Cryptology Annual News Update and Vignette

Bill Ricker

for BLU.org (http://blu.org/cgi-bin/calendar/2023-sep)

Sept 20, 2023

Cryptology News Bulletins 2022-09 to 2023-08

“Abundance of Caution” is C-suite lingo for *“Oopsie, oh flying squirrel”*

OpenSSL 3 near-critical patch (Oct-Nov 2022)

https://www.phoronix.com/news/OpenSSL-1-November-2022 (https://www.phoronix.com/news/OpenSSL-1-November-2022) https://arstechnica.com/information-technology/2022/11/openssl-3-patch-once-critical-but-now-just-high-fixes-buffer-overflow/ (https://arstechnica.com/information-technology/2022/11/openssl-3-patch-once-critical-but-now-just-high-fixes-buffer-overflow/)

CVE-2022-37786 and CVE-2022-3602

> Second-ever OpenSSL critical vulnerability teased, 10 years after Heartbleed

downgraded from critical to merely high; but still important.

Passkeys

https://www.passkeys.io/ (https://www.passkeys.io/)

https://arstechnica.com/information-technology/2022/10/passkeys-microsoft-apple-and-googles-password-killer-are-finally-here/ (https://arstechnica.com/information-technology/2022/10/passkeys-microsoft-apple-and-googles-password-killer-are-finally-here/)

Jill (NatickFOSS) notes that this makes things harder on Executors. It requires both physiology and BT devices.

How do you change phones securely but prevent *jacking a phone change?

(Bob (NatickFOSS) says Fido alliance can provide a backup dongle for executors that overrides eyeball+phone in range?)

Integer Overflow in extended precision arithmetic

> Changes for libgmpxx4ldbl versions: Installed version: None Available version: 2:6.2.0+dfsg-4ubuntu0.1 Version 2:6.2.0+dfsg-4ubuntu0.1:

- SECURITY UPDATE: Integer overflow
  - Debian/patches/CVE-2021-43618.patch: prevent integer overflow in function mpz_inp_raw in mpz/inp_raw.c on 32-bit platforms.
  - CVE-2021-43618

Version 2:6.2.0+dfsg-4:

[Steve Robbins] * Add breaks for packages known to be broken by GMP 6.2.0. Closes: #950608.

LastPass break update

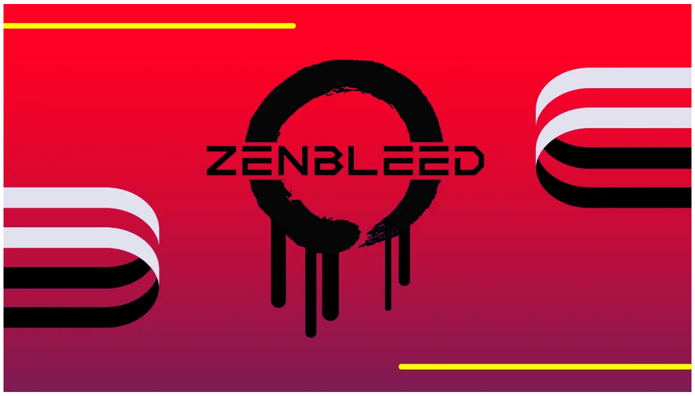
> [2022.12.26] Last August, LastPass reported a security breach, saying that no customer information—or passwords—were compromised. Turns out the full story is worse https://www.schneier.com/blog/archives/2022/12/lastpass-breach.html (https://www.schneier.com/blog/archives/2022/12/lastpass-breach.html)

possibly exploited to steal Craptocoyns ?!

Did victims have a weak passphrase, or were they actually victims of a Wallet breach and blaming it on LastPass ?

“ZENBLEED” - “Encryption-breaking, password-leaking bug in many AMD CPUs

could take months to fix”



> “Zenbleed” bug affects all Zen 2-based Ryzen, Threadripper, and EPYC CPUs.

July: <Ars (https://arstechnica.com/information-technology/2023/07/encryption-breaking-password-leaking-bug-in-many-amd-cpus-could-take-months-to-fix/)>; <CVE-2023-20593 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-20593)>; all Zen 2 products in shared use. Fix has up-to 15% performance impact except gaming? (*Your gaming system ought not be running others’ work anyway!*)
<Cloudflare analysis + remediation (https://blog.cloudflare.com/zenbleed-vulnerability/)>

& August: <HN: Collide+Power, Downfall, Inception (https://thehackernews.com/2023/08/collidepower-downfall-and-inception-new.html)>; <Google Security Blog: Downfall + Zenbleed (https://security.googleblog.com/2023/08/downfall-and-zenbleed-googlers-helping.html)>

- Not exactly a “side-channel attack” - though some call it such, including the linked HN article. but yes similar in that it leaks data from one process or virtual server to another, contrary to the conceptual Hardware/Software divide.
- Once again, this is a threat in shared server environments - But that’s the public cloud for you: no control over who is using the other cycles of the physical cores your code executes on.
- And alas an argument against participating in net-wide cycle-sharing for charity

> computations: evil code inserted into their calc module could use Zenbleed &/or Downfall to watch your code._

Backdoor? in TETRA TEA1 encrypted Police radios

80-bit commercial export-semi-restricted TEA1 key has far less than 80 bits entropy, deemed intentional backdoor – one of 5 CVEs resulting from reverse engineering.

- <TheReg (https://www.theregister.com/2023/07/24/tetra_radio_security_flaws/)>; <Wired (https://www.wired.com/story/tetra-radio-encryption-backdoor/)>; <HackADay (https://hackaday.com/2023/07/27/did-tetra-have-a-backdoor-hidden-in-encrypted-police-and-military-radios/)>
- Crypto Museum? <TEA (https://www.cryptomuseum.com/crypto/algo/tea/)>, <Tetra:Burst (https://www.cryptomuseum.com/radio/tetra/burst.htm#pub)>
- 2023-07 (https://www.schneier.com/blog/archives/2023/07/backdoor-in-tetra-police-radios.html)
- <Computerphile> video (https://youtu.be/Fy3Odm-dny0?si=2jGWY-44nGCUzByI)
- <tetraburst (https://www.tetraburst.com/)> by <MidnightBlue (https://www.midnightblue.nl/)> ; Slide Deck (https://www.cryptomuseum.com/radio/tetra/files/tetra_pres_20230809.pdf); GitHub (https://github.com/MidnightBlueLabs/TETRA_crypto)

The also found inadequate entropy in IV, using spoof-able network time, in the protocol, so applies to all `TEA{1..4}` levels. Incompetence or backdoor? Unclear.

They used a cache-timing side-channel attack to extract the firmware algorithm once, from the one compliant manufacturer that included the least reverse-engineering countermeasures; only needed doing once, and exposed the proprietary licensed algorithm used by all more careful manufacturers.

Once again, Kerckhoffs's principle (https://en.wikipedia.org

/wiki/Kerckhoffs%27s_principle) or Shannon’s Maxim is re-validated; secrecy of the mechanism or algorithm is a false protection. It protects proprietary manufacturers from competition, but it does not protect the users’ (customers’!!) data. (Their paper/Slide-deck (https://www.cryptomuseum.com/radio/tetra/files/tetra_pres_20230809.pdf) has a long list of other failures from ignoring Kerckhoffs’s principle.)

TEA1 was advertised pre-1997 as export-allowed to export-restricted nations, so the “80bit key” should have been understood by those knowledgeable in the art back then to have had a max useful strength of 40 bits anyway. That it was 32 instead of 40 is only mildly shocking - *32 is such a convenient number of bits* - and that it was 32 and therefor OK for export was even shared back in the day somewhat. (Without Internet, it didn’t spread far until messages were rediscovered.)

So, that the export-allowed product was WEAK should NOT have surprised any customer! Calling that a Backdoor may be an overstatement; assumption was any encryption allowed for export to hostile regimes was crap by law. (OTOH the missing 8 bits of Internet 1.0 max legal export key are a 2^{32}/2^40 = ⎅1/128 complexity factor, the difference between brute-force cracking N keys per day vs N keys per calendar quarter.)

That TEA1 was still sold - or worse, bought! - so weakened after 1997 128-bit limit is the only shocker.

The weak, spoof-able entropy in the IV (Initialization Vector (https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Initialization_vector_(IV)) in block cipher modes) *OTOH* is possibly an intentional back door to make key reuse/collision compromises blindingly obvious to State Actor cognoscenti - which makes message and key recovery trivial in specific cases, and depending upon keying, can break a whole network for a day, week, year after one pair of messages is broken.

ChatGPT implements Dunning-Kruger

Crypto

Miguel de Icaza (https://fosstodon.org/@Migueldeicaza@mastodon.social/110089879420243546)

> Tired: don’t implement your own cryptographic stack
> Wired: have Chat-GPT write it for you

If you want greater efficiency in writing bugs …

Similarly, reports seen that AutoPilot etc will cough up someone else’s secret key in suggested source code for a secret-key encryption module. Because it memorizes whatever it sees, and regurgitates on command.

Existence of Fernet Encryption implies Existence of Malört Encryption?

Fernet is Python recipe for symmetric encryption with authentication, using AES-128 CBC, SHA-256, PKCS#7 - so **if** *competently* implemented **and** application key mgt is likewise competent, could be better than Fernet/Malört simile might imply.

Fernet also supported in Scala, Rust, Perl.

Malware has started using Fernet for their payloads!

Should Fernet-using Malware be called Malörtware ?

<SANS ISC (https://isc.sans.edu/diary/Have+You+Ever+Heard+of+the+Fernet+Encryption+Algorithm/30146/)>

Key management is hard

- <2023-08 (https://www.schneier.com/blog/archives/2023/08/microsoft-signing-key-stolen-by-chinese.html)> Expired **Microsoft** signing key exfiltrated, use to sign code then accepted by Azure! Spin-off of Solar Winds network management vulnerabilities, compounded by not checking for key expiry.
  as usual, a bad fail is a chain of bugs and vulnerabilities that amplify one another.

- **Micro-Star** International Signing Key Stolen <2023.05.15 (https://www.schneier.com/blog/archives/2023/05/micro-star-international-signing-key-stolen.html)> aka MSI—had its UEFI signing key stolen last month.

- **Github ssh** fiasco

  - <GitHub Blog (https://github.blog/2023-03-23-we-updated-our-rsa-ssh-host-key/)>; <MJG (mjg59.dreamwidth.org/65874.html)>; <DF (https://web.archive.org/web/20230324160312/twitter.com/d_feldman/status/1639152037307744258)>; <SANS ISC (https://isc.sans.edu/podcastdetail.html?id=8426)>
  - Lessons:
    - Don’t add/push secrets to public repos.
    - Comb private repos for secrets not just licenses+IP before publishing

- and q.v. **Prime Trust** below

Craptocoyns aren’t crypto and aren’t coins

It’s Ponzi all the way down.

> Bitcoin - the most successful bug bounty program ever

- <Gerard’s recent review (https://davidgerard.co.uk/blockchain/2023/09/18/crypto-collapse-fortress-custody-hack-binance-stonewalls-sec-ftx-to-dump-its-crypto-genesis-sues-dcg-new-york-restricts-crypto-listings/)>
    - latest convictions
    - latest bankruptcies
    - wallstreet losing interest
- <Nick Weaver on Regulating Cryptocurrency (https://law.yale.edu/sites/default/files/area/center/isp/documents/weaver_death_of_cryptocurrency_final.pdf)>

… continued …

Craptocoyns: Wallet Key loss = bankruptcy

“Craptocoyn startup loses wallet key”

<2023-09 (https://www.schneier.com/blog/archives/2023/09/cryptocurrency-startup-loses-encryption-key-for-electronic-wallet.html)>

> The cryptocurrency fintech startup Prime Trust lost the encryption key to its hardware wallet—and the recovery key—and therefore $38.9 million. It is now in bankruptcy.

ironic name!

> I can’t understand why anyone thinks these technologies are a good idea.

agree totally.

Craptocoyns: “MILKSAD”: Cryptographic Flaw in Libbitcoin Explorer Cryptocurrency Wallet

More Dunning-Kruger crapto? or intentional backdoor to facilitate thefts?

Low entropy, non-random seed (clock) renders a secure PRNG insecure; lib docs supposedly have caveat not to use the `bx seed` but general Bitcoin docs recommend using it for wallet generation.

- <2023-08 (https://www.schneier.com/blog/archives/2023/08/cryptographic-flaw-in-libbitcoin-explorer-cryptocurrency-wallet.html)> ↘ <MilkSad (https://milksad.info/)> & <Medium (https://medium.com/asecuritysite-when-bob-met-alice/a-novice-mistake-meet-milk-sad-and-the-32-bit-key-ba308fb2b633)> *“A Novice mistake”*
- <CVE-2023–39910 (https://nvd.nist.gov/vuln/detail/CVE-2023-39910)>

> “Never attribute to malice that which is adequately explained by incompetence.”

But … as a scam it looks pretty smooth.

- <CVE-2023–39910 (https://nvd.nist.gov/vuln/detail/CVE-2023-39910)>:

> The cryptocurrency wallet entropy seeding mechanism used in Libbitcoin Explorer 3.0.0 through 3.6.0 is weak, aka the Milk Sad issue. The use of an mt19937 Mersenne Twister PRNG (https://duckduckgo.com/?t=ffab&q=mt19937+Mersenne+Twister+PRNG&atb=v265-1&ia=web) restricts the internal entropy to 32 bits regardless of settings. This allows remote attackers to recover any wallet private keys generated from "bx seed" entropy output and steal funds. (Affected users need to move funds to a secure new cryptocurrency wallet.)

> NOTE: the vendor's position is that there was sufficient documentation advising against "bx seed" but others disagree. NOTE: this was exploited in the wild in June and July 2023.

From https://milksad.info/ (https://milksad.info/) :

> Popular documentation like "Mastering Bitcoin" suggests the usage of `bx seed` for wallet generation.

> Why the silly "Milk Sad" name? Running bx seed on 3.x versions with a system time of 0.0 always generates the following secret:

```
milk sad wage cup reward umbrella raven visa give list decorate bulb
gold raise twenty fly manual stand float super gentle climb fold park
```

> When?
> The main theft occurred around 12 July 2023, although initial exploitation likely began at a smaller scale in May 2023.
> A separate but similar vulnerability in another wallet software was detected in November 2022 and actively exploited shortly after, which may be the prequel to this story.

Crypto News Feature: updating Post Quantum Cryptography

Review: What’s Quantum Computing?

See last year’s status (blu.org/meetings/2022/09/Cryptology_Annual_News_Update_and_Vignette_NOTES.flipbook/#page/n1)

The only known photo of
Schrodinger's cat.

Quantum Superposition (https://en.wikipedia.org/wiki/Quantum_superposition) when used for computing.

- QC measured in **“qubits”** not bits

- 30% True, 70% False.

> Such bits are in quantum superposition of True and False, which is a *bug* in classical computing but a *feature* in QC.

> This allows non-deterministic algorithms (https://en.wikipedia.org/wiki/Nondeterministic_algorithm).

Review: Kinds of Quantum Hardware

- Quantum Annealing (https://en.wikipedia.org/wiki/Quantum_annealing) - big qubit counts, great for optimization problems

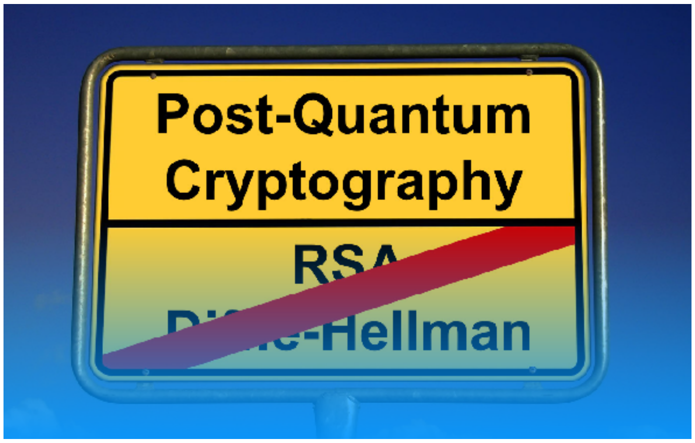
  but not cryptology. (?yet?) *Not general purpose.*

- Quantum Circuit/Logic (https://en.wikipedia.org/wiki/Quantum_circuit) - small numbers of qubits so far.

> In theory, algorithms for these hardware types can use non-deterministic parallelism to evade classical performance limits, and in particular, could allow factoring fast enough to be dangerous, provided big enough quantum circuits can be made to work.

Review: We’re discussing PQC before QC?

Yes !



- **Quantum Cryptography** (https://en.wikipedia.org/wiki/Quantum_cryptography)
  - theoretically using entangled quantum states
  - to create an encryption
  - or an anti-eaves-droppable connection

(***Chinese Space Agency claimed to have demonstrated?***)

- Quantum Crypt**analysis**
  - Using Quantum Computing to defeat classical PKI encryption
- **Post Quantum Cryptography** (https://en.wikipedia.org/wiki/Post-quantum_cryptography)
  - new classical encryptions that can resist Quantum Cryptanalysis,
  - so read as Ready for post-**(**Quantum-Computing**)** Cryptography.

Review: What’s the problem?

- Unbreakable ciphers aren’t always unbreakable, for always.
- QC *could theoretically* break most PKI
  - Schor’s Algorithm / Grover’s / VQF
  - discrete log as well as prime factoring, even elliptic curves

Every unbreakable cipher has been broken eventually (at least partially[1]).

20thC RSA and other PKI not guaranteed proof against either of:

- major breakthrough in number theory (factoring &/or discrete log), or
- quantum hardware + algorithms (practical fast factoring)

<Schor's Algorithm (https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm)> in theory would factor fast on enough quantum circuits but 21 is not a large number yet. (see also Wikipedia (https://en.wikipedia.org/wiki/Shor%27s_algorithm). Some say 433 bits on IBM Osprey QC is enough for RSA2048 with Schor's algo needing 372 Qubits (with pre-processing and post-processing), but will it work? Schneier and Schor doubt it. (https://www.schneier.com/blog/archives/2023/01/breaking-rsa-with-a-quantum-computer.html) *Shouldn't someone try it?*)

Other probabilistic quantum algorithms (Grover (https://en.wikipedia.org/wiki/Grover%27s_algorithm), GEECM (https://en.wikipedia.org/wiki/Lenstra_elliptic-curve_factorization#Quantum_version_(GEECM)), Variational Quantum Factoring (VQF) (https://arxiv.org/abs/2012.07825)) can do *some* much bigger numbers (*which may just define new class of unsafe primes??*), and with classical pre-processing, can use a much smaller number of qubits than the ^*obvious*^ $\log_2 N$.

not clear this will ever be able to generally break RSA4096, but it's not impossible, so prudent to plan for that day.

Review: Generalization of Forward Secrecy

- Classical "Forward Secrecy" - old messages not broken by later loss of host key

- Generalized: old saved messages not broken by breakthroughs either.

- Realistic threat?

  - VENONA + GEE 1940s {BLU Sept 2018 (http://blu.org/meetings/2018/09/)}
  - NSA Utah Data farm, 2013 (https://en.wikipedia.org/wiki/Utah_Data_Center#Structure)

```
* VENONA: It worked Once! {[BLU Sept 2018](http://blu.org/meetings
/2018/09/)}
* We now have a Vacuum Cleaner of Holding (_Greenpeace photo c/o
Wikimedia_)
```

So yes, it **can** happen again.

Normal Forward Secrecy (https://en.wikipedia.org/wiki/Forward_secrecy) requires that if e.g. the Host Key is compromised later, any retained cryptograms sent with nonce keys negotiated with the compromised Host Key aren't also compromised.

This is nice, but we'd also like to protect against advances of technology, e.g. fast factoring or solutions of discrete logs.

This may not be within *your* threat model, yet, but in dystopian plausible futures, things you've already discussed/downloaded might be retroactively illegal/disloyal and oops.

Review: NIST’s Post-Quantum Cryptography Standards

> The goal of post-quantum cryptography (also called quantum-resistant cryptography) is to develop cryptographic systems that are secure against both quantum and classical computers, and can interoperate with existing communications protocols and networks. – **NIST**

… and have it ready for use not only before quantum breakthrough but early enough (roughly now) that anyone who wishes to avoid save-intercepts-now-to-break later; although it may already be too late WRTO NSA archive?

Review: NIST PQC Competition

National Institute of Standards & Technology started a multi-round competition, similar to with AES and SHA3 competitions

- NIST announcement (https://csrc.nist.gov/projects/post-quantum-cryptography)
- NIST Q&A (https://www.nist.gov/blogs/taking-measure/post-quantum-cryptography-qa-nists-matt-scholl)
- <Schneier on PQC (https://www.schneier.com/blog/archives/2022/08/nists-post-quantum-cryptography-standards.html)>

NIST, the Bureaucracy formerly known as NBS.

This competition was “more brutal” than prior; of 69 candidates, peer cryptanalysis has broken 62. So far.

PQC 2023

- Post-Quantum Cryptography Conference (https://pkic.org/events/2023/post-quantum-cryptography-conference/) (Friday March 3, 2023 - Ottawa, Canada)
- How to Prepare Your PKI for Quantum Computing (https://www.keyfactor.com/blog/how-to-prepare-your-pki-for-quantum-computing) (April 28, 2022)
- (podcast) Root Causes 286: PKI and PQC in New White House Cybersecurity Initiative (https://www.sectigo.com/resource-library/root-causes-286-pki-and-pqc-in-new-white-house-cybersecurity-initiative) (Mar 16, 2023)

Quantum Cracking / PQC Update

- **RSA2048 in play or not?** - Chinese academic paper claiming 2k bit RSA within range of current gen NON-fault-tolerant QC, no great surprise given Qubits available and theoretical algorithm size. Schor and Schneier unconvinced - does it actually converge w/o FT? <Schneier 2023-01 (https://www.schneier.com/blog/archives/2023/01/breaking-rsa-with-a-quantum-computer.html)>

- Schneier “You Can’t Rush PQC Standards” (https://www.schneier.com/blog/archives/2023/08/you-cant-rush-post-quantum-computing-standards.html)

- Quantum resistant hybrid-signing FIDO2 keys for 2FA (https://youtu.be/QrAbO5G6UnM?si=Q4pI7YcLf1jUQ92H)

- **Side-Channel Attack against CRYSTALS-Kyber** (https://www.schneier.com/blog/archives/2023/02/side-channel-attack-against-crystals-kyber.html)

> [2023.02.28] CRYSTALS-Kyber is one of the public-key algorithms currently recommended by NIST as part of its post-quantum cryptography standardization process. Researchers have just published a side-channel attack—using power consumption —against an implementation of the algorithm that was supposed to be resistant against that sort of attack. The algorithm is not “broken” or “cracked”—despite headlines to the contrary—this is just a side-channel attack. What makes this work really interesting is that the researchers used a machine-learning model to train the system to exploit the side channel.

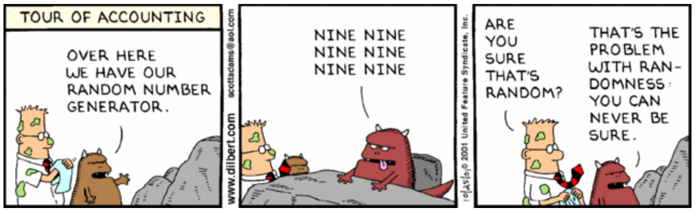
OTOH as seen in TETRA:BURST, a side-channel attack can be used to extract key or algorithm from a piece of equipment that falls into opponent lab.

NIST PQC Schedule

- 2023-08 FRN RFC (https://csrc.nist.gov/news/2023/three-draft-fips-for-post-quantum-cryptography) Draft Standards FIPS 203, 204, 205.
- 2024-04 Fifth PQC Standardization Conference
- 2024 FIPS Standards; `FIPS Allowed`.
- 202⅚ FIPS certification for the PQC algorithms; `FIPS Approved`.

https://www.nist.gov/programs-projects/post-quantum-cryptography (https://www.nist.gov/programs-projects/post-quantum-cryptography)

REVIEW: Known weaknesses


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
www.dilbert.com scottadams@aol.com
NINE NINE NINE NINE NINE NINE
© 2001 United Feature Syndicate, Inc.
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

- breaks have eliminated 62 of 69 entrants in Rounds 1 to 4
- including the two front-runners, Rainbow and SIKE
- 7 remain, will they survive?
- FALCON would be compromised by a lack-of-randomness in salt, or failure to salt, as repeating same key and hash again gives too much information.

Isn’t non-random or uniformly-blank Salt an unlikely failure?

No. It’s happened.

- Numerous implementations have failed to salt encryption of small data despite warnings.
- DEBIAN broke system `random`[2] which compromised many SSH keys.
- our historical vignette in prior years has discussed danger of key reuse in WW2 and Cold War. Lack of Salt = Key Reuse.

Lack of randomness failure isn’t just hypothetical, lots of SSH keys got invalidated in 2008 because they were well-known-primes.

(WTF? Yep. Debian packagers applying *normal best practices* where they shouldn’t even touch had *removed* the entropy-harvesting because Valgrind and Purify gave “accessing uninitialized memory” warnings. Well yeah, that’s how we harvest entropy! Another problem (mostly solved?) is host key generation at VM start - the VM’s entropy is rather deterministic at that point. Similarly, optimizing compilers removing zeroing memory prior to releasing it can allow keys to leak into the memory pool. Cryptographic software is an ongoing a battle against computer ^science^ that ^knows better^.)

And failure to salt wouldn’t surprise me when non-specialists (applications developers, database programmers, protocol developers) who should stick to packaged PKI use-case libraries (e.g. NaCl (https://en.wikipedia.org/wiki/NaCl_(software))) try to use cryptographic primitive routines directly to avoid dependencies.)

And 2023 has a few more low-entropy initialization examples added to the list!

History Vignette - Philips PX-1000Cr - NSA and the consumer

Text Lite “pocket telex” / pocket teletype

NSA Logo + PS1000 mashup by Klaus Schmeh

- 1980-1982 “Pocket Telex” PX-1000 text communicator for POTS network
  - developed by NL “Text Lite BV” corp
  - an OEM alphanumeric keyboard with one-line display, with onboard DES encryption and acoustic coupler.
- 1983 Philips became the major seller of
- 1983(late) NSA/GCHQ objected to even 56-bit DES being available to civilians.

    - (At the time, Export Netscape was limited to 40-bit for naughty countries. DES was fairly strong vs slow computers.)
- 1984 Crypto-free edition with calculator function released to temporize
    - crypto-free version had blue button instead of red key/text/code button
- 1985 PX-1000Cr NSA-approved LSFR edition were developed in 1984 and sold in 1985 and later
    - and all unsold DES units and ROMs were "*sold*" to NSA at a nice wholesale profit.
- 1985 C-Mail capable versions of both Calc and Crypto units
- followed by compatible improvements PX-1200 (https://twitter.com/Foone/status/998706980049989632) and PX-2000 (manufactured by Seiko/Epson for Text Lite), sold as Philips and TEXT TELL brands.

red-code-button - Crypto Museum.jpg

- Acoustic coupler was aggressively half-duplex: single transducer, would have to move it to microphone or earpiece when sending or receiving. Which made it much more pocket friendly! *Brilliant!* (See pix on Crypto Museum page (https://www.cryptomuseum.com/crypto/philips/px1000/index.htm) for how this worked.) The PX-2000 (https://www.cryptomuseum.com/crypto/philips/px2000/index.htm) had nearly-as-slim and brilliant full-duplex acoustic coupler, RS232 serial port, terminal emulation, & word processing on a huge (for a pocket device!) 80x8 LCD, and still brief-case or large-pocket friendly.
- Although was not built or exported from US or UK, Philips defense division had NATO contracts, including a cryptographic subsidiary, so their Consumer products division was expected to play nice.

Analysis Timeline

- In 2014, Ben Brücker's Bachelor Thesis "Government intervention on consumer crypto hardware: A look at the PX-1000 before and after the NSA's involvement" (working with Crypto Museum example).
    - Disassembled & reverse-engineered and validated original DES block cipher implementation
    - Disassembled & partially analyzed the NSA LSFR code - determined it was a stream cipher
    - Raised question if this meant NSA compromised Nelson Mandela's support-team's comms!
- 2019 NL Crypto Museum reports on the above research publicly.
- In 2021, Stefan "Stef" Marsiske completed analysis of the NSA LSFR algorithm in the PX-1000Cr
    - determined key-space was actually 32 bits
    - and developed a full break: 17 chars of ciphertext yield the message key
    - which might or might not have been tractable back then
    - and observed another weakness that might be 1980s backdoor.
- LSFRs are maximal length so not obviously suspect, but
- Leaks keystream bits, so detecting key reuse (plain-plain compromise) is trivial

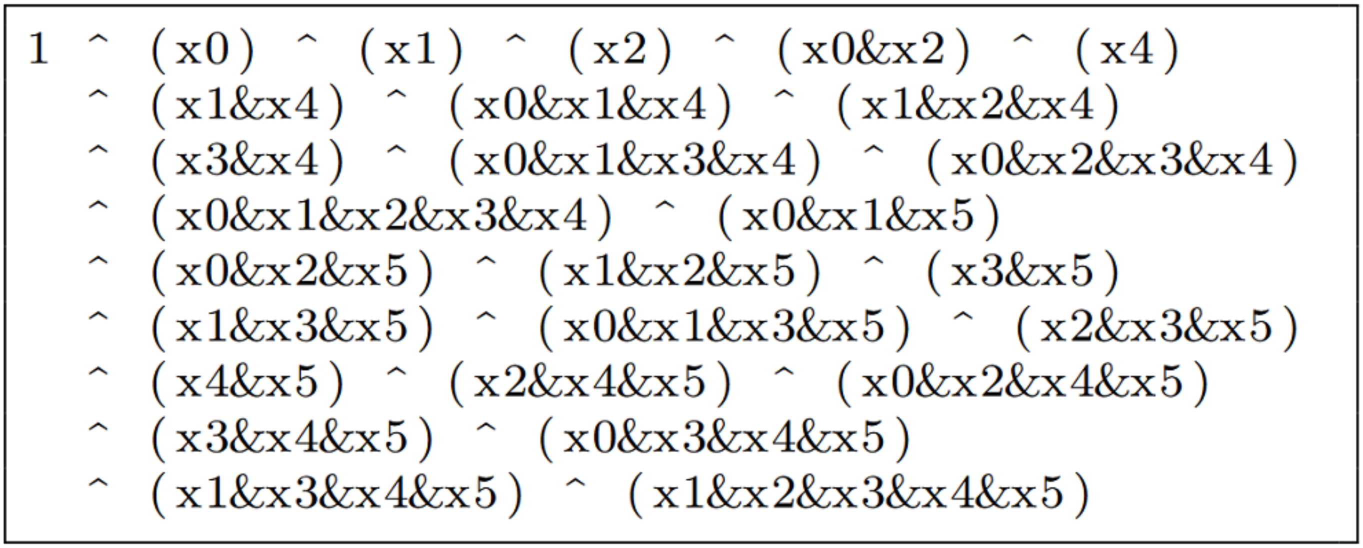
---

PX-1000 internals NL Crypto Museum.jpg

- Since the covert effort to prepare Nelson Mandela for freedom and return to active politics (in parallel with negotiations) was organized from Netherlands (with Comms center in London), PX-1000 units were used.

- Historians wondered Mandela’s Dutch support team used the compromised 1985 PX-1000Cr models and thus were read by NSA+GCHQ?
- Indications are that a sympathetic Philips Consumer insider directed them to find the older DES units, raising their security.

> As far as I understand it, Marsiske converted the encryption scheme into a Boolean function, which he was able to solve using a SAT solver. – Klaus Schmeh (https://scienceblogs.de/klausis-krypto-kolumne/how-the-nsa-weakened-an-encryption-device/)



```
1  ^  (x0)  ^  (x1)  ^  (x2)  ^  (x0&x2)  ^  (x4)
   ^  (x1&x4)  ^  (x0&x1&x4)  ^  (x1&x2&x4)
   ^  (x3&x4)  ^  (x0&x1&x3&x4)  ^  (x0&x2&x3&x4)
   ^  (x0&x1&x2&x3&x4)  ^  (x0&x1&x5)
   ^  (x0&x2&x5)  ^  (x1&x2&x5)  ^  (x3&x5)
   ^  (x1&x3&x5)  ^  (x0&x1&x3&x5)  ^  (x2&x3&x5)
   ^  (x4&x5)  ^  (x2&x4&x5)  ^  (x0&x2&x4&x5)
   ^  (x3&x4&x5)  ^  (x0&x3&x4&x5)
   ^  (x1&x3&x4&x5)  ^  (x1&x2&x3&x4&x5)
```

Boolean functional equivalent - Stef

(SAT solver (https://en.wikipedia.org/wiki/SAT_solver)=“theorem prover” ie. analogous to Prolog and related resolution theorem provers, but specifically optimized for the “Boolean satisfiability problem” (https://en.wikipedia.org/wiki/Boolean_satisfiability_problem); NP-complete *in general* but tractable in normal cases. 50 seconds on a single modern thread might have been too slow for easy NSA cracking in 1985-1995, but likely still cheaper on NSA’s CRAY-1, CRAY-MP, & CRAY-2, than DES key search. Effectively solving multiple equations but over binary variables instead of natural numbers or wild floats.)

DES by Brute Force: Estimated cost of a specialized hardware dedicated DES cracker dropped from $20M(1977) to $1M(1993), but that was theory as far as the open literature knows. One *might* presume NSA and GCHQ could afford at least custom unit between them since they apparently could and did afford multiple

CRAY-1 at $8M@, and so likely *should* be presumed to have had at least one and eventually several. But use of a singular or scarce resource would have had to have been prioritized, so unless a message was believed important it might never have been cracked. Practical custom hardware didn’t appear publicly until 1998 (EFF $250k). First open literature crack was 1997, using Internet screensaver cycles. As of 2012, and still on offer in 2023, a commercial custom server ($100K build) offered multiple levels of service and SLA (95% chance free if your NTLM challenge uses *their* known plaintext 1122334455667788 for which they boast “world’s largest rainbow table” (amortized cost of $20 if it fails 5% of time is $1@), and from $20 up-to $1000 for full keyspace search for other network tokens. Same or similar server presumably could attack full blocks at potentially higher prices, but that’s not turnkey there⎅. https://crack.sh/get-cracking/ (https://crack.sh/get-cracking/) https://crack.sh/#faq (https://crack.sh/#faq)

Bibliography & Footnotes

My talks

The **YouTube** of this presentation will be linked on <BLU.org (https://BLU.org)> along with these slides and extended notes *etc* as <2023-sep (http://blu.org/cgi-bin/calendar/2023-sep)> as per usual.

Prior talks in this series (http://blu.org/cgi-bin/calendar/speakers/b-ricker1)
- most talks have slides &/or YouTube attached, sometimes extras.
Alas the YouTube audio pre-pandemic wasn’t great, BLU will need a donation of a wireless clip-on mike if we ever return to Hybrid/In-Person meetings. Or we all need to wear a wired or BT headset while presenting in person?

News + Focus

News and **Focus** sections have embedded links.

Good security news streams to either research history or to follow year round are https://www.schneier.com/crypto-gram/ (https://www.schneier.com/crypto-gram/) and https://isc.sans.edu/ (https://isc.sans.edu/), the latter being less cryptologic and more operational in focus – but both cover the wide span of vulnerabilities, tools, remediations, etc, not just the cryptologic that I'm cherry-picking here.
Highly recommended.
Start your day with the 5 minute SANS Internet Storm Center StormCast pod-cast; the Red Team is, so, so should you.

Historical Vignette - Bibliography specific for this year

- Klaus's Krypto Kolumne blog (https://scienceblogs.de/klausis-krypto-kolumne/how-the-nsa-weakened-an-encryption-device/)
- NL Crypto Museum (https://www.cryptomuseum.com/news/)
  - PX-1000 (https://www.cryptomuseum.com/crypto/philips/px1000/index.htm)
  - documenting NSA algorithm (https://www.cryptomuseum.com/crypto/philips/px1000/nsa.htm)
  - Stefan Marsiske, Breaking the PX-1000Cr (https://www.cryptomuseum.com/crypto/philips/px1000/stef.htm)
  - PX-2000 (https://www.cryptomuseum.com/crypto/philips/px2000/index.htm)
- Stef's Feb 2022 attack published in PoC||GTFO 21:21 (https://www.alchemistowl.org/pocorgtfo/pocorgtfo21.pdf) (from p.59 in PDF)
  - & additional on Stef's blog (https://www.ctrlc.hu/~stef/blog/posts/pocorgtfo_21_12_apocrypha.html)
  - & Stef's GitHub (https://github.com/stef/px1000cr)
- Computing History(UK) (https://www.computinghistory.org.uk/det/16481/Text-Tell-PX-1000/) {the Camb.Uni one, *not* BP one}

- Videos from Stef's 2021 Camp++ talk
  - Y-T: Camp++ 0x7e5 // A historical NSA backdoor by Stef (https://www.youtube.com/watch?v=8VTmfiifkRU) 0:52:57
  - S3: another copy (https://hsbp.s3.eu-central-1.amazonaws.com/campppp7e5/backdoor.mp4) 0:52:58
  - *audio gets better quickly, don't despair*
- Foone's PX-1200 photos, archived (https://web.archive.org/web/20220223142223/twitter.com/Foone/status/998706980049989632)
  - which box wears the OEM brand "Text Lite"
  - and shows PX-1000 on the display photo, despite box saying PX-1200!

Cryptologic History - general references

- ***Bletchley Park Podcast*** (https://duckduckgo.com/?t=ffab&q=Bletchley+Park+Podcast) on your favorite pod server
  - FB (https://www.facebook.com/BletchleyParkPodcast/)
  - BPP Keyword index (file:///home/wdr/Bill/git-other/articles/Cryptology/BletchleyParkPod/BPP-Keywords.html)
    by me, current thru fall '22
- **Books**
  - *The Code Breakers*, revised & updated; Kahn, David; 1996: NY S&S.
  - *Decrypted Secrets*; Bauer, F.L.; 1997: Heidelberg, Springer.
  - https://www.schneier.com/books/ (https://www.schneier.com/books/)
- **Websites**
  - https://en.wikipedia.org/wiki/Cryptography (https://en.wikipedia.org/wiki/Cryptography)
  - https://cipherhistory.com/ (https://cipherhistory.com/)
  - https://cryptomuseum.com/ (https://cryptomuseum.com/)
  - https://www.nsa.gov/History/Cryptologic-History/ (https://www.nsa.gov/History/Cryptologic-History/)
  - https://nsarchive.gwu.edu/ (https://nsarchive.gwu.edu/) (Academic National

Security Archive at GWU; FOIA Declassified)

- Declassified post-WW2 **TICOM** (https://en.wikipedia.org/wiki/TICOM) reports http://www.ticomarchive.com/ (http://www.ticomarchive.com/) (*Signals and Cryptologic Intelligence equivalent of better-known Operation Paperclip, etc.*)

---

1. See our prior discussions (http://blu.org/meetings/2018/09/) of GEE, VENONA for breaks of One Time Pad↩

2. `DSA-1571-1 openssl` predictable random number generator (https://www.debian.org/security/2008/dsa-1571) <CVE-2008-0166 (https://security-tracker.debian.org/tracker/CVE-2008-0166)> <Schneier (https://www.schneier.com/blog/archives/2008/05/random_number_b.html)> ↩