KUDELSKI SECURITY

Password Hashing: the Future is Now

Jean-Philippe Aumasson, Principal Cryptographer

March 2013

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

Naked Security | Follow | Reblog

Evernote hacked - almost 50 million passwords reset after security breach

Join thousands of others, and sign up for Naked Security's newsletter

you@example.com Do it!

Don't show me this again X

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

KUDELSKI SECURITY

April 2013

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

MAIN MENU ▾ MY STORIES: 0 ▾ FORUMS SUBSCRIBE VIDEO

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

HACKING INTERNET CRIME 138

KUDELSKI SECURITY

May 2013

news.softpedia.com/news/Reputation-com-Hacked-All-User-Passwords-Reset-350034.shtml

Home > News > Security > Hacking News

May 1st, 2013, 11:08 GMT · By Eduard Kovacs

Reputation.com Hacked, All User Passwords Reset

SHARE: +1 3 Tweet Adjust text size:

reputation.com

ENLARGE

Online reputation and review management firm Reputation.com has suffered a security breach. The company has started notifying customers, informing them that their passwords have been reset.

Reputation.com representatives have stated that the attack was "interrupted and swiftly shut down" before the attackers could complete it.

"Following the attack, our engineering and security team immediately conducted an exhaustive investigation, working closely with independent security experts to determine what information may have been accessed," reads the notification sent to customers (via Dave Lucas).

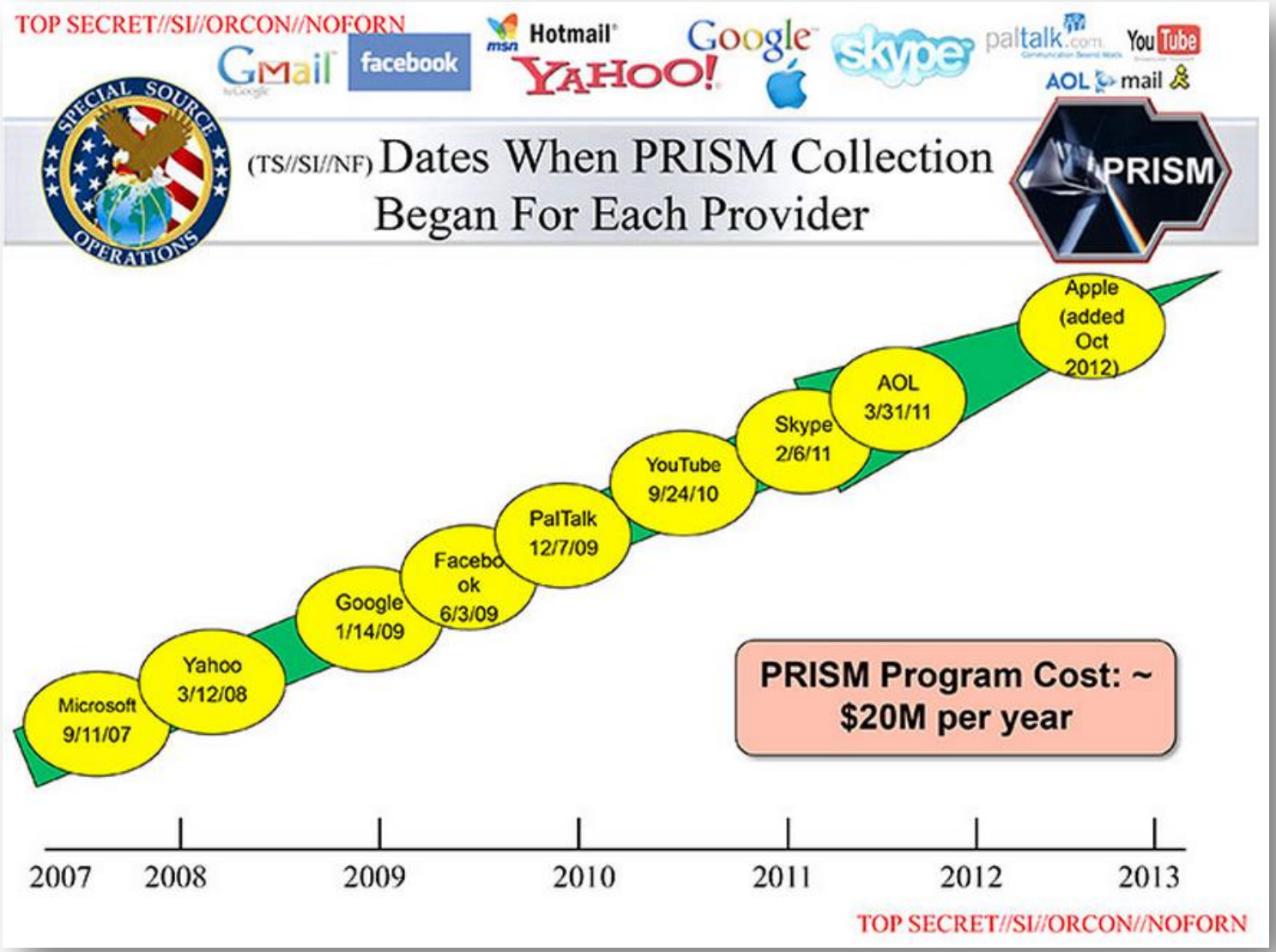
"We are also implementing additional security measures, beyond the high level of security that is already in place, to ensure your continued protection."

According to the firm, they're confident that the cybercriminals haven't been able to access financial information – which is said to be stored on third-party systems –, account details, communication between the user and the Reputation.com team, and information about the provided services.

On the other hand, the attackers have accessed names, email addresses, physical addresses and, in some cases, dates of birth, phone numbers and occupational information.

KUDELSKI SECURITY

June 2013



KUDELSKI SECURITY

July 2, 2013

news.cnet.com/8301-1009_3-57592088-83/ubisoft-hacked-users-e-mails-and-passwords-exposed/

c|net

Reviews | News | Download | CNET TV | How To | Deals

CNET › News › Security & Privacy › Ubisoft hacked; users' e-mails and passwords exposed

Ubisoft hacked; users' e-mails and passwords exposed

The video game developer, known for creating Assassin's Creed, announces that its account database was breached and that all users should to reset their passwords.

by Dara Kerr | July 2, 2013 7:50 PM PDT

Follow @darakerr

KUDELSKI SECURITY

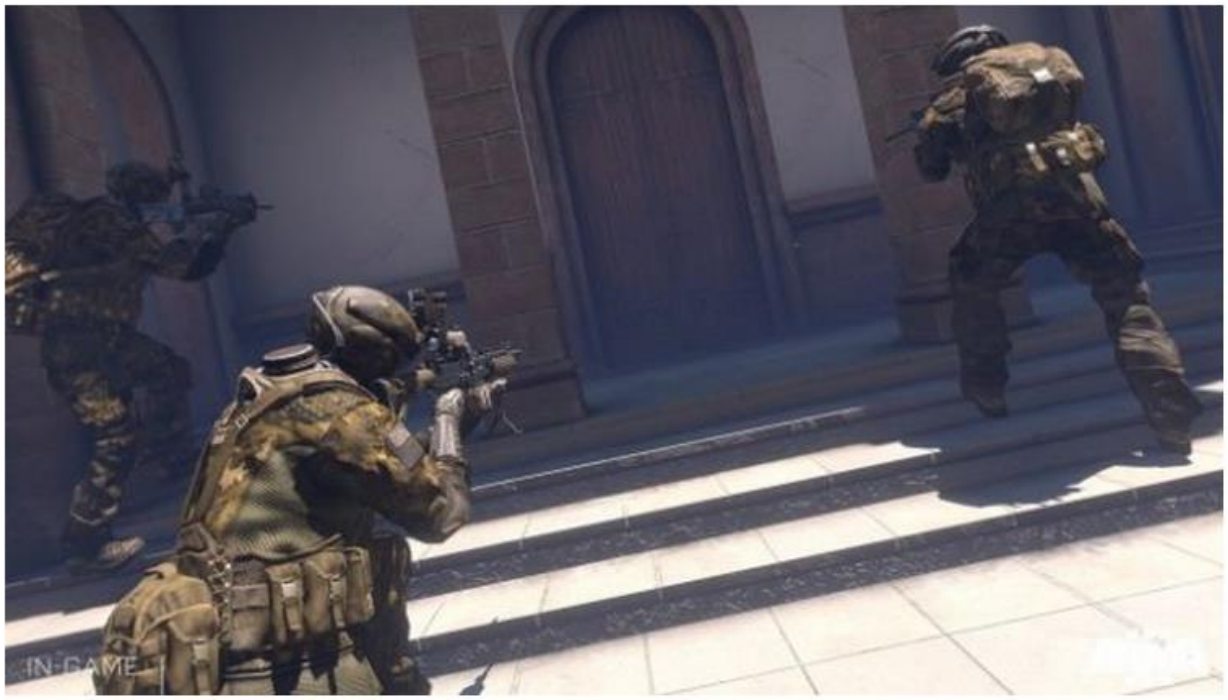
July 13, 2013

www.ign.com/blogs/retrocortana101/2013/07/13/bohemia-interactive-hacked-usernames-emails-and-encrypted-passwords-taken/

IGN Comic-Con 2013 Prime Sign

Bohemia Interactive hacked – usernames, emails and encrypted passwords taken

July 13, 2013 by RetroCortana101



KUDELSKI SECURITY

July 18, 2013

grahamcluley.com/2013/07/nasdaq-hackers/

Hackers hit the NASDAQ community forum, email addresses and passwords compromised

Graham Cluley | July 18, 2013 8:45 am | Filed under: **Privacy**, **Vulnerability** | 2

If you're new here, you may want to subscribe to the RSS feed, like us on Facebook, or sign-up for the free email newsletter which contains computer security advice, news, hints and tips. Thanks for visiting!

There is bad news if you are in the habit of discussing stocks on the NASDAQ community forum, because hackers have managed to break into the site, and could have compromised usernames, email addresses and passwords.

The only silver lining on the cloud is that trading and commerce platforms were not impatced by the hack.

Users of NASDAQ's community messageboards should have received an email from the site, warning users about the security breach and advising members to change their passwords on *other* websites if the same password was being used.

KUDELSKI SECURITY

July 21, 2013

grahamcluley.com/2013/07/ubuntu-forums-hack/

Ubuntu Forums hacked, 1.8 million passwords and emails stolen

Graham Cluley | July 21, 2013 2:32 pm | Filed under: **Data loss**, **Linux**, **Privacy**, **Vulnerability** | 1

There has been a massive data breach impacting over 1.8 million users of the Ubuntu operating system this weekend.

Canonical, the lead developers of the Ubuntu Linux-based operating system, has admitted that its online forums were not just defaced this weekend, but also that hackers managed to steal every users' email address, password and username from the Ubuntu Forums database.

The first clue that anything was amiss was when hackers posted a (hard-to-miss) message on the Ubuntu Forums homepage of a penguin holding a sniper's rifle:

KUDELSKI SECURITY

We have a problem

Protecting passwords

~~Tell users with weak passwords they are stupid and deserve to be hacked~~

Strengthen the server, to prevent a compromise

- Fails in practice

Strengthen the hashes, to mitigate a compromise

- Fails in practice (so far)
- Probably easier to fix
- **The point of this talk**

How (not) to hash

No hash (1960's)

return password

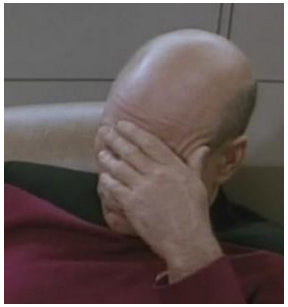

```
$result = mysql_query(
  "SELECT * FROM users " .
  " WHERE SHA1(username) = SHA1('" . $_REQUEST["username"] . "') " .
  "   AND SHA1(password) = SHA1('" . $_REQUEST["password"] . "')");
```

http://thedailywtf.com/Articles/Topgrade,-SHA1-Encryption.aspx

Crypto hash (early 1970's)

```
return hash( password )
```

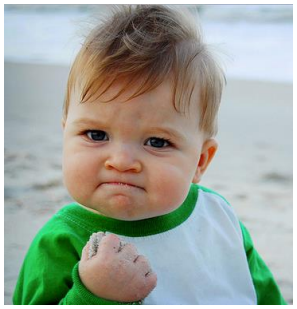
- One-way (cannot be efficiently inverted)



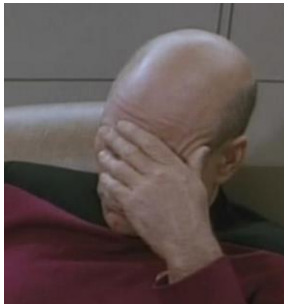
- Efficient dictionary & bruteforce attacks
- Vulnerable to rainbow tables

Crypto hash with a salt (late 1970's)

```
return hash( password, salt )
```



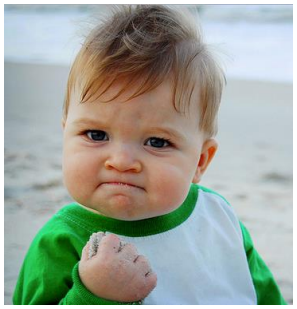
- One-way (cannot be efficiently inverted)
- Not vulnerable to rainbow tables



- Efficient dictionary & bruteforce attacks

Password hashing scheme (2000's)

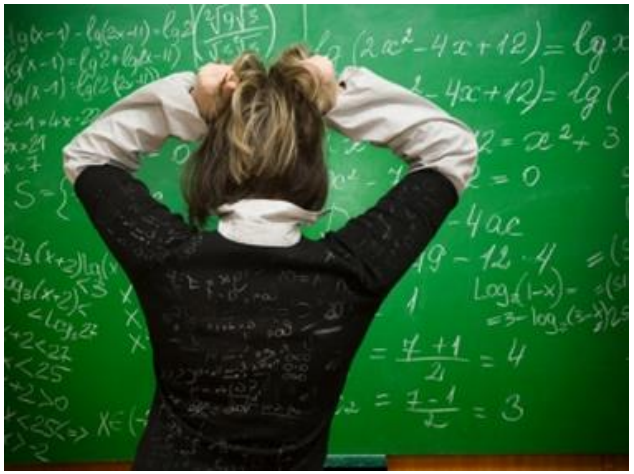
```
return hash( password, salt, cost )
```



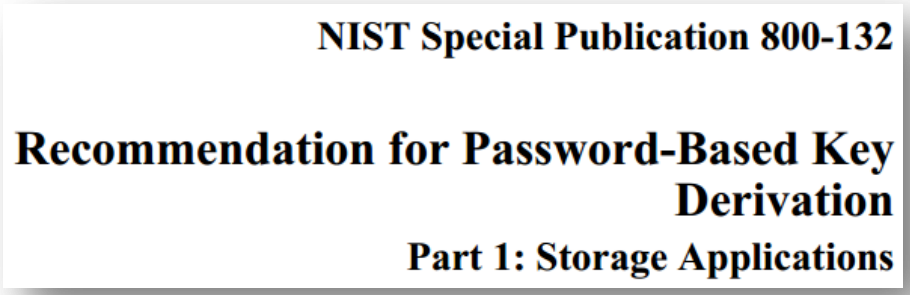
- One-way (cannot be efficiently inverted)
- Not vulnerable to rainbow tables
- Inefficient dictionary & bruteforce attacks
- Minimizes the advantage of GPU/FPGA

Forces attackers to use much more resources to test a password

- Arithmetic operations
- Memory usage and read/writes



PBKDF2 (Kaliski, 2000)


NIST Special Publication 800-132

Recommendation for Password-Based Key Derivation

Part 1: Storage Applications

TRUECRYPT

iOS

GnuPG

Initially designed for **key derivation**

Iteration of a pseudorandom function (MAC)

- Typically HMAC-SHA-1×10000

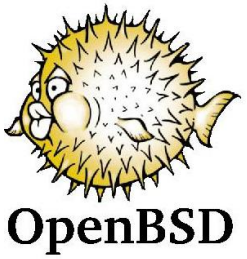
No attempt to reduce the efficiency of attackers

KUDELSKI SECURITY

bcrypt (Provos/Mazières, 1999)

A Future-Adaptable Password Scheme

Niels Provos and David Mazières
{provos,dm}@openbsd.org
The OpenBSD Project



livingsocial

"4KB of constantly accessed and modified memory"

Cost parameter affects only hashing time, not memory

- 4KB manageable with today's FPGAs and GPUs

Not parallelizable

- Defenders can't exploit SIMD or multicores

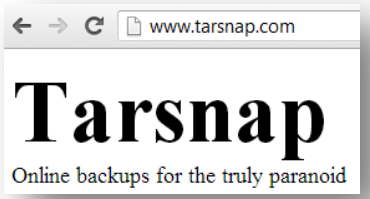
scrypt (Percival, 2009)

Introduced the notion of **memory-hard** function

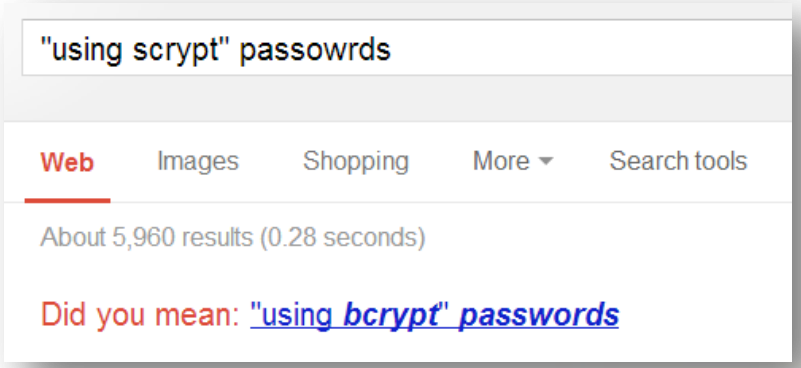
Definition

A *memory-hard* algorithm on a Random Access Machine is an algorithm which uses $S(n)$ space and $T(n)$ operations, where $S(n) \in \Omega\left(T(n)^{1-\epsilon}\right)$.

Less popular than bcrypt...



litecoin



KUDELSKI SECURITY

scrypt (Percival, 2009)

scrypt (Percival, 2009)

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

b83546b4	**b2e2a2f5**				

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

b83546b4	b2e2a2f5	**10cbd82a**			

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

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

1) Sequential initialization of a large array V

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

scrypt (Percival, 2009)

1) Sequential initialization of a large array V

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

2) Sequential unpredictable accesses

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

scrypt (Percival, 2009)

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

1) Sequential initialization of a large array V

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

2) Sequential unpredictable accesses

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

scrypt (Percival, 2009)

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

1) Sequential initialization of a large array V

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

2) Sequential unpredictable accesses

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

scrypt (Percival, 2009)

b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f

1) Sequential initialization of a large array V

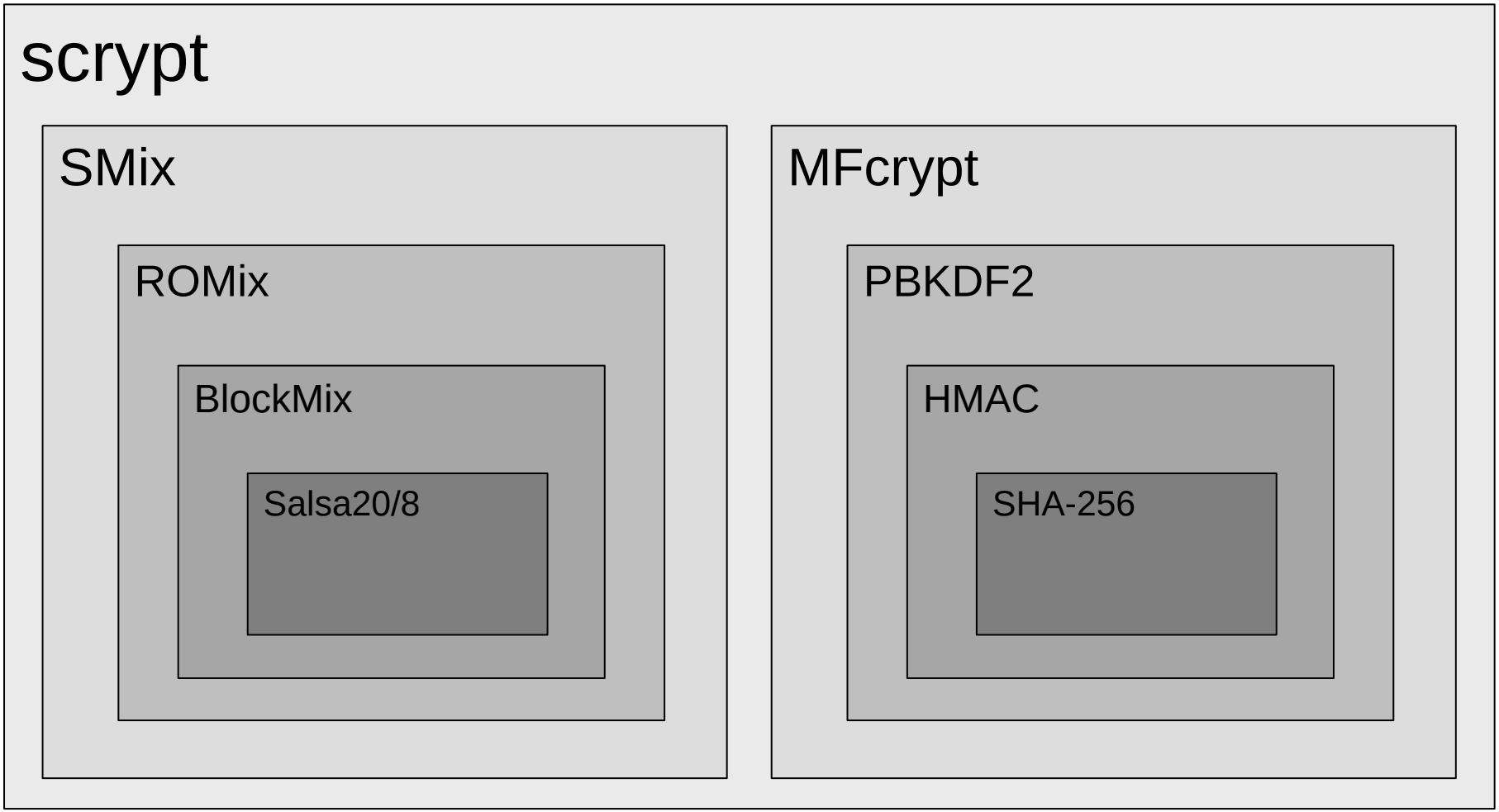
$$V[i] = H(V[i-1]),\ i=0..N-1$$

2) Sequential unpredictable accesses

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

scrypt limitations: simplicity

4-level nested structure with 4 distinct crypto schemes



Is scrypt user-friendly?

3 parameters:

N: "Integer work metric"

r: "Block size parameter"

p: "Parallelization parameter" (**r** also affects parallelism)

Which parameters should one choose?

Some recommendations in the 2009 paper, but different applications have different requirements

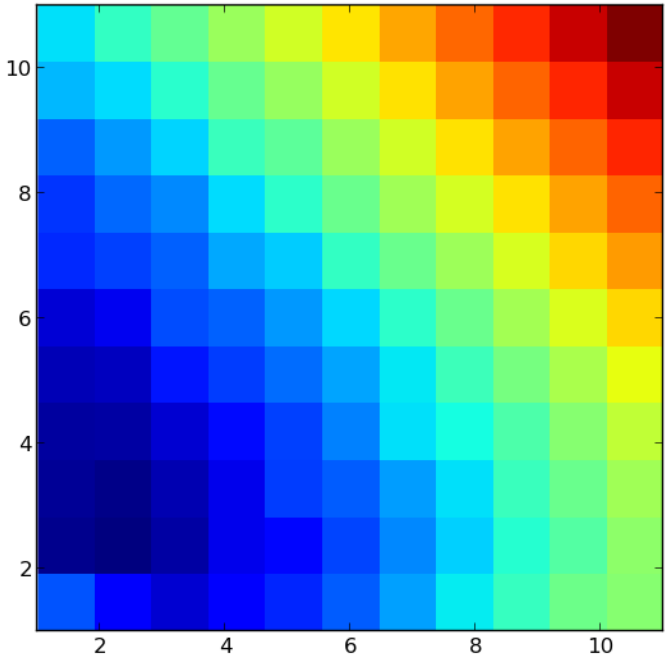
How are these *affecting scrypt performance*?

Is scrypt user-friendly?

N and **r** have a *similar effect* for the defender:

- **N** × **r** basic operations
- **N** × **r** × 128 bytes of memory

log(time) of scrypt with
X=log(**N**)
Y=log(**r**)
color range ~ 0.1 to 2000ms

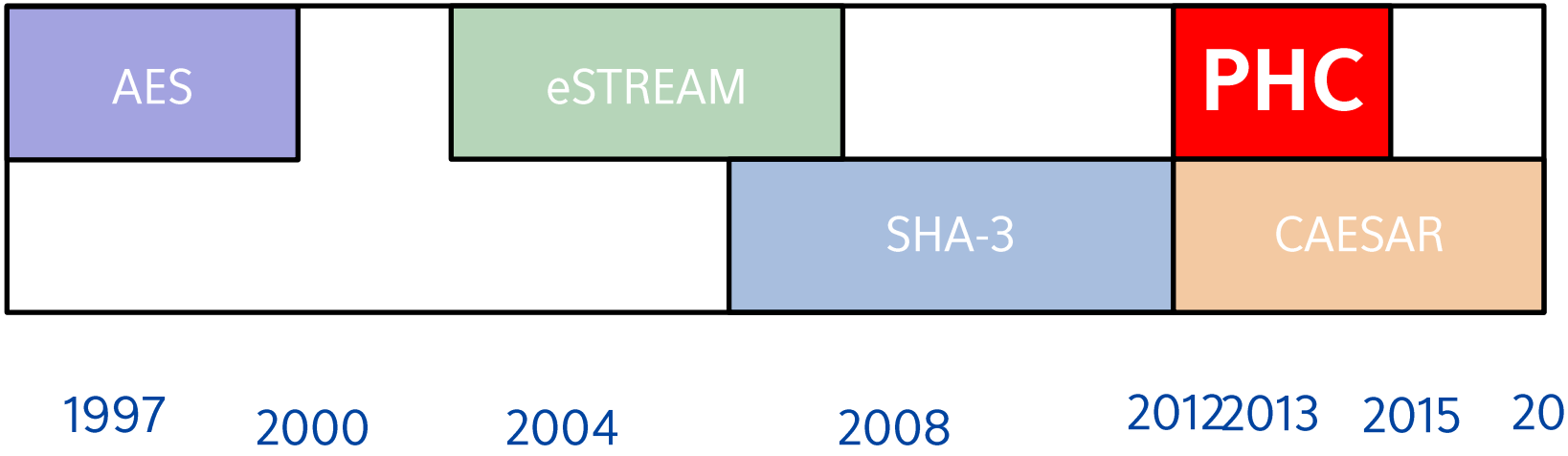


Impossible to increase only time and not memory

- Could be a problem for low-memory devices
- Doable indirectly by increasing parallelism (**p**)

Demo

The Password Hashing Competition (PHC)

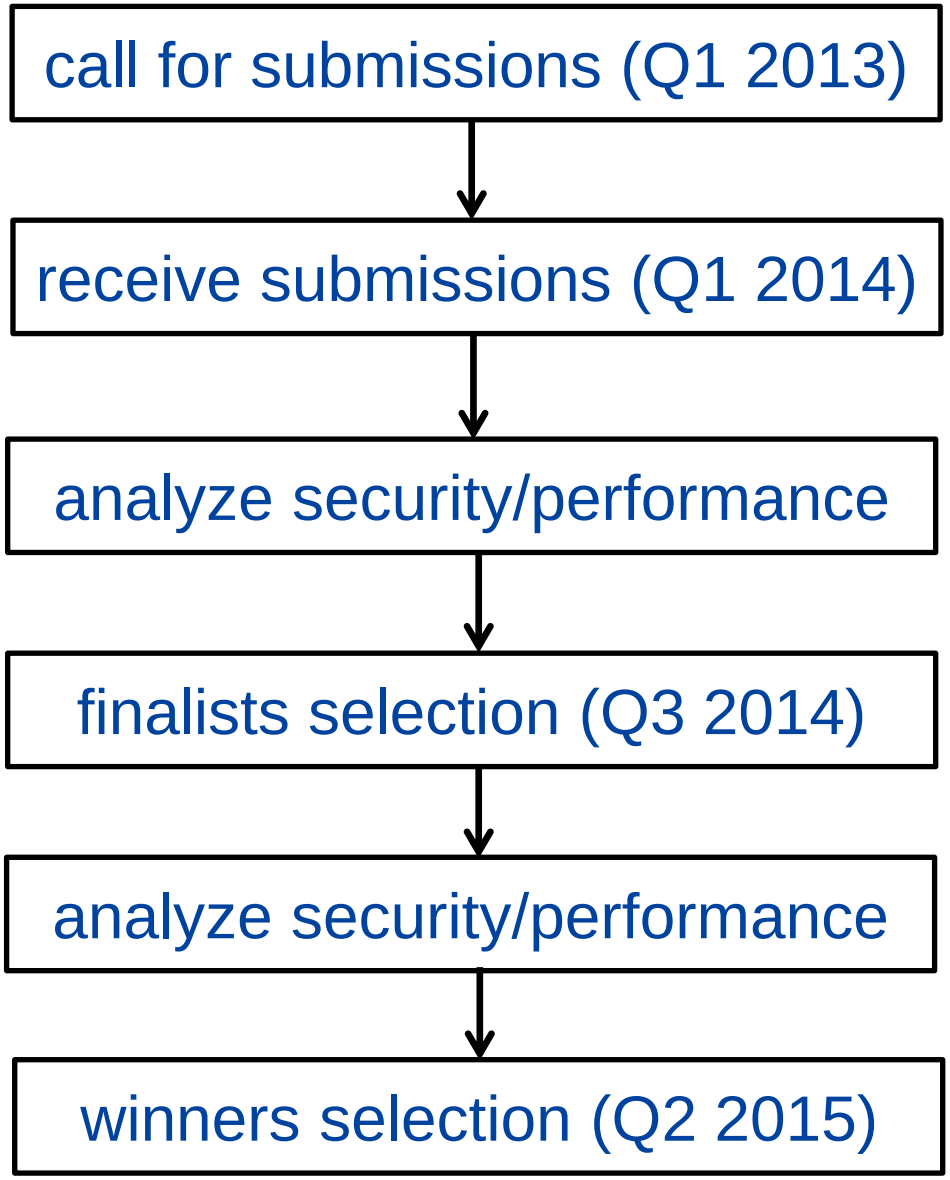


Experts panel: cryptographers, crackers, software engineers

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)

KUDELSKI SECURITY

Expected timeline



Analysis phase in crypto competitions

PHC submission requirements

Specification

Comprehensive definition, no-backdoor statement, etc.

Security analysis

Security claims and arguments, proofs, etc.

Efficiency analysis

Expected speed on amd64, suitability for GPU/FPGA, etc.

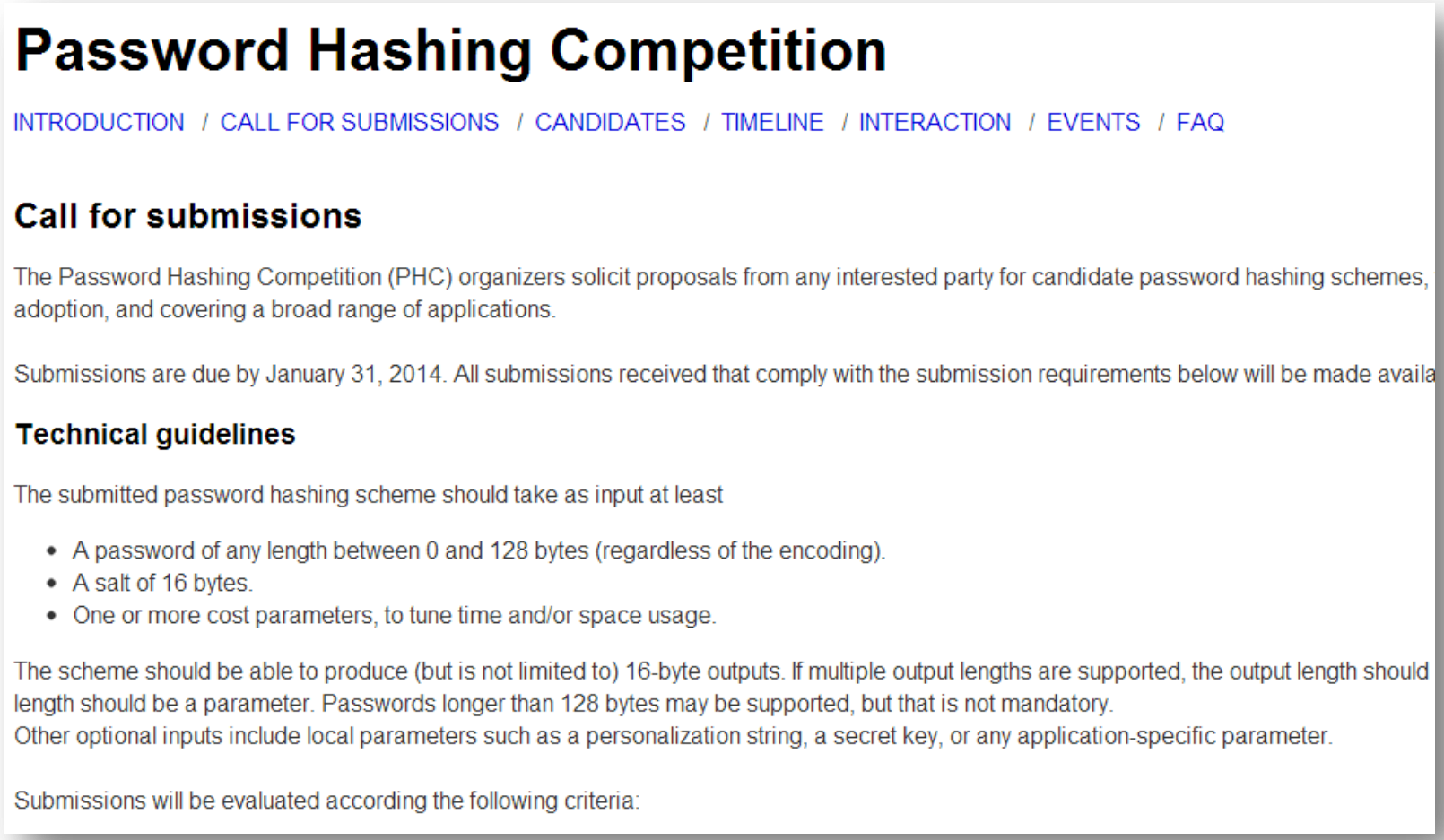
Code

Portable C(++), test vectors, etc.

Intellectual property statement

Must be available worldwide royalty-free and patent-less

Submit before January 31, 2014



Password Hashing Competition

INTRODUCTION / CALL FOR SUBMISSIONS / CANDIDATES / TIMELINE / INTERACTION / EVENTS / FAQ

Call for submissions

The Password Hashing Competition (PHC) organizers solicit proposals from any interested party for candidate password hashing schemes, adoption, and covering a broad range of applications.

Submissions are due by January 31, 2014. All submissions received that comply with the submission requirements below will be made availa

Technical guidelines

The submitted password hashing scheme should take as input at least

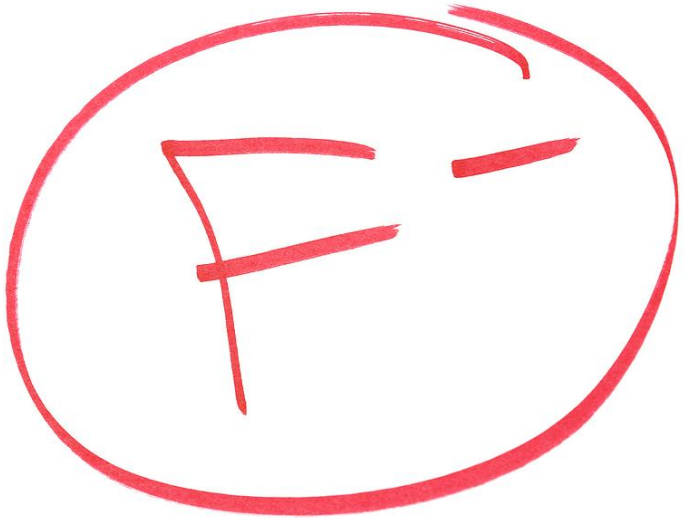
- A password of any length between 0 and 128 bytes (regardless of the encoding).
- A salt of 16 bytes.
- One or more cost parameters, to tune time and/or space usage.

The scheme should be able to produce (but is not limited to) 16-byte outputs. If multiple output lengths are supported, the output length should length should be a parameter. Passwords longer than 128 bytes may be supported, but that is not mandatory.
Other optional inputs include local parameters such as a personalization string, a secret key, or any application-specific parameter.

Submissions will be evaluated according the following criteria:

All details on https://password-hashing.net

Crypto research



Todos:

- Design plug-and-play constructions (like HMAC for MACs)
- Prove bounds on time/memory usage
- Cryptanalyze PHC candidates

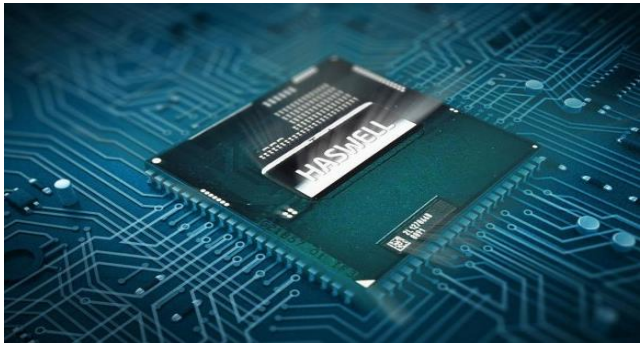
Optimization and technology-dependency

Password hashing is **technology-dependent**

- How will server chips look like in 10 years?
- What will be the most cost-efficient cracking hardware?

For example, hashes could be optimized for Intel CPUs:

- AVX2's 256-bit registers
- SIMD arithmetic and shuffles
- Gather instructions (parallel LUTs)
- Future 512-bit registers



The more optimized for 1 platform, the less consistent performance across CPU (micro)architectures...

Leakage resilience

Protection against the extraction of information from the **physical implementation** of a hashing scheme

Pure timing

If passwords of any length are supported, etc.

Cache timing

Password-dependent lookups in large tables, etc.

Memory leaks

Is it necessary to securely wipe the memory? etc.

Client-side hashing?

Should hashing be performed by the **clients**?

- For which application?
- Share effort between server and client?
- How to deal with diversity of client CPUs?
- Does it make sense to optimize a hash for JavaScript?

Addresses the **risk of DoS** on servers

- Like TLS-based DoS' initiate many connection attempts
- Probably not an issue for Google, but maybe for you
- May be addressed with standard anti-DoS defenses

Updatability

How to update the hashes to a different security level?

- **Without requiring a fresh login**
- Use fast hash as a proxy and store weak hashes offline?

Motivations: adapt to new technology and research

- Defenders (server CPU, cores available, etc.)
- Attackers (hardware, techniques, etc.)

More ideas

Programmable hashes

- Algorithm = F(password)
- Defeats custom hardware
- ≈ Code generator for a custom VM
- Consistency? Interoperability?

Security through obesity (J. Spilman)

- Pollute the DB with dummy hashes
- Hide usernames from the DB
- Huge DB (e.g. 1TB) complicates download

Password Hashing: the Future is Now

Thank you!

@veorq

https://131002.net

https://www.kudelskisecurity.com

KUDELSKI SECURITY