Random Acts of Secrecy

Entropy: What it is, why it matters to digital security now and in the future, and what needs to change. *A short, friendly discussion*

Scott Bledsoe, Chief Executive Officer
Brian Anderson, Chief Marketing Officer
Joseph Mulvihill, Senior Technical Business Analyst
Theon Technology

THEON TECHNOLOGY

Abstract

Cryptography runs on a marriage of algorithmic locks and numerical keys, and entropy is on the shortlist of essential factors that support the encryption and decryption of data. In modern information theory entropy is a measure of the randomness or uncertainty of a given variable or system. We need entropy – the purer, the better – as an ingredient in the production of effective cryptographic keys.

Long-established notions of sufficient entropy are being challenged today by more ingenious hackers, advances in higher mathematics, and, on the horizon, the threat of quantum computing. Some mainstay hardware-based random number generators use methods with exploitable vulnerabilities. Some keys, particularly those seeded by prime numbers or multiples thereof, could be cracked by quantum processing force powerful enough to expose heretofore invisible, embedded patterns. In the words of one researcher, reading cryptographic keys this way is "still exponentially hard, but it's exponentially easier than we thought." Those with a stake in digital security should therefore be discriminating about entropy quality.

Theon Technology offers a cryptography solution that generates more random, less vulnerable keys with intrinsically better entropy. It sets a new standard for software-based cryptography, moving the world closer toward the goals of quantum-proof encryption and perfect secrecy for businesses.

CONTENTS

Introduction

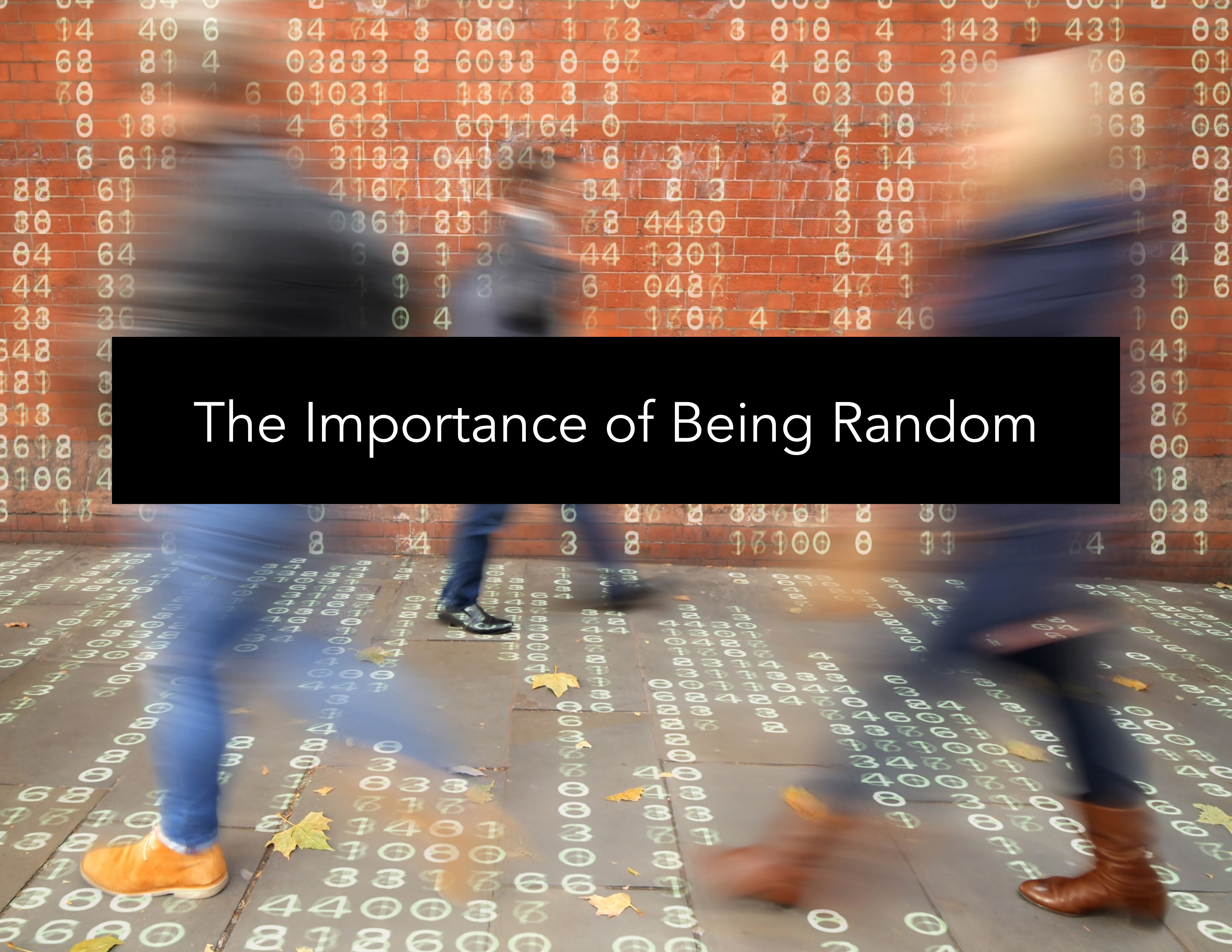
It's the random, wasted energy you see thrown off by a dying campfire or melting ice bucket. It's the residual, unproductive heat rising from the hood of your car when you shut the engine off after a long drive. In its purest form it's entirely patternless and unpredictable. You cannot forecast with confidence which glowing log in the campfire will crumble next, nor when – let alone which way the sparks will fly.

There's a name for this phenomenon: entropy.

First identified as a byproduct of thermal energy transfer, entropy in most situations is unremarkable and can even be a nuisance, as when melting ice dilutes the lemonade in your glass. But in digital security, weirdly enough, entropy is anything but a nuisance. It's centrally important. In the cryptography world entropy is mother's milk, on the shortlist of essential factors that help assure security, therefore the safety and stability of our world. We need entropy: the purer, the better. In fact, low-entropy security conditions can drive cryptography failures.

How did critical security technology, and by connection human productivity, grow so dependent on unproductive randomness? Why does the quality of entropy affect our ability to defend data and keep secrets? Why is weak entropy a security threat? All these things are worth knowing. The answers may even help equip you to evaluate security solutions. Because the subject matter is typically rare-air science talk, many people don't get entropy or appreciate the finer, more efficacious varieties. Let alone why it matters in the field of cryptography.

But in just a few minutes, you will.



The Importance of Being Random

Secrets demand to be kept in our digital world – secrets from nuclear launch codes all the way down to your supermarket loyalty card number.

Encryption technology is our essential, go-to secret-keeping engine. In olden days businesspeople told one another nothing happens until somebody sells something. Today nothing much can happen until sensitive data is properly secured. Digital data is now the world's most coveted, protection-worthy resource. The amount of it we keep on hand expands at a neck-snapping rate, and countless malefactors are laboring 24/7 trying to steal it. Subpar cryptography means no security for anyone.

Modern digital cryptography runs on a marriage of algorithmic locks and numerical keys – a paradigm rooted in pre-digital cipher formulas and decoding tools. (Spies and military commanders depended on them ages before modern computers.) Today's standard algorithms encrypt digital information into ciphertext, which to unauthorized eyes looks like gibberish, and decrypt it into coherent form only for parties who wield the proper key.

These crucial lock-and-key combos come in two basic models. When two parties use the same private key for encoding and decoding, it's symmetric encryption. When a system uses one private key plus one public one, connected via a complex mathematical algorithm, it's asymmetric. Asymmetric encryption is a very common security solution: RSA, DSA, and ECC are a few of the well-known algorithms pumping out public keys at scale, all day long, to keep the internet humming. You couldn't e-sign a contract or check your bank balance without asymmetric encryption at the ready.

But the model sure puts a lot of pressure on those keys, public or private. We need them to be secure.

A cryptographic key in action is just a string of numerals mixed with other characters. Most encryption algorithms are commonly leveraged, off-the-shelf hunks of math, so the quality of cryptographic keys really matters. To achieve good encryption, we need cryptographic keys to tick three boxes as seen on the next page.

1

Keys really ought to be *unique*: one-off "bit strings" of characters. It would undermine the system to have duplicate, identical specimens of bit strings stamped out at scale like sleeves of Ritz crackers.

2

Those bit strings are more secure when *long*; our math alphabet only supplies ten numerals to work with. Almost immediately, a cryptographic key starts reusing numerals. So longer bit strings are better – and bigger, more complex digital files call for commensurately longer keys.

3

The order in which characters are arrayed on the bit string ought to be genuinely *random*. And real, pattern-free randomness is fiendishly difficult to achieve. Bit string randomness depends in large part on the quality of numerical "seeds" used to generate keys.

Here's a simple way to think about cryptographic key complexity. Consider how difficult it is for you to come up with truly random account passwords.

You probably keep track of dozens of passwords in your life, maybe hundreds. Your goal, of course, is to make them unguessable. *Password123* is a terrible, obvious, hackable password, and your birthday is almost as bad. If you're at all security-conscious, you try harder. But you immediately find yourself striking compromises – between things hackers are unlikely to guess and passwords you stand some fighting chance of remembering.

Your use your childhood dog's name, or the makes of cars you once owned, or your first pop concert. But those aren't *random* passwords; they refer to some antecedent in your life. (Please quit stepping up for those "fun" quizzes on social media asking you to identify your first car or pop concert, by the way. No good can come of this.) The cleverest passwords you can reasonably recall – say, *FidoPlusJeep!* – are far from random. A properly informed eye, belonging to someone who knows something about you, can discern patterns embedded in there.

Randall Munroe, the physicist who authors the epic science- and math-themed webcomic xkcd, says, "Through 20 years of effort, we've successfully trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess." Say you're (wisely) unsatisfied with *FidoPlusJeep!* as a password and try to make it more random, ergo stronger and more secure. But you want it to stick in your own mind, too. So you replace certain vowels like so:

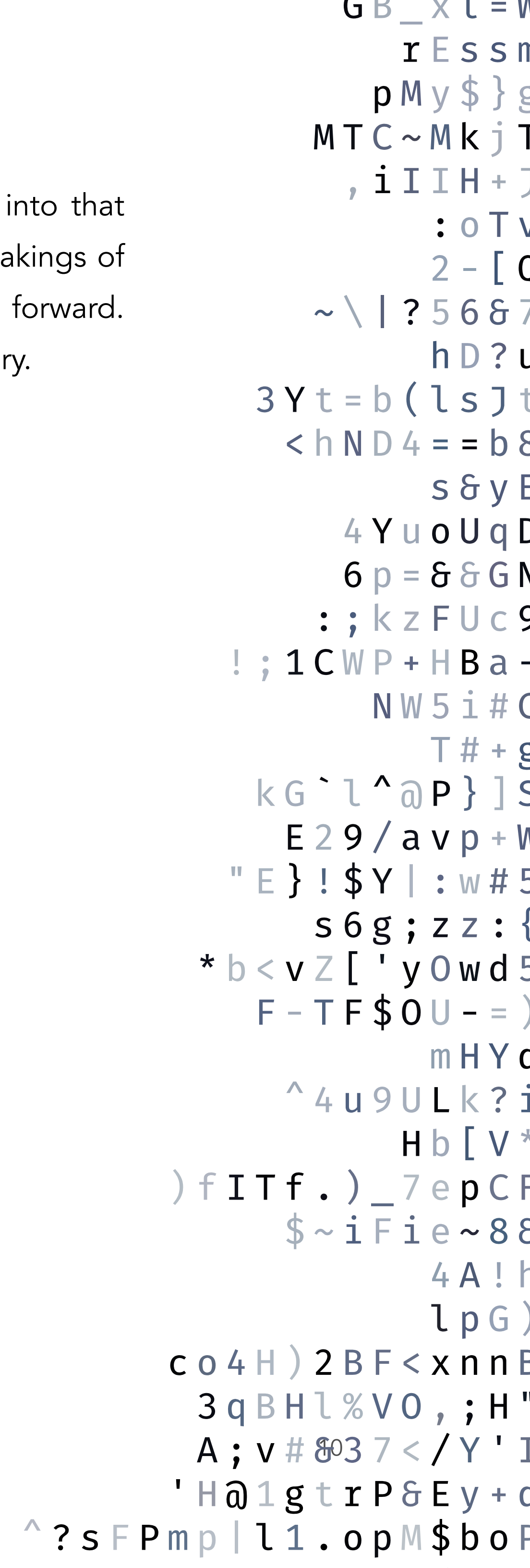
F1doPl3sJe@p!

That's only three characters changed out of 13, though, and the remnants of your dog and vehicle ownership history still lurk in there. Not that hard for an attacker to guess, and in the meantime harder for you to remember, too. It's actually a lose-lose, and the security problem is too little randomness.[1]

To circle back to our theme: not enough entropy. *F1doPl3sJe@p!* is a low-entropy barrier to viewing your secrets. More entropy, more unpredictable randomness and (ideally) zero pattern recognition, means more effective security. If you really want to do better than *Password123*, you'd go for a patternless string of characters as lengthy as you're allowed to enter: *#gnzis310%mq!veal89bjkl;6[hubcap%* is a pretty good password, with high entropy. But keeping track of a hundred or more like that, not to mention dutifully changing them every 60 or 90 days, will drive most mortals to distraction.

The password puzzle is a serviceable metaphor for the entropy challenge in modern cryptography. We require a large, steady supply of high-entropy, super-random, pattern-free cryptographic keys. But the more demand materializes for keys, the harder it is to achieve that standard at scale, and the greater the potential for compromised security. Entropy, the essential chaos ingredient in effective digital secret-keeping, needs a QA campaign.

In a page or two, we'll dig into that problem and point to the makings of a technology-centric path forward. But first a little drive-by history.



There’s Always Leakage

You’ll recall we began with images of sputtering campfires, melting ice cubes, and hot car hoods. What do those manifestations of entropy have to do with digital security?

Random, chance-fueled disorder is as old as nature itself. Scientists first named and claimed entropy only about 150 years ago – only yesterday from a big-picture standpoint, but well before the advent of modern electronic computing, or electronic anything for that matter.

At its birth, entropy was a label for a freshly understood thermodynamic phenomenon. In the 1850s and '60s, French engineer Sadi Carnot[2] and Austrian physicist Rudolf Clausius deduced that no matter how efficient an engine you construct, you'll never transform heat into mechanical action with pure, total, 100-percent efficiency. Waste is a standard, unavoidable byproduct of physical energy transfer. In other words, there's always leakage, and the longer an isolated system pounds away producing something, the more leakage you're going to see. Entropy increases as a physical system continues working, and, in the case of a campfire or pitcher of ice, eventually degenerates. (Thus did Clausius formulate the Second Law of Thermodynamics in 1865. It remains conventional wisdom to this day in high school physics classrooms.[3])

Almost immediately, though, the term was kind of hijacked to stand for much more than waste. What established pattern does a dying campfire follow? None. How many sparks and embers emanate each minute, and where are they going to land? Nobody knows. You can eyeball the fire and estimate, well, it'll probably be out by midnight – but the specific events comprising its degeneration are well and truly unpredictable. As physicists considered this, they appropriated the term entropy to describe not just waste, but unruly physical forces: chaos. Random factors cluttering the landscape of predictable, law-abiding physics. J. Willard Gibbs, the nineteenth-century American theoretical physicist, called entropy "mixedupness."[4]

Debate still crackles today over how big a concept entropy is, and how many meanings it has. Yes, there's a little intellectual chaos over the concept of scientific chaos. In the late 1940s, John von Neumann, pioneer of the computer age, advised communication theorist Claude Shannon to employ the term liberally whenever discussing information, because: "no one knows what entropy really is, so in a debate you will always have the advantage."[5]

Entropy remains an elastic term:

- In thermodynamics - hat tip to Rudolf Clausius - it still means waste and deterioration.
- In astrophysics, entropy means mystery. Black holes are assigned a level of entropy representing information they are thought to be concealing, their interior and contents being unobservable.[6]
- In sociology or MBA coursework, it's the idea that unchecked disorder increases over time; given long enough, an unmanaged club, factory, or classroom will degenerate into chaos. *Lord of the Flies* depicted a microcosm of social entropy.
- In information science, which is where we sit, entropy is a measure of the randomness or uncertainty of a given variable or system. The more random it is, the more entropy is present. Perhaps ironically, Claude Shannon invented a formula to calculate such randomess with some precision; the Shannon entropy calculator is still a pillar of modern information theory, though it may not be unbudgeable, as we shall see.[7]

So, in its 170-year definitional lifetime, entropy has evolved – from a scourge, a nemesis of engine designers everywhere, to, in the digital sphere, a security tool we can scarcely contemplate functioning without.

Of course, we go to great lengths in normal, everyday life to squeegee randomness out of most systems. We design and reward predictability. A train service that departs at random times makes no friends; a TV show that airs whenever the station engineer feels like transmitting it will attract no loyal viewers. Manifesting as chaos, entropy in the physical sphere is still mostly unloved.

But if our regular, ordered world these days seems to stagger from a surplus of entropy – more random difficulties than we want – the world of cryptography isn't always getting the quality of entropy it needs.

Less Random Than They Look

Think again about those numerical cryptographic keys, core components of both symmetric and asymmetric encryption systems. Insufficient entropy can make them vulnerable, and if our keys are vulnerable, so are we. Is our everyday entropy good enough to drive genuine randomness in key generation? Really formidable keys that betray no buried patterns and defy codebreaking, even as the global population of keys continues growing?

The U.S. National Institute of Standards and Technology (NIST) registers what for the government qualifies as moderate alarm:

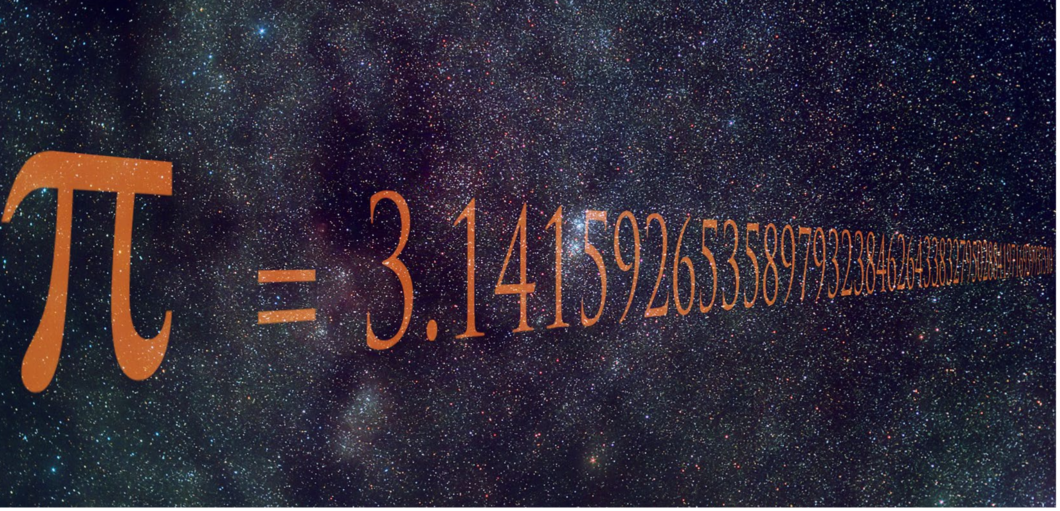
> "The importance of obtaining and using highly unpredictable keys is not just an academic question. There are many practical examples showing that failure to obtain sufficient entropy destroys any security provided by long keys and sound algorithms. Even the best algorithms cannot compensate for weak keys generated using sufficient entropy. Such systems are vulnerable to attackers – with potentially disastrous results. A study revealed that thousands of network devices had generated easily guessable keys because of insufficient entropy from the entropy source used."[8]

The notion that there are *grades* of randomness in cryptography – better and lesser breeds of entropy – is a new one. But it's valid.

What makes a "weak key"? Just as a master whisky distiller will tell you the product in his bottle is only as good as the quality of its water source[9], in cryptography, it's largely about the sources and methods you tap to seed your (ideally completely random) bit strings. As they say in journalism, consider the source.

A key-generation process that leverages repeating, patternistic seed sources, however faintly those patterns may show up and however much scanning and analysis it takes to expose them, is second-rate and represents greater security risk. Consider:

- Sometimes keys are generated from seeds of prime numbers. A prime number is one that only be divided by 1 and itself. Easily divisible numbers like 10 and 16 aren't invited to the prime party, but 13 and 17 can walk right in.



- Then there are irrational numbers, which are less neat – far less prone to concealing patterns. You can't express an irrational number as a fraction built out of whole numbers, that is, two integers stacked up, numerator over denominator. 5, a nice round prime number, is rational: you can render it as 10/2. 3.14, schoolboy shorthand for π, is rational. But π itself, which in numeral form stretches on forever (3.14159265358979323846264338327950288419716939937510 etc.)? π is firmly, proudly irrational.

A key generation process is suboptimal if it's using seed data that might be repeating itself, however subtly, however much work detection might require. Prime number series can be patternistic, and provide fewer potential values to work with, too. You can multiply primes for a dose of added complexity, a tactic used by the RSA asymmetric-encryption algorithm, but then they can become vulnerable to factorization. Given the option, then, we want to leverage seeds of irrational numbers to generate better keys. They're naturally more capable of producing random, not-repeating results.

As NIST points out, cryptographic applications demand a higher grade of randomness:

> "Random numbers are used in other scientific and engineering fields, but the goals and needs are different ... For many simulation uses, the only requirement on the seed is that it does not repeat or, equivalently, that it is unique across runs of the simulation. ... In cryptographic applications the requirements for seeding a DRBG [deterministic random bit generator, or more simply a number generator] are much more precise and stringent. The seed must possess sufficient entropy ..."[10]

There are a lot of ways to generate numbers that *appear*, on the surface, to possess a reassuring level of entropy.

Because we need so many keys, and ever-bigger ones, it's common to resort to random number generators, or RNGs. To keep abreast of demand *and* maintain a complexity edge in the face of ingenious hackers, RNGs (see also TRNGs, true random number generators, and PRNGs, pseudorandom number generators) resort to some fairly standard gambits. For one, they've steadily increased the bit size of their output, from 512, to 1024, to 2048 – on and on. (At a blackjack table in Vegas, you'll note the house does much the same thing, upping the number of decks in play in order to reduce your odds of guessing the next card to be turned up. Entropy keeps casinos in the black.)

There are hardware-based and software-based RNGs, and some hardware-based ones use pretty creative ways to manufacture entropy by tapping external inputs. A simple local hardware-based RNG might monitor and interpret a computer user's erratic mouse movements or keystrokes. It might monitor thermal noise, or the random on-off patterns of computer cooling fans. More ambitious hardware-based RNGs harvest random atmospheric sounds, or signals from radioactive decay.

But flawed RNGs are commonplace. Some of their physical entropy sources can produce outputs that are only outwardly random. Their product can turn out, upon stringent analysis, to be biased or correlatable. Some values may be more likely to occur than others. Throughput can be limited, implementations can be expensive, and hardware can be vulnerable to tampering and side channel attacks – which exploit knowledge of system characteristics such as processing time or power consumption. An interloper might thwart hardware-based RNGs by extrapolating patterns from hard drive behavior or CPU clock speed variations.[11] In those cases you run the risk of entropy deficits of the sort NIST is worried about.

Software-based RNGs, on the other hand, can offer certain advantages; they are less costly and demanding, and easier to deploy. (More about software-based RNGs in a moment.)

In any case, with soaring demand for data protection plus certain new threats on the horizon, it is time to take a new, sterner look at the quality of our cryptographic keys – which essentially requires us to elevate and scrutinize entropy factors. In addition to favoring better RNGs and deemphasizing use of prime numbers as seeds for key strings, we've got to raise our entropy game.

The Quantum Hazard

A not-quite-totally random, lower-entropy cryptographic key creates security risks. It is not that anyone is going to sit down with a number-two pencil and crack it like today's five-letter Wordle challenge. The risk is owing to modern accelerants in the codebreaking game, namely advances in higher mathematics and quantum, or exascale, computing. It's important to realize that, like their virtuous counterparts in digital security, cyber saboteurs have virtually equal access to such mighty – and potentially destabilizing – resources and will readily deploy them against less hardened targets.

Though the exact date of arrival is not clear, quantum computing is on the way. Standard computers encode data in simple, binary, zero-and-one bits, but with quantum computers, you get upgraded to *qubits*, or quantum bits, which are mind-bogglingly more capacious and complex. As with much of quantum mechanics, the dynamics of quantum computing defy breezy description. Suffice it to say it's proving hard to stabilize enough qubits for consistent task performance, but when that's figured out, as it inevitably will be, it's likely to change the security game. Unleashed against cryptographic keys with insufficient protective entropy, quantum computing horsepower may not be elegant or discreet, but a so-called "brute force" assault might still be effective. (Recall that we made an example of a low-entropy password featuring a small subset of replaced characters within a familiar word-frame. Imagine some illiterate yet fast, tireless codebreaker testing every alternative character, in every combination. In a quantum environment they could hit pay dirt before you finish reading this sentence. Cryptography keys are obviously longer and more complex, but the principle of attack is the same.) The chances of success, which of course translates as failure if you are pro-security, grow steadily with the passage of time.

Merge these new levels of quantum force with ever-larger qubit counts plus established threats to status quo encryption such as Shor's algorithm, and one scenario is a world of hurt. (Invented by American mathematician Peter Shor, this algorithm performs exponentially faster large-integer factoring. Standard cryptography proceeds from the assumption that factoring number strings composed of more than one thousand numerals is a pragmatic impossibility. Well, goodbye to all that.) In this brave new world, factoring giant prime numbers gets increasingly feasible.

An algorithm like the RSA public key cipher, whose secret security sauce is multiplying prime values and using the result to crank out – that is, seed – ever-larger keys? In the long run, it's probably waging a losing battle.

To be fair, current conventional wisdom is split over the true danger of "brute force" attacks. A few years ago Mohit Arora, a systems engineer at Freescale Semiconductor, calculated the time needed for a supercomputer to crack a 128-bit AES algorithm at one billion billion years – somewhat longer than the universe itself has been around.[12] But a less assured assessment came from researchers at MIT and the National University of Ireland (NUI). They concluded our old notions of sufficient entropy, notions that date back to the mid-20th century and Claude Shannon's celebrated entropy calculator, are overdue for an overhaul.

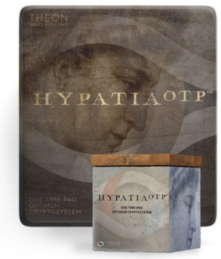
Brute force decryption is "still exponentially hard, but it's exponentially easier than we thought," said NUI's Ken Duffy – because, his team reasoned, while the orthodox idea of sufficient entropy is based on statistical averages, we really ought to be worrying about worst-case unexpected outcomes. A hacker only needs to nail down one single correlation between the encrypted and unencrypted versions of a file to have a basis for making more correlations, and things unravel fast.[13]

An objective security risk assessment by a conscientious CIO or CISO would do well to weigh such scenarios – and think about the best available countermeasure.

.

The Theon Technology Contribution

The Theon Technology cryptography solution is a step forward: It produces more random, less vulnerable keys with intrinsically better entropy. The solution is comprised of:



- **Archimedes**, a software-based RNG employing irrational numbers, and
- **HypatiaOTP**, an advanced cryptographic library that processes Archimedes' numerical seeds into more secure private keys for use in a symmetric encryption-decryption environment.

To achieve a new margin of entropy quality, truly random and free of embedded patterns, **Archimedes** taps into the mantissas of irrational numbers – that is, everything to the right of the decimal point; the mantissa is the 141592... part of 3.141592, or π. **HypatiaOTP**, meanwhile, is a software-based OTP (one-time pad) implementation – OTP being the acknowledged gold standard among modern symmetric encryption techniques. It delivers an innovative reduced-volume key transmission protocol that mitigates the challenges presented by very large private keys.

Together, **Archimedes** and **HypatiaOTP** present a powerful rejoinder to the entropy crisis: software-based generation of high-entropy, quantum-resistant private keys, performing at scale with speed and economy. Theon Technology thereby sets a new standard for software-based cryptography, moving the world closer toward the goals of quantum-proof encryption and perfect secrecy for businesses.

Readers may request white papers detailing the capabilities of Archimedes and HypatiaOTP directly from Theon. Contact channels appear after the conclusion.

Conclusion

The rise in the strategic importance in cryptography of quality entropy should be accompanied by two additional factors. First, the digital world would benefit from a more discriminating marketplace, where IT managers and business decision makers lend increasing focus to the mechanics of key generation; standard-issue randomness is no longer accepted as "good enough."

Second, emerging standards and ratings should serve as reference points for the market. Even though the quality of entropy is becoming a tier-one cybersecurity issue, there is still a distinct lack of standards and governance surrounding entropy. The ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) are working to set benchmarks for RNGs,[14] but if we are transitioning to an era of quantitative entropy standards and governance, progress is uncomfortably slow.

The digital world is moving faster than the regulators. Yet the market is predictably fragmented and multilateral -- a free-for-all. A high number of cryptography vendors are bringing niche approaches to market amid insufficient appreciation for the quality factors that distinguish best-in-class solutions. This leaves thoughtful students of security on their own to make the best empirical judgments they can.

We depend on effective cryptography. Cryptography depends on entropy – initially a half-defined, half-abstract concept of disorder; in the everyday physical world a chaos factor that, with a bad card or infernal dice roll, prevents you from hitting 21 at blackjack or scoring a Yahtzee; but today a key ingredient and ally in digital security – one that must be refined if computer security is to keep advancing.

Random numbers have become essential pillars of the security framework that holds society together. As an aid to keeping secrets, we want not just bulk shipments of entropy, but the highest-quality entropy possible. Randomness, in the end, is how we keep our secrets. It's imperative that we keep advancing the state of entropy play.

Contact Theon Technology
info@theontechnology.com

Or visit
theontechnology.com

ENDNOTES

[1] Randall Munroe, xkcd #936, "Password Strength." https://xkcd.com/936/. We're indebted to Munroe for inspiring the discussion of password entropy in this text.

[2] Kausik H. Manish, "A Brief History of Entropy: Part I," Towards Data Science, 15 August 2020. https://towardsdatascience.com/a-brief-history-of-entropy-chapter-1-9a2f1bc0d6de

[3] Jim Lucas, "What is the Second Law of Thermodynamics?" LiveScience, 22 May 2015. https://www.livescience.com/50941-second-law-thermodynamics.html

[4] Brig Klyce, "Cosmic Ancestry: The Second Law of Thermodynamics," Panspermia.org. https://www.panspermia.org/seconlaw.htm

[5] Ibid.

[6] "Beckenstein-Hawking Entropy," Scholarpedia, 2017. http://www.scholarpedia.org/article/Bekenstein-Hawking_entropy

[7] Try it out at https://www.shannonentropy.netmark.pl/.

[8] NIST, "True Randomness Can't Be Left to Chance: Why Entropy is Important for Information Security," 2012. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=915121

[9] Jake Emen, "Is Water the Most Important Ingredient in Liquor?" Eater.com, 25 November 2015. https://www.eater.com/drinks/2015/11/25/9784954/spirits-water-purity-water-sources

[10] NIST, "True Randomness Can't Be Left to Chance: Why Entropy is Important for Information Security," 2012. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=915121

[11] Unitychain Core Team, "Security in Chaos and Entropy: True Random Number Generators," Unitychain.io blog. https://www.unitychain.io/blog/true-random-number-generators/

[12] Mohit Arora, "How Secure is AES Against Brute Force Attacks?" EE Times, 7 May 2012. https://www.eetimes.com/how-secure-is-aes-against-brute-force-attacks/

[13] Larry Hardesty, "Encryption is Less Secure Than We Thought," MIT News Office, 14 August 2013. https://news.mit.edu/2013/encryption-is-less-secure-than-we-thought-0814

[14] ISO/IEC, Information technology — Security techniques — Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408, ISO Online Browsing Platform. https://www.iso.org/obp/ui/#iso:std:iso-iec:20543:dis:ed-1:v1:en