Why Massive IoT Needs True Randomness for a Secure 5G Network

Author: Vladislav Petkov

February, 2018

Introduction

The importance of secure data encryption continuously increases with every device that joins a public network. In the emerging IoT these will exceed the usual computers, laptops, tablets or smartphones, adding many new targets and ports for malevolent parties to steal our private information. As a major channel for data transfer between the users on the IoT, 5G networks will be a prime target for attacks. Therefore, the implementation of safe and cost and energy efficient encryption protocols on the 5G grid is fundamentally important.

With the new variety of devices that join the IoT there will be a multitude of new access points where a malevolent party may try to hack a system. At the same time a successful attack on a central server or transfer node will affect far much more end users. In view of the future development of technology one should look for long lasting solutions.

In the history of cryptography there has been a constant competition of inventing more secure and effective encryption algorithms and the development of more powerful computing tools that can be used to break the current codes. The nature of the data transmission has also changed the rules.

In the second half of the twentieth century the development of computers and the need for secure exchange of data between users, who can communicate only over public channels, has made all usual deterministic encryption methods virtually unusable. Today every algorithm relies on some random choice to create a public or a private key, which is used to protect our data.

Encryption security and pseudo random seeds

Today's encryption protocols share one essential flaw - their security level assessment starts with the assumption that a purely random seed has been generated at the start of the process. If there were some inherent bias for the distribution of this seed, whether accidental or intentionally planted by a malevolent party, there might be a very significant difference between the theoretical prediction and the actual level of security. The lack of such bias is called the entropy of the seed's generation [1].

Computer chips and processors are deterministic by nature and thus most commonly the seed used to create a security key or a hash function is picked not by a True Random Number Generator (TRNG), but by a Pseudo Random Number Generator(PRNG). Although PRNGs try to imitate a true random distribution and produce bits with high entropy, historically it has been observed that sometimes they fall very far from this goal [2].

The aim of classical cryptanalysis is to crack our secret information by intercepting our data over a public channel and deducing our encryption key. The strength of an algorithm is evaluated by how hard it is to find the key, even if large quantities of data have been intercepted. Viable encryption algorithms are always proposed with a quantifiable measurement of security, usually pointing that their security may not be breached with the development of technology in the foreseeable future. Some are even developing effective algorithms that are safe against the theoretical quantum processors [4].

An alternative approach to cracking a code is not to attack the already encrypted message, but to generate a large set of decryption keys using the same method and to try if one of them works. This will circumvent any type of security proposed by the later steps in the encryption algorithm. Private keys are generated from a very large pool of numbers and ideally the entropy of the random seed is high enough to guarantee that the probability of picking the same key is virtually zero. However, if the entropy is low the number of

tries necessary to crack the key is significantly smaller.

The importance of true randomness in data security for the IoT

Precise measurment of entropy in a process is much harder than knowing the time effectivness of a cryptography algorithm, yet guaranteeing sufficient randomness is essential for data security [3]. Thus, it is necessary to analyze the PRNG used to produce this entropy and seek its possible flaws. To avoid historical weaknesses more advanced PRNGs are invented to better imitate a TRNG. However, this is once more a race between finding a temporary solution and the development of better ways to attack this inherent fault of PRNG-relying protocols.

On the contrary, if a TRNG is used for encryption this weakness is resolved once and for all. The strength of the data security guaranteed by the cryptography algorithms will be correctly estimated. This is even more essential for the devices on the IoT. Most cryptography algorithms are flexible, in the sense that we can pick the optimal length of a security key and exchange protocol to balance our need of high security and cost and energy efficiency of the data transfer. Some of the devices on the IoT will not have anything close to the processing power of an ordinary computer, thus we cannot blindly rely on strengthening the data encryption in the usual way, that requires more resources.

Efficiency is central and to achieve it we must have correct concept of the level of security guaranteed. The use of a TRNG removes one unknown and hard to predict variable in this problem.

The development of of TRNGs has made great progress in the last few years. It has moved away from seemingly random inputs like processor clock, mouse motion or keyboards, which do not provide sufficient entropy. Some of the modern TRNG utilize physical processes with high randomness, such as radioactive decay [5], random dice rolls [6, 7] or atmospheric noise [8]. With the spread of such TRNG large pools of random bits may be produced reliably and distributed to a variety of devices at low costs.

Why is now the time to change 5G to use TRNG

With the vast amount of expected data transferred over 5G between millions of users and a multitude of devices, it is critical to be able to monitor the stress on the network as well as the security of the data. Users demand to exchange larger packets of data with more and more sources at higher speed and smaller latency, while at the same time they want to be sure that their data will be protected at every point. Finding the balance between users' preferences and the network energy and maintenance costs is hard enough today. In the age of the IoT it will be much harder.

Encryption algorithms and protocols may easily be updated on a device years after its production. In contrast a source of real random seeds that would be used by these algorithms cannot be safely installed at a later point. It is usually imprinted on the hardware during the production process. For example a reliable source of high entropy may be integrated within a device as a "system on a chip". Even today some producers of microchips implement this architecture [9]. In this way any password created on the device will have the actual security guaranteed by the cryptography algorithm.

Many of the devices on the IoT will have poor entropy sources of their own, yet they may present equal gates for malware to the network. In such cases it is a viable idea to provide the device directly with a number of randomly generated seeds, which it may use in turn to securely communicate over the network. In cases when a TRNG is not available locally a central server may offer the users with a pool of purely

random bits, that may be updated frequently. Thus the security of each device will be as strong as that of the server.

With a TRNG used from the beginning it will be possible to transparently inform the users on the 5G network about the real strength of their data protection. Commitment to such transparency and foresight will essentially rely on accessible and reliable TRNGs that can produce random numbers on the massive scale demanded by the IoT.

References

[1] Vassilev, A., "The Importance of Entropy to Information Security", Computer (Volume: 47, Issue: 2, Feb. 2014)

[2] Dodis, Yevgeniy, David Pointcheval, Sylvain Ruhault, Damien Vergnaud, and Daniel Wichs. "Security Analysis of Pseudo-Random Number Generators with Input: /dev/random is not Robust." Cryptology ePrint. (2013). Web. 12 Dec. 2013. http://eprint.iacr.org/ 2013/338.pdf

[3] Vassilev, A., "Entropy as a Service: Unlocking Cryptography's Full Potential", Computer (Volume: 49, Issue: 9, Sept. 2016)

[4] SecureRF, https://www.securerf.com

[5] NIST Randomness Beacon, National Institute of Standards and Technology, Computer Security Division, https://beacon.nist.gov/home

[6] Real Random, LLC, https://www.realrandom.co/.

[7] NIST Entropy as a Service initiative, National Institute of Standards and Technology, Computer Security Resource Center, https://csrc.nist.gov/projects/entropy-as-a-service

[8] Random.org, https://www.random.org/

[9] Arm Holdings, https://www.arm.com/products/processors