

NORX+PHC

JP Aumasson — @veorq
Dagstuhl 2014

NORX

NORX

our **CAESAR** submission, with

Philipp Jovanovic
(@Daeinar, Uni Passau, DE)

Samuel Neves
(@sevenps, Uni Coimbra, PT)

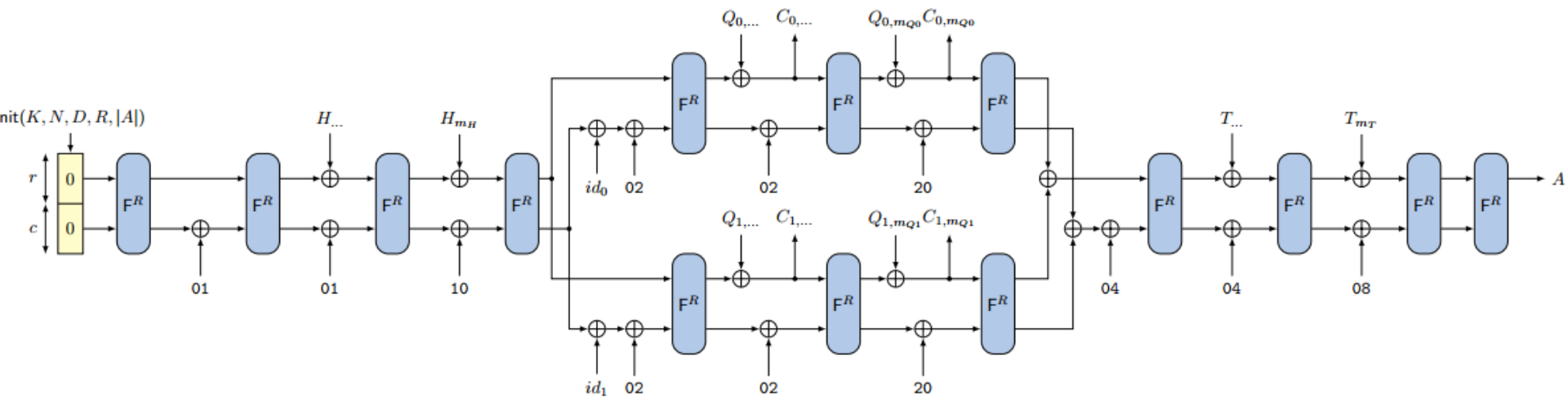
NO(T A)RX

$$s / a + b / a^b (a \& b) \ll 1$$

ADD approximation

Simpler differential analysis/proofs

Hardware efficiency++



monkeyDuplex-like mode
Tunable parallelism

Dedicated datagram

Avoids users the trouble of inventing their own encoding (append/prepend the nonce? etc.)

Optional signalling of parameters

Lengths part of the *header*

Tag added as *trailer*

(as in many data link protocols)

-> Interoperability++

AVX, AVX2, NEON code
~2.5/#cores cpb on Haswell

Ref code not much slower
-> Cheaper dev/testing/deployment

All code under public domain-like licence (CC0)

Collaboration with HW designers

Thorough efficiency evaluation

Led to a couple of optimizations

HDL code will be released

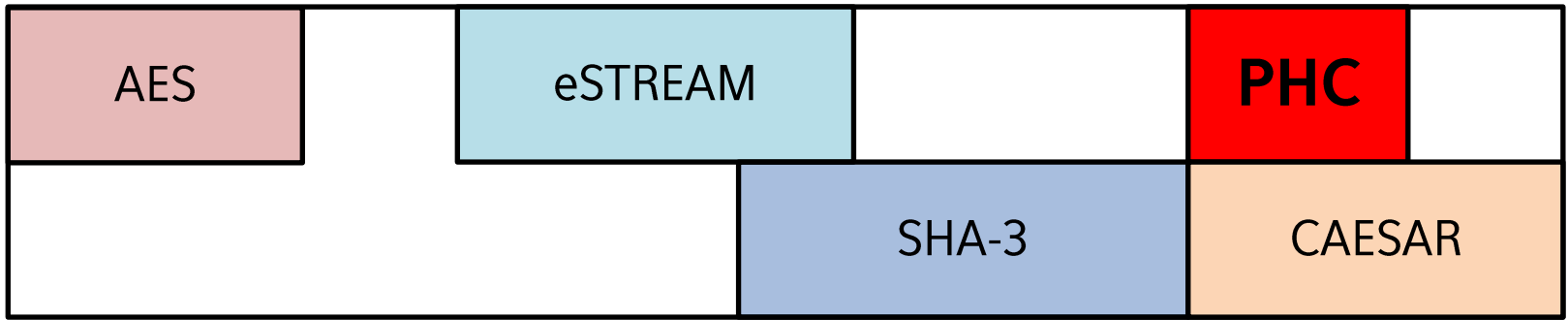
<https://norx.io>

PHC

The Password Hashing Competition

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

Another crypto competition



1997

2000

2004

2008

2012

2013 2015 2017

Password Hashing Competition (PHC)

2013-2015

Password hashing schemes

Organized by a group of passionate experts

Open to everyone, vendor-neutral, no sponsors

← → ↻ <https://password-hashing.net> 🔍 🗑️ 📌 🌐 🖨️ 🌱

Password Hashing Competition

[INTRODUCTION](#) / [CALL FOR SUBMISSIONS](#) / [CANDIDATES](#) / [TIMELINE](#) / [INTERACTION](#) / [EVENTS](#) / [FAQ](#)

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 panel

From industry, academia, US government

Crackers, software engineers, cryptographers...

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)

Motivations?

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

July 13, 2013

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



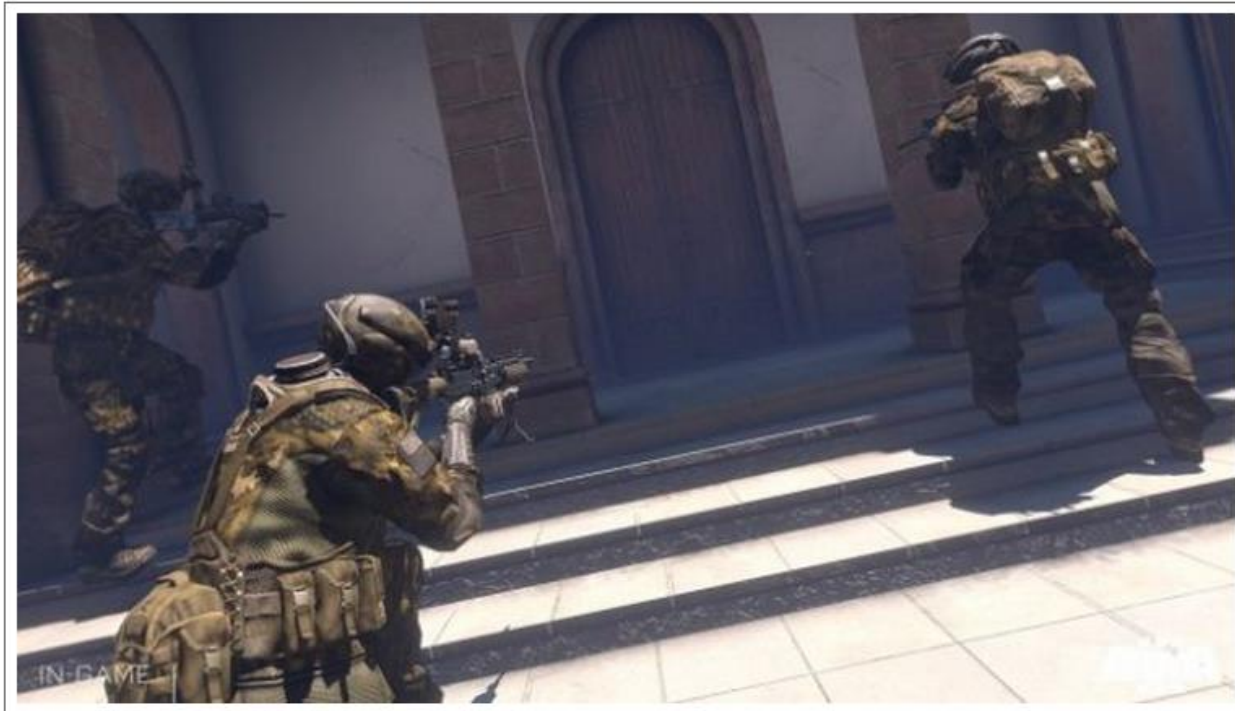
Comic-Con 2013

Prime


Sign

Bohemia Interactive hacked – usernames, emails and encrypted passwords taken

July 13, 2013 by [RetroCortana101](#)



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 impacted 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.

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:



January 3, 2014

← → ↻ 🏠 www.openssl.org/news/secadv_hack.txt

Website defacement: final details.
=====

Last updated: 3rd January 2014

On Sun 29th December 2013 at around 1am GMT the home page of www.openssl.org was defaced. We restored the home page just after 3am GMT and started forensics, investigation, and recovery.

The OpenSSL server is a virtual server which shares a hypervisor with other customers of the same ISP. Our investigation found that the attack was made through insecure passwords at the hosting provider, leading to control of the hypervisor management console, which then was used to manipulate our virtual server.

The source repositories were audited and they were not affected.

Other than the modification to the `index.html` page no changes to the website were made. No vulnerability in the OS or OpenSSL applications was used to perform this defacement.

Steps have been taken to protect against this means of attack in future.

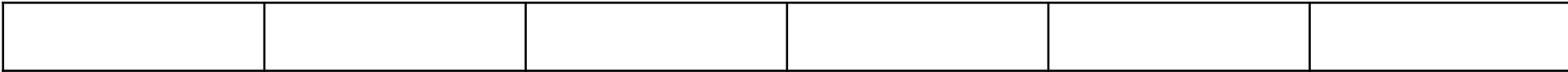
Experts' recommendation?

Just use **scrypt**!

script

1) Sequential initialization of a large array V

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



script

1) Sequential initialization of a large array V

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

b83546b4					
-----------------	--	--	--	--	--

script

1) Sequential initialization of a large array V

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

b83546b4	b2e2a2f5				
----------	-----------------	--	--	--	--

script

1) Sequential initialization of a large array V

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

b83546b4	b2e2a2f5	10cbd82a			
----------	----------	-----------------	--	--	--

script

1) Sequential initialization of a large array V

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

b83546b4	b2e2a2f5	10cbd82a	...		
----------	----------	----------	-----	--	--

script

1) Sequential initialization of a large array V

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

b83546b4	b2e2a2f5	10cbd82a	...	57500361	
----------	----------	----------	-----	-----------------	--

script

1) Sequential initialization of a large array V

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

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

script

1) Sequential initialization of a large array V

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

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

2) Sequential unpredictable accesses

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

script

1) Sequential initialization of a large array V

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

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

2) Sequential unpredictable accesses

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

script

1) Sequential initialization of a large array V

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

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

2) Sequential unpredictable accesses

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

script

1) Sequential initialization of a large array V

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

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

2) Sequential unpredictable accesses

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

script

1) Sequential initialization of a large array V

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

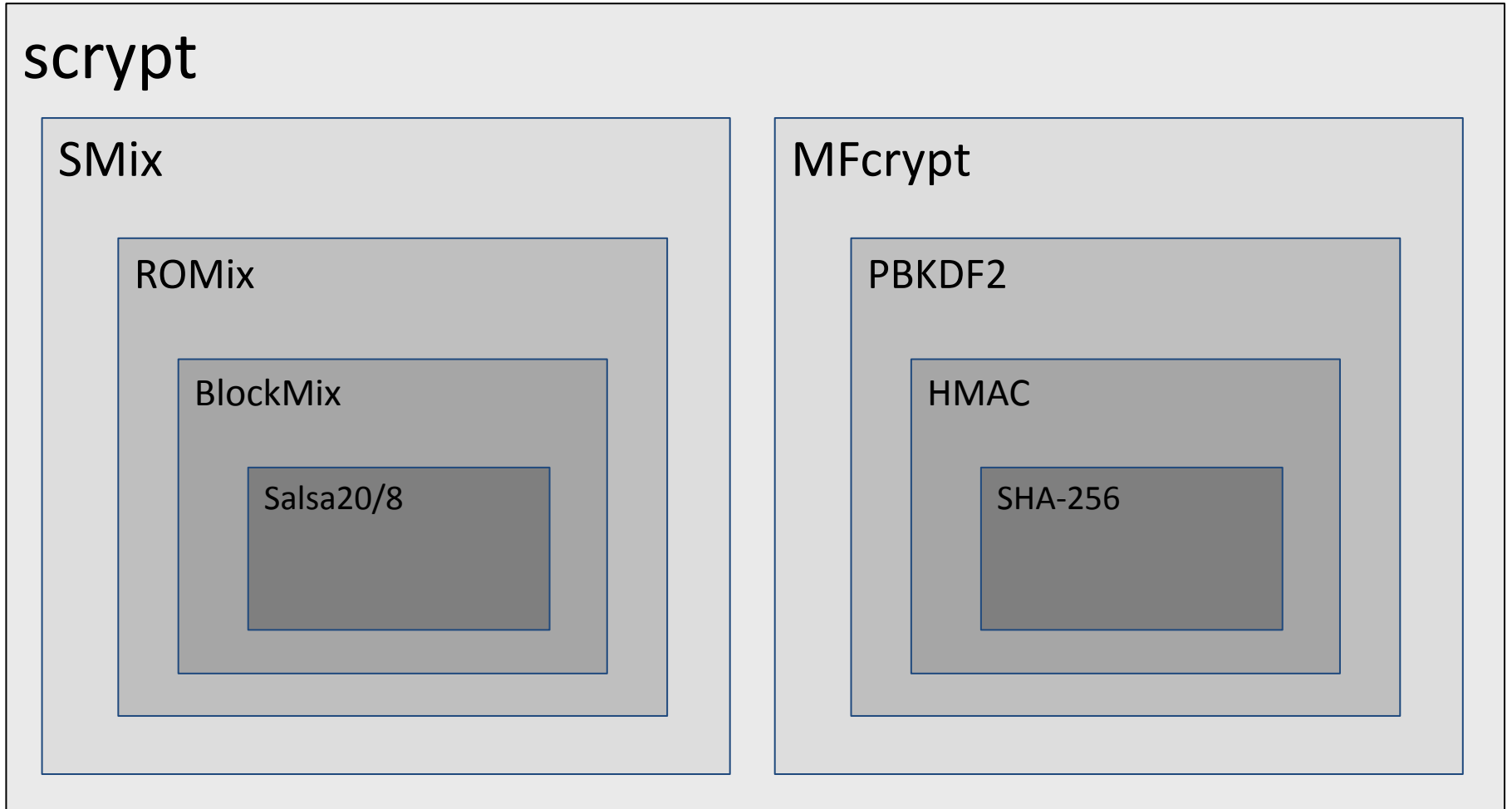
b83546b4	b2e2a2f5	10cbd82a	...	57500361	299c689f
----------	----------	----------	-----	-----------------	----------

2) Sequential unpredictable accesses

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

is script simple enough?

More core = more bugs = more tests, etc.



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*?

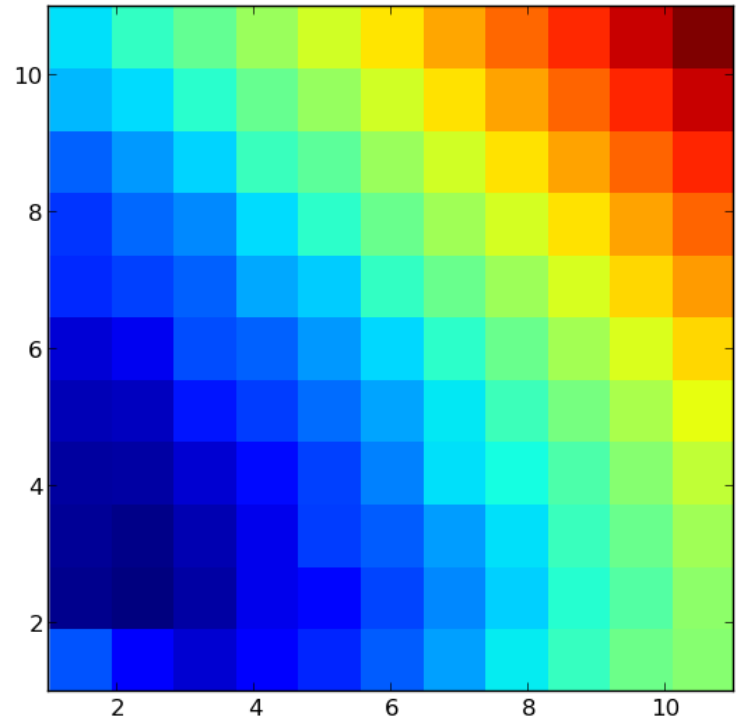
is scrypt user-friendly?

N and **r** have *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)

Potential problem for low-memory devices

Also impossible to increase only memory

We need something better

Call for submissions

The Password Hashing Competition (PHC) organizers solicit proposals from any interested party for candidate password hashing schemes, to be considered for inclusion in a portfolio of schemes suitable for widespread 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 available on the website of the project, <https://password-hashing.net>.

<https://password-hashing.net/call.html>

Minimal I/O requirements

- 0 to 128-byte password
 - Encoding of characters to bytes is up to users
- 16-byte salt
 - May support shorter and longer salts as well
- 1 cost parameter
 - May support 2 or more (e.g. time and memory)
- 32-byte hash
 - May support short and longer hashes as well
 - Convenient to support 256-bit key derivation

Evaluation criteria

Security and functionality

- Pseudorandom behavior
- Minimal speedup with crackers' SW or HW
- Effectiveness of the cost parameters
- Flexibility and scalability
- Resilience to side-channel attacks

Evaluation criteria

Simplicity

- Often overlooked in “clever” schemes
- **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 choices

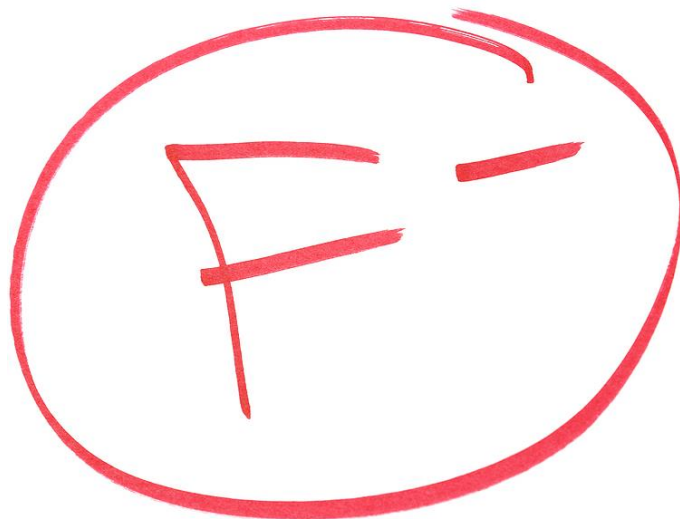
- *Application?* (key derivation, storage...)
- *Platform?* (64-bit SW, mobile, low-end...)
- platform-*optimized vs generic*
- *Length*: do we need more than 16 bytes?
- How to implement “*memory hardness*”?
 - reads vs. writes; blocks size; predictability and order; etc.
 - prove rigorous bounds on time-memory-tradeoff?
- What degree and type of *parallelism*?

WARNING



**CHALLENGES
AHEAD**

Cryptographic research



Todos includes:

Create generic constructions (like HMAC for MACs)

Prove rigorous security bounds on time/memory

Define minimal security requirements

Dedicated hardware architectures?

Cryptanalyze PHC candidates

Optimization and technology-dependency

Password hashing is very *technology-dependent*

For both defenders and attackers

How will server chips look like in 10 years?

What will be the most effective cracking method?

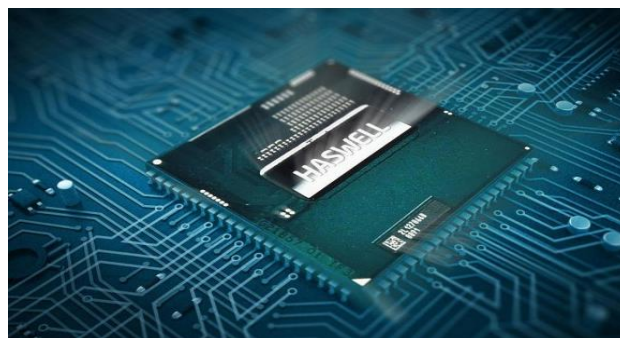
For example, hashes could be optimized for AVX2:

256-bit registers

SIMD arithmetic

Gather instructions

VPERMD, VFM*, etc.



-> Better security for AVX2 servers, but inconsistent performance accross platforms...

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?

Optimize a hash for JavaScript?



Addresses the *risk of DoS* on servers

Different models than store-and-compare-hashes?

Updatability



How to update hashes to a different security level?

Without requiring a new login

Schemes based on a fast hash as a proxy?

Motivations: adapt to new technology and research

Defenders (server CPU, cores available, etc.)

Attackers (hardware, techniques, etc.)

Catena: A Memory-Consuming Password Scrambler

Christian Forler, Stefan Lucks, and Jakob Wenzel

Bauhaus-Universität Weimar, Germany

{Christian.Forler, Stefan.Lucks, Jakob.Wenzel}@uni-weimar.de

Abstract. It is a common wisdom that servers should better store the one-way hash of their clients' passwords, rather than storing the password in the clear. This paper introduces **Catena**, a new one-way function for that purpose. **Catena** is memory-hard, which can hinder massively parallel attacks on cheap memory-constrained hardware, such as recent “graphical processing units”, GPUs. Furthermore, **Catena** has been designed to resist cache-timing attacks. This distinguishes **Catena** from **scrypt**, which may be sequentially memory-hard, but which we show to be vulnerable to cache-timing attacks.

Additionally, **Catena** supports (1) *client-independent updates* (the server can increase the security parameters and update the password hash without user interaction or knowing the password), (2) a *server relief* protocol (saving the server's resources at the cost of the client), and (3) a variant **Catena-KG** for secure *key derivation* (to securely generate many cryptographic keys of arbitrary lengths such that compromising some keys does not help to break others).

<http://eprint.iacr.org/2013/525.pdf>

More ideas

Programmable hashes

Algorithm = $F(\text{password})$

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

Conclusions

NORX CAESAR submission <https://norx.io>
with Philipp Jovanovic and Samuel Neves

PHC is starting, submit before ~~Jan~~ **March 31**
“Only lame crypto competitions do not postpone deadlines” 😊
<https://password-hashing.net>

Subscribe to the mailing lists

discussions-subscribe@password-hashing.net

crypto-competitions+subscribe@googlegroups.com