV4C.', 'ddoel', 'David.Doel@amc
        ,(13, 'peteroruba', '$P$BEQrXSYuYdjYBV2MSSq.wAxK2XSnb61', 'peteroruba', 'Peter
```

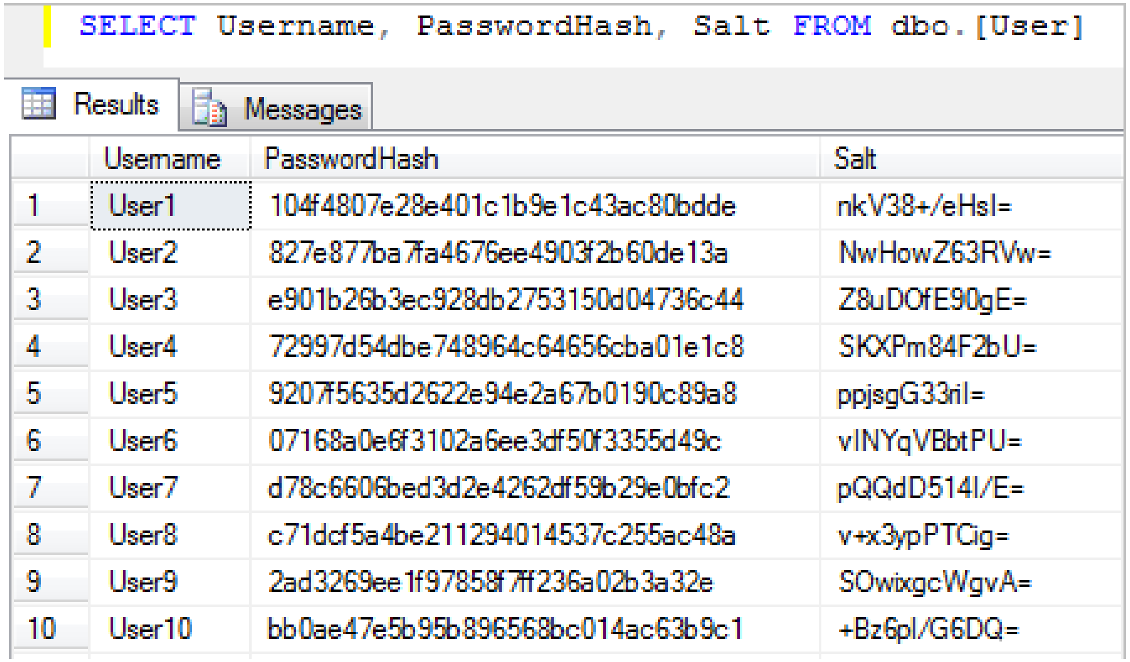
strongest passwords safe

“0%@-xWE3,2jmn__92jfoadfjA{WADS” etc.

but most passwords **vulnerable** to

time-memory trade-offs, dictionary attacks

B crypto hash *with a salt*



```sql
SELECT Username, PasswordHash, Salt FROM dbo.[User]
```

	Username	PasswordHash	Salt
1	User1	104f4807e28e401c1b9e1c43ac80bdde	nkV38+/eHsI=
2	User2	827e877ba7fa4676ee4903f2b60de13a	NwHowZ63RVw=
3	User3	e901b26b3ec928db2753150d04736c44	Z8uDOfE90gE=
4	User4	72997d54dbe748964c64656cba01e1c8	SKXPm84F2bU=
5	User5	9207f5635d2622e94e2a67b0190c89a8	ppjsgG33riI=
6	User6	07168a0e6f3102a6ee3df50f3355d49c	vINYqVBbtPU=
7	User7	d78c6606bed3d2e4262df59b29e0bfc2	pQQdD514I/E=
8	User8	c71dcf5a4be211294014537c255ac48a	v+x3ypPTCig=
9	User9	2ad3269ee1f97858f7ff236a02b3a32e	SOwixgcWgvA=
10	User10	bb0ae47e5b95b896568bc014ac63b9c1	+Bz6pI/G6DQ=

strongest passwords safe

“0%@-xWE3,2jmn__92jfoadfjA{WADS” etc.

but most passwords **vulnerable** to

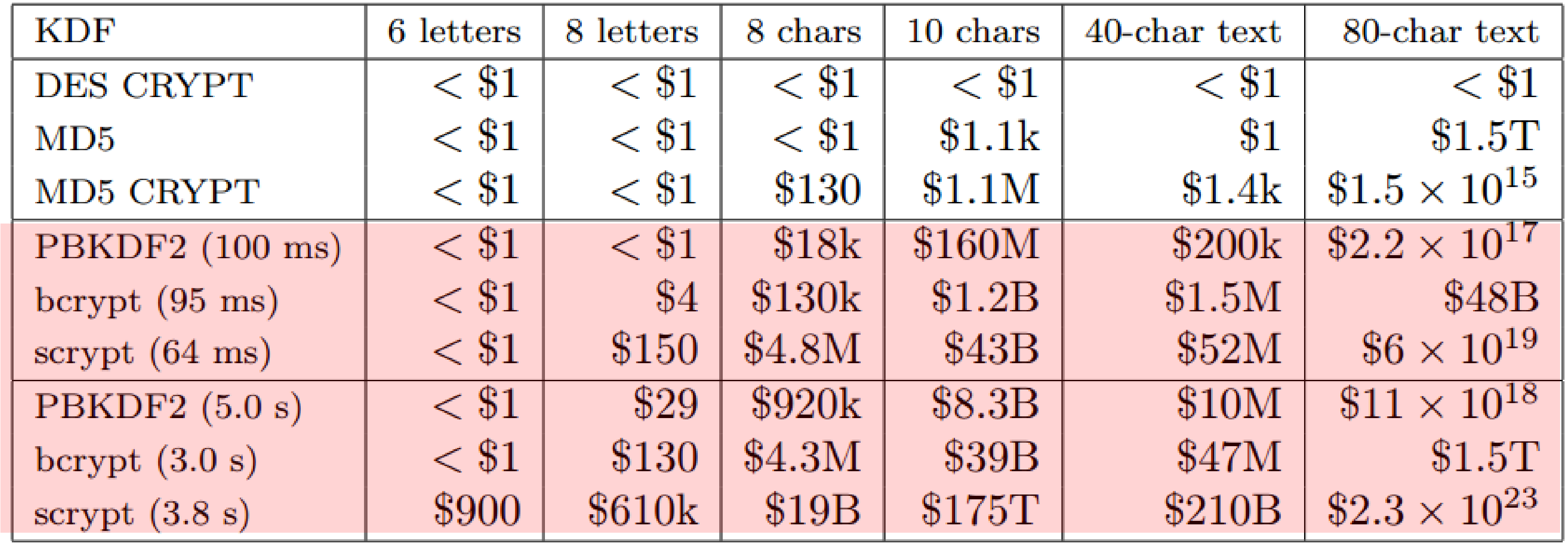
dictionary attacks

A password hash

much slower than crypto hashes

minimizes the advantage of GPU/FPGA

tweakable speed and/or memory req'ts



KDF	6 letters	8 letters	8 chars	10 chars	40-char text	80-char text
DES CRYPT	< \$1	< \$1	< \$1	< \$1	< \$1	< \$1
MD5	< \$1	< \$1	< \$1	\$1.1k	\$1	\$1.5T
MD5 CRYPT	< \$1	< \$1	\$130	\$1.1M	\$1.4k	$\$1.5 \times 10^{15}$
PBKDF2 (100 ms)	< \$1	< \$1	\$18k	\$160M	\$200k	$\$2.2 \times 10^{17}$
bcrypt (95 ms)	< \$1	\$4	\$130k	\$1.2B	\$1.5M	\$48B
scrypt (64 ms)	< \$1	\$150	\$4.8M	\$43B	\$52M	$\$6 \times 10^{19}$
PBKDF2 (5.0 s)	< \$1	\$29	\$920k	\$8.3B	\$10M	$\$11 \times 10^{18}$
bcrypt (3.0 s)	< \$1	\$130	\$4.3M	\$39B	\$47M	\$1.5T
scrypt (3.8 s)	\$900	\$610k	\$19B	\$175T	\$210B	$\$2.3 \times 10^{23}$

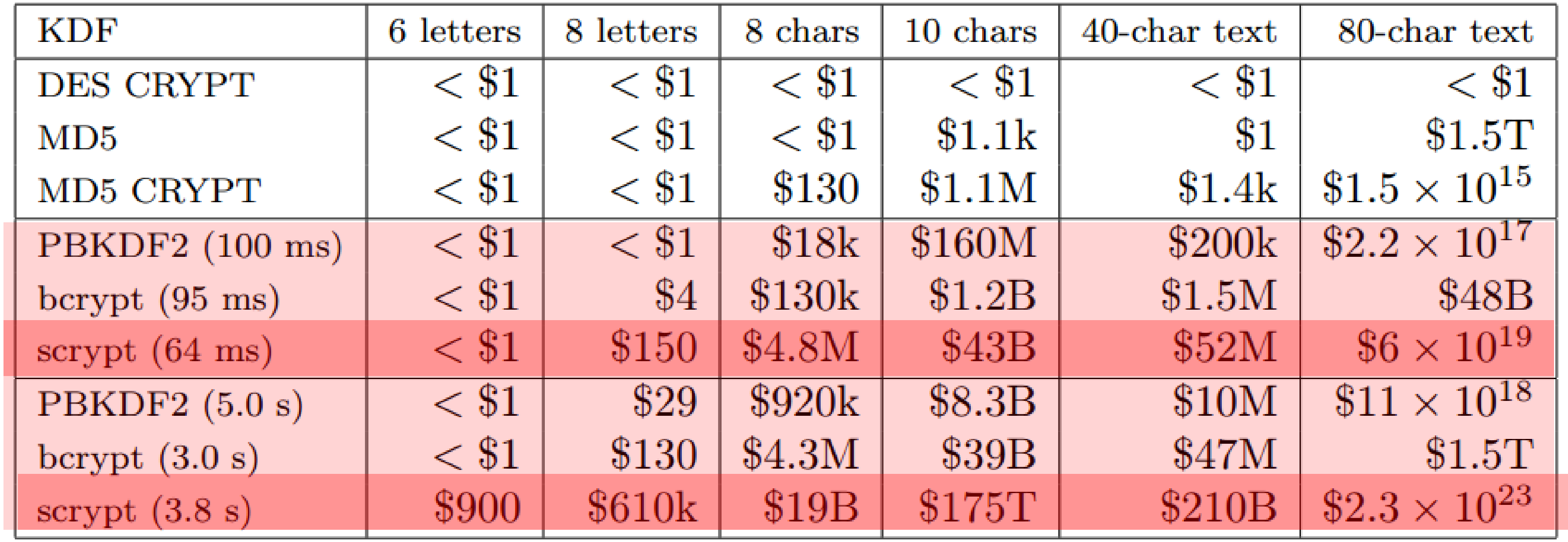
2009 estimates in http://www.tarsnap.com/scrypt/scrypt.pdf

A password hash

much slower than crypto hashes

minimizes the advantage of GPU/FPGA

tweakable speed and/or memory req'ts



KDF	6 letters	8 letters	8 chars	10 chars	40-char text	80-char text
DES CRYPT	$< \$1$	$< \$1$	$< \$1$	$< \$1$	$< \$1$	$< \$1$
MD5	$< \$1$	$< \$1$	$< \$1$	\$1.1k	\$1	\$1.5T
MD5 CRYPT	$< \$1$	$< \$1$	\$130	\$1.1M	\$1.4k	$\$1.5 \times 10^{15}$
PBKDF2 (100 ms)	$< \$1$	$< \$1$	\$18k	\$160M	\$200k	$\$2.2 \times 10^{17}$
bcrypt (95 ms)	$< \$1$	\$4	\$130k	\$1.2B	\$1.5M	\$48B
scrypt (64 ms)	$< \$1$	\$150	\$4.8M	\$43B	\$52M	$\$6 \times 10^{19}$
PBKDF2 (5.0 s)	$< \$1$	\$29	\$920k	\$8.3B	\$10M	$\$11 \times 10^{18}$
bcrypt (3.0 s)	$< \$1$	\$130	\$4.3M	\$39B	\$47M	\$1.5T
scrypt (3.8 s)	\$900	\$610k	\$19B	\$175T	\$210B	$\$2.3 \times 10^{23}$

2009 estimates in http://www.tarsnap.com/scrypt/scrypt.pdf

scrypt (Percival, 2009)

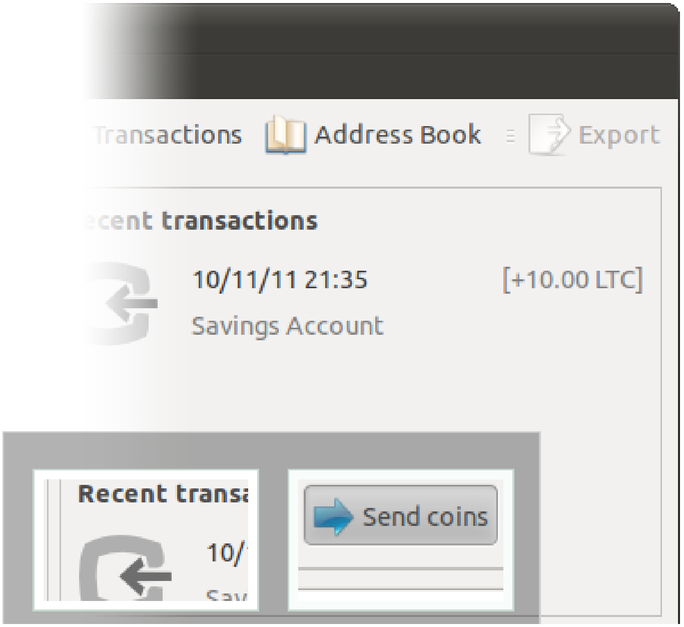
notion of ‘memory-hard’ functions

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

“Litecoin uses scrypt as a proof-of-work scheme. Scrypt adds memory-intensive algorithms to reduce the efficiency of the kind of parallelization that GPUs offered in early Bitcoin mining.”

litecoin



10/11/11 21:35 [+10.00 LTC]
Savings Account

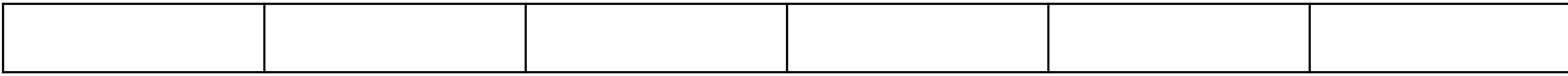
10/
Sav

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

V[i] = H(V[i-1]), i=0..N-1



scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

V[i] = H(V[i-1]), i=0..N-1

b83546b4					

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(V[i-1]), i=0..N-1$

b83546b4	b2e2a2f5				

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

b83546b4	b2e2a2f5	10cbd82a			

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

V[i] = H(V[i-1]), i=0..N-1

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(V[i-1]), i=0..N-1$

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

sequential unpredictable accesses

$X = H(\ X \oplus V[\ X \bmod N\ ]\ ),\ i=0..N-1$

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

b83546b4	b2e2a2f5	**10cbd82a**	...	57500361	299c689f

sequential unpredictable accesses

$X = H(\ X \oplus V[\ X \bmod N\ ]\ ),\ i=0..N-1$

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

`b83546b4`	`b2e2a2f5`	`10cbd82a`	...	`57500361`	**`299c689f`**

sequential unpredictable accesses

$X = H(\ X \oplus V[\ X \bmod N\ ]\ ),\ i=0..N-1$

scrypt (Percival, 2009)

notion of 'memory-hard' functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

sequential unpredictable accesses

$X = H(\ X \oplus V[\ X \bmod N\ ]\ ),\ i=0..N-1$

scrypt (Percival, 2009)

notion of 'memory-hard' functions

sequential initialization of a large array V

$V[i] = H(\ V[i-1]\ ),\ i=0..N-1$

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

sequential unpredictable accesses

$X = H(\ X \oplus V[\ X \bmod N\ ]\ ),\ i=0..N-1$

scrypt (Percival, 2009)

notion of ‘memory-hard’ functions

bounds on the time-memory tradeoffs

password hashing scheme using

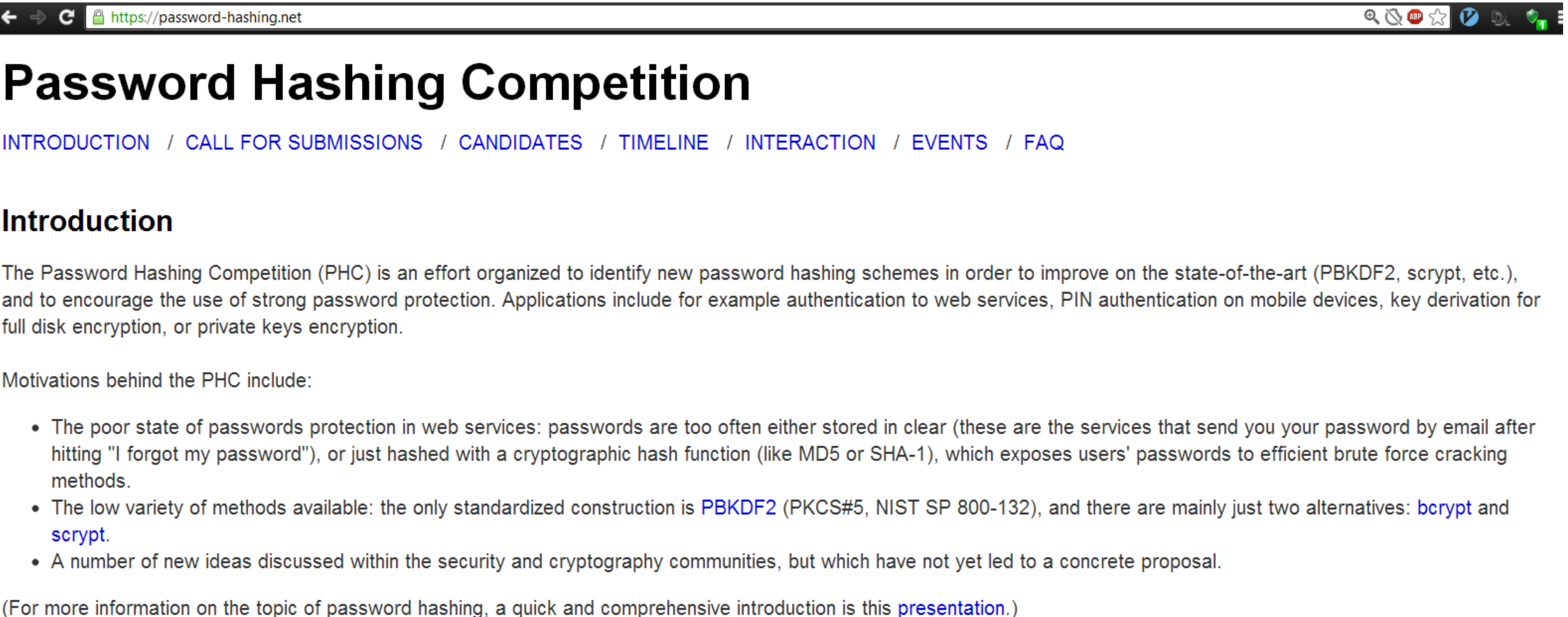
- PBKDF2-HMAC-SHA-256
- Salsa20

not much used in practice…

(Tarsnap, Chromium’s key vault, etc.)

PHC goals

- raise awareness
- encourage the use of strong password hashes
- propose new designs suiting applications' needs (and std'ze them?)


https://password-hashing.net
INTRODUCTION / CALL FOR SUBMISSIONS / CANDIDATES / TIMELINE / INTERACTION / EVENTS / FAQ

Password Hashing Competition

Introduction

The Password Hashing Competition (PHC) is an effort organized to identify new password hashing schemes in order to improve on the state-of-the-art (PBKDF2, scrypt, etc.), and to encourage the use of strong password protection. Applications include for example authentication to web services, PIN authentication on mobile devices, key derivation for full disk encryption, or private keys encryption.

Motivations behind the PHC include:

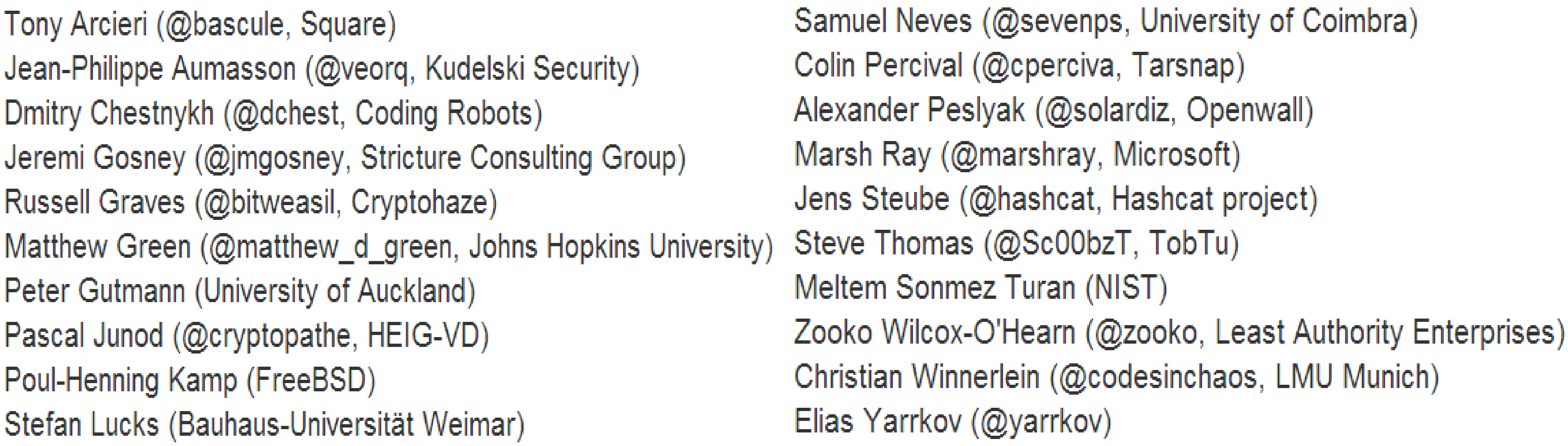
- The poor state of passwords protection in web services: passwords are too often either stored in clear (these are the services that send you your password by email after hitting "I forgot my password"), or just hashed with a cryptographic hash function (like MD5 or SHA-1), which exposes users' passwords to efficient brute force cracking methods.
- The low variety of methods available: the only standardized construction is PBKDF2 (PKCS#5, NIST SP 800-132), and there are mainly just two alternatives: bcrypt and scrypt.
- A number of new ideas discussed within the security and cryptography communities, but which have not yet led to a concrete proposal.

(For more information on the topic of password hashing, a quick and comprehensive introduction is this presentation.)

PHC organization

panel of experts

- industry, academia, gov
- designers, crackers, coders



Tony Arcieri (@bascule, Square)
Jean-Philippe Aumasson (@veorq, Kudelski Security)
Dmitry Chestnykh (@dchest, Coding Robots)
Jeremi Gosney (@jmgosney, Stricture Consulting Group)
Russell Graves (@bitweasil, Cryptohaze)
Matthew Green (@matthew_d_green, Johns Hopkins University)
Peter Gutmann (University of Auckland)
Pascal Junod (@cryptopathe, HEIG-VD)
Poul-Henning Kamp (FreeBSD)
Stefan Lucks (Bauhaus-Universität Weimar)
Samuel Neves (@sevenps, University of Coimbra)
Colin Percival (@cperciva, Tarsnap)
Alexander Peslyak (@solardiz, Openwall)
Marsh Ray (@marshray, Microsoft)
Jens Steube (@hashcat, Hashcat project)
Steve Thomas (@Sc00bzT, TobTu)
Meltem Sonmez Turan (NIST)
Zooko Wilcox-O'Hearn (@zooko, Least Authority Enterprises)
Christian Winnerlein (@codesinchaos, LMU Munich)
Elias Yarrkov (@yarrkov)

PHC call for submissions

minimal I/O requirements

- 0..128 *bytes* **password**
- 16-byte **salt**
- 1 or more **cost parameters**
- returns a 16-byte **hash**

PHC call for submissions

security and functionality

- ‘random’ behavior
- minimal speed-up from cracking-optimized SW or HW
- resilience to side-channel attacks
- effectiveness of the cost parameters
- flexibility and scalability

PHC call for submissions

simplicity

- criterion often overlooked…
- **specs**: clarity, conciseness, number of components, prior knowledge, etc.
- **implementation**: mapping from spec, support for existing instructions, etc.

“complexity provides both opportunity and hiding places for attackers“ --Dan Geer

Design of a PHS (PH Scheme)

- **application**? (key derivation, storage?)
- **platform**? (64-bit SW, mobile, low-end?)
- degree of platform-**specific vs generic**
- support for **arbitrary-length** passwords?

  if Y, timing leaks difficult to prevent…
- **length**: do we need more than 16 bytes?
- how to implement “**memory hardness**”?

  reads vs. writes; blocks size; predictability and order; etc.

  prove rigorous bounds on time-memory-tradeoff?

  RAM vs ROM storage requirements
- what degree/type of **parallelism**?

Uncertainty factors

- How will **technology** evolve?

  PHS engineering depends highly on technology…

  SSDs vs. mechanical drives latency

  CPU’s cache size/latency in 20 years?

  GPUs/FPGAs features and pricing in 20 years?

- How will **attackers/crackers** evolve?

  which platforms will be the most cost-effective?

  product- vs service-based solutions (cloud, etc.)

  ever increasing rate of “passwords dumps”?

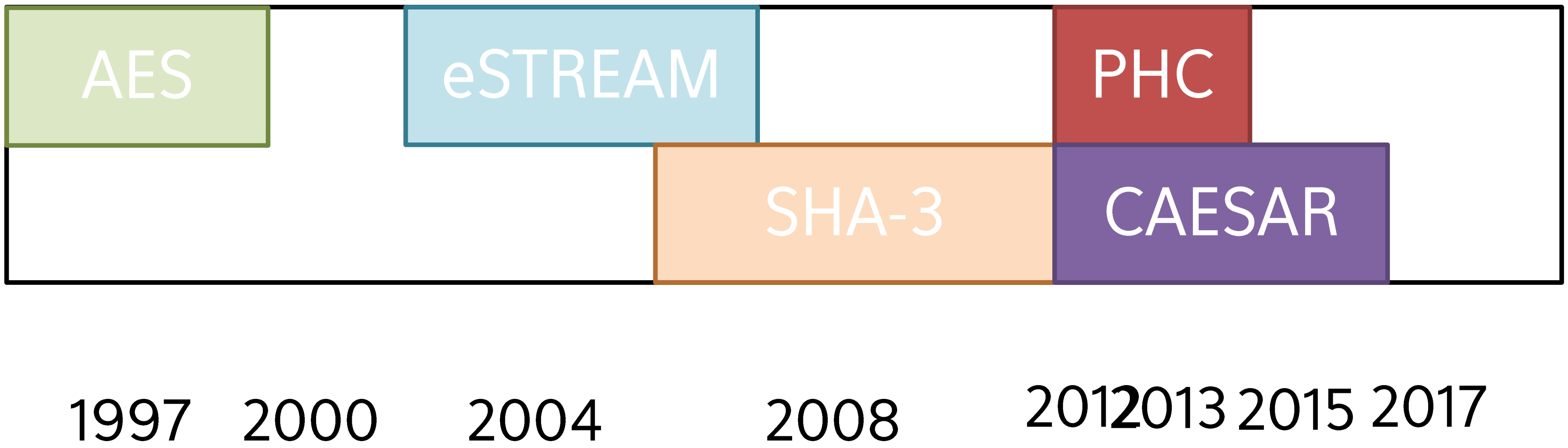
- How will **defenders** evolve?

  sufficient incentives to move to strong PHS?

  issues of servers load and DoS protection

What is certain:

- CAESAR and PHC are starting
- engineering/scientific innovation ahead
- and ***lots of fun***!



Dates and events

CAESAR

- 2013 summer: DIAC workshop (Chicago)
- 2014 Jan 15: **submission** deadline
- 2014 summer: DIAC workshop (?)

PHC

- 2013 Jul 30-31: PasswordsCon (Las Vegas)
- 2013 Dec: PasswordsCon (Norway)
- 2014 Jan 31: **submission** deadline

thank you!