The Struggle for Secure WiFi in the Era of the IoT and the Weakness of Pseudorandom Passwords

Author: Vladislav Petkov

April, 2018

Introduction

The importance of WiFi access for the majority of citizens in the modern world can hardly be overestimated. The expanding provision of internet access over the past two decades has turned something that was considered a luxurious whim into a profound dependence. In a sense, WiFi has changed our own culture. Whether it is for work, study, news reading or entertainment, people use devices connected to the internet more frequently and universally than ever. We expect to find an available network everywhere - at airports and train stations, in restaurants, libraries and malls, even inside subway cars. A whole culture of Internet cafes has risen. Cities and even countries are ranked over the spread and quality of free WiFi they have.

Whether we are obsessed with following our friends on social networks, streaming a video while riding on a train or occasionally checking for the next arriving bus we cannot resist using a WiFi service when it is available. Unfortunately, we are also commonly willing to compromise our security in doing so, by signing in to unprotected networks and impatiently accepting the disclosure agreement without reading it. In case our device is contaminated with a malevolent software it can spread quickly to other devices we connect to.

While five years ago this could mean that we unintentionally compromise our desktop while sending an email from our smartphone, today in the emerging era of the Internet of Things (IoT) the contamination might reach many more devices in our possession. Adding to the danger of our private files or bank information stolen or erased from our computer, it is possible for someone to unlock our front door or turn on our smart oven when we are not at home. Perhaps these new dangers will make us more conscious about what networks we use. However, even in that case, what guarantees can we have that a network is indeed secure?

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 the data we exchange or the devices we access.

Security of a WiFi network in the IoT

The expected increase in number of devices on the internet that we will be accessing very commonly over a WiFi network will tremendously increase the points through which the network may be attacked. We have to emphasize that it is possible that our devices are a danger for the network as much as the network is for them. Further, this is not a trouble we may have only when we are logging to "a shady" WiFi network. Take for example the more and more common access to WiFi during flights on many airlines. This is definitely a type of network you would expect to be up to modern standards of encryption security. Nevertheless, there is a case where a passenger claimed that his computer was hacked during the flight by a fellow passenger[11]. Now imagine how somebody may unwillingly spread a virus over the airplane WiFi, because thirty minutes earlier they checked whether they had left the stove on as they were leaving for the airport.

This example is not accidental. Airports are a place where thousands of people access the internet everyday as well as a place where we desire a strong level of security both for the network and for the customers. Thus it is essential to make sure that these networks are protected as well as we can.

Our dependence on reliable and free WiFi everywhere is constantly increasing and people are more likely

to use it to connect to more devices in the future. The increased number of users and data traffic raises the importance of security. One way to ensure it for the safety of the network and the comfort of the customer is to make sure that the predicted strength of the encryption is indeed true, because we have used a truly random key.

Encryption algorithms and random keys

With advancing technology and increasing channels through which we may be attacked, providing protection from malevolent parties is becoming harder. Fortunately for all of us, cryptography is up to the challenge. Not only are new algorithms constantly being developed and weaknesses being addressed. Very often these algorithms are made to challenge the conceptual power of future threats. For example the foundations of elliptic curve cryptography (ECC) were set approximately 30 years ago. Cryptographers proved mathematically that algorithms using ECC are superior in security and efficiency than the ones which rely on the so called Discrete Logarithm Problem (DLP). Yet it was only into the second half of the 2000s when the implementation of ECC became more widespread[1]. This is only because until then the predicted security level of the old algorithms was adequate for the capability of the existing technology that might attack them.

Modern encryption algorithms may utilize a variety of mathematical problems, however, they are expected to provide a clear estimate to their effectiveness and security. They are patented to provide certain level of robustness against classical cryptanalisys even if we consider the possible development of new technology in the foreseeable future. Some are even developing effective algorithms that are safe against the theoretical quantum processors [2].

On the other hand, almost all 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. The level of randomness of this seed is called entropy and is very important for the security of the generated key [3]. Private keys are picked from a very large pool of numbers and ideally the randomness of their choice ensures us that guessing the key is statistically impossible. This is of course if we had generated the key with significant entropy. Using a powerful encryption method, while generating a key with low entropy, can be compared to setting your bank account password to be your first name.

Measuring entropy and random number generators

If entropy is essential, how can we measure it or guarantee that it is up to standards? How do we create randomness?

Computer processors and microchips are deterministic by nature and thus usually the seed used to create a private 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 [4].

Precise measurment of the entropy produced by a certain PRNG is much harder than proving the security level of a cryptography algorithm, yet guaranteeing sufficient randomness is essential for data security [5]. PRNGs have evolved when important weaknesses have been discovered. Initially the computer processor used to pick a random number by referring to the its internal clock at the time of request. Since people soon discovered that this does not imitate sufficiently well a uniformly distributed random variable, newer

PRNGs were proposed. However, working on new PRNGs, that imitate a TRNG better, 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.

Another obstacle is the new type of devices that will need protection in the world of IoT. In a modern smart home a refrigerator that is accessible from the internet may have a microchip of power bigger than the Voyager 2 spacecraft, yet the entropy it may produce is not proportionally high enough to guarantee its protection should we access it trough a remote and insecure network and contaminate it with a malware. The lower expected power of many of the devices on the IoT relative to the improved desktops available to the malevolent parties, must change our perception on fluidly improving encryption security. We cannot rely on the standard ways of increasing security, because they will require more resources and on the IoT there is no longer even a relative equality of processing power. Efficiency is central and to achieve it we must have a 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 strength of the data security guaranteed by the cryptography algorithms can be correctly estimated.

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 [6], random dice rolls [7, 8] or atmospheric noise [9]. 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.

One way to implement the so generated random seeds is to install them on the hardware during manufacturing. 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 [10]. In this way any password created on the device will have the actual security guaranteed by the cryptography algorithm. Alternatively, the random seeds may be pooled from a certified source that generates and changes them with time. This is not a radical novelty - after all commonly when we create a username to some website a temporary password is generated for us.

References

[1] Joppe W. Bos, J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig1, and Eric Wustrow, "Elliptic Curve Cryptography in Practice", Microsoft Research TechReport, MSR-TR-2013-119.

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

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

[4] 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

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

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

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

[8] 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

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

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

[11] Time magazine: "Reporter Says He Was Hacked on a Flight", http://time.com/4237280/reporter-says-he-was-hacked-on-a-flight/.