The Performance of Blum-Blum-Shub Elliptic Curve Pseudorandom Number Generator as WiFi Protected Access 2 Security Key Generator

Challiz D. Omorog
Technological Institute of the Philippines
Quezon City, Philippines
+639284846955
challizomorog@cspc.edu.ph

Bobby D. Gerardo
Western Visayas State University
IloIlo City, Philippines
+639209291848
bobby.gerardo@gmail.com

Ruji P. Medina
Technological Institute of the Philippines
Quezon City, Philippines
+6329110964
ruji_p_medina@yahoo.com

ABSTRACT

WiFi Protected Access 2 (WPA2) is considered the most secure network security protocol in wireless routers, despite the discovery of partial key exposure vulnerability. In light of, an experiment was conducted to investigate the strength of WPA2 default passwords generated by the algorithms embedded in routers using a simulated brute-force attack. The findings ascertain the prevalence of insecurities in the default WPA2 passwords due to low charset size and weak encryption algorithm. For these reasons, we propose Blum-Blum-Shub Elliptic Curve Pseudorandom Number Generator (BBS-ECPRNG) algorithm as a replacement to the algorithms embedded in routers. To prove its validity, we generated distinct sequences of 10^6 bits each and analyzed sequence output using the NIST statistical test suite. The generated bit sequence of BBS-ECPRNG was converted to password characters and subjected to simulation test. Findings reveal that the BBS-ECPRNG password significantly decreased the password-cracking success by 25 times more as compared to the default WPA2 passwords generated by router-based algorithms in the Philippine market.

CCS Concepts

• **Computing methodologies → Modeling and simulation → Simulation evaluation** • **Security and privacy → Network security → Mobile and wireless security**

Keywords

Blum-Blum-Shub, Elliptic Curve, Pseudorandom Number Generator, Password strength, WiFi, WPA2, Brute-force.

1. INTRODUCTION

Wireless Local Area Network (WLAN) popularly known as WiFi stands as the basic standard for wireless communication today [33]. Since WiFi networks transmit connection over radio frequency technology [11] [41], it too inherent deficient characteristics particularly security vulnerabilities from its predecessor [32]. Due to its broadcast nature, (encrypted) traffic is easily intercepted making it susceptible to attacks such as eavesdropping and jamming [44].

The exposed issues involving WiFi [15] [3] [13] [24], however, did not constraint the demand for ubiquitous connectivity. In fact, most people want to be online all the time [23] making WiFi access points (APs) known as “hotspots”, a commonplace. As predicted by CISCO, WiFi will account 63% of the Internet traffic by 2021 [5]. Meanwhile, cybercriminal activities are reported to increase sharply [6] including identity theft, financial heists and computer hacking [35], inadvertently fostered by poor security practices and behavior [38], and weak understanding of Internet security and its implications [36] [39] [43]. When everything appears secure and proper, Internet users rarely consider security. Consequently making possibilities for Internet security risks and exploits virtually limitless.

Currently, IEEE 802.11i or WPA2 is the current security standard mechanism that encrypts traffic on WiFi networks to thwart attackers [30] [40]. The security strength of WPA2 Personal mode, designated for small office and home office (SOHO) networks, gets its authentication component from the plaintext or WiFi password[1] [37] [17]. In this mode, the user must provide a match WiFi key to the router to get connected [10]. By default, WiFi keys are composed of seemingly complicated characters generated by an algorithm pre-configured in the router.

According to Tripwire, 46% of SOHO users do not change default configuration or won’t bother to read the AP manual to change the WiFi password [2]. Since WiFi traffic is easily sniffed or intercepted and given the risk posed by the unsecured practices of SOHO users, a strong password [4] [14] can slow or defeat router attacks such as dictionary and brute force methods [12] [8]. However, WiFi key generation algorithms in routers received less research attention whereas underground hacker websites, videos and blog posts flourish reports about cracking WPA2-Pre Shared Key (PSK) password. Despite knowledge to the contrary, current researches related to WPA2 are more concentrated on four-way handshake authentication/ deauthentication [7], encryption [1], and frames [18].

ICBIM’18, September 20–22, 2018, Barcelona, Spain.

ACM ISBN 978-1-4503-6545-1/18/09 …$15.00.
DOI: http://doi.org/10.1145/3278252.3278262

[1] From here onward, the terms password and key(s) will be used interchangebly, which both refer to the security information required to access WiFi connection intended to be secret to two or more entities.

In this paper, we analyze the WPA2 random key generation algorithms currently embedded in wireless routers and demonstrate its insecurity. Next, we present a novel approach to increase WiFi password randomness using the proposed algorithm called BBS-ECPRNG. Then, we evaluate the performance of BBS-ECPRNG as WPA2 security key generator through simulation using penetration-testing tools in different platforms.

2. RELATED WORK

2.1 WPA2 Authentication

As illustrated by the blue line in Figure 1, without transmitting the WiFi password or the red key over the air, both the client and the AP independently attempts to prove that each side knows the WiFi password by each generating the Pair-Wise Master Key (PMK). The PMK is the hash result of the network Service Set Identification (SSID) and the WiFi password represented by the blue and violet keys for the AP and client, respectively. The PMK is transmitted and decrypted on each side [19]. Once WiFi password is authenticated, WPA2-PSK generation starts or authentication dance called “4-way handshake” is performed [10].

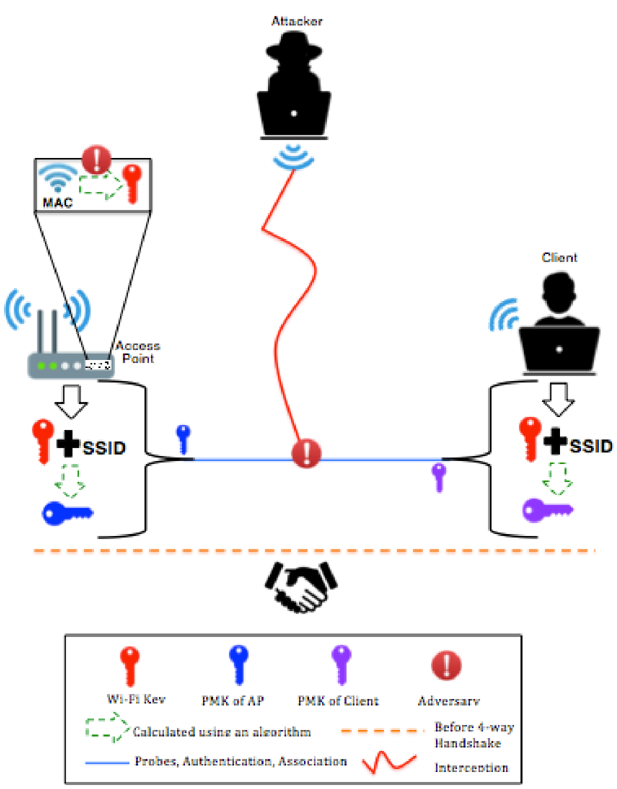

Figure 1. WPA2 Authentication Workflow

On the other hand, generating the WiFi password is done from the router side as depicted by the green arrow in broken lines inside the small box in Figure 1. The pre-configured WiFi password is generated using a pseudorandom key generator, which usually takes it seed from the MAC address of the router device. Several recent reports [16] [21] [27] [20] across the country recounted that this practice exhibits failure to utilize the PRNG effectively, attributed by the low entropy or poorly chosen seed [28] and weak or predictable algorithm [31] .

2.2 Router-based PRNGs

The article of Tsitroulis et al. [40] details the unprecedented findings on how the flaw in WPA2 protocol could be exploited by malicious attacks based on partial PMK exposure vulnerability. The article successfully exposed WPA2 security issues and demonstrates the weaknesses in detail. Also, Lorente et al. [21] initiated on how to reverse-engineer wireless routers to identify the password generating algorithms embedded on each. The study reveals that massively deployed routers use weak algorithms to generate the default WiFi passwords. Several techniques and approaches identified in the article are calculated in several ways to produce series of characters from eight (8) to 12 characters long such as (1) decrementing one from the last digit of the MAC address, (2) substituting and moving each pair of the MAC address in iteration, (3) adding an exclusive-or (XOR) operation in the algorithm, or (4) using the MAC address or a combination of Internet Service Provider (ISP) name plus a random seven digit number. Given all these possibilities, the attacker can easily manage a dynamic analysis to recover the algorithm and in a matter of seconds generate the WiFi password then carry out security assaults. Yet, access to these cheap routers pre-equipped with weak generating password algorithm in the market is common and still currently distributed by large ISPs in many countries.

2.3 WPA2 Password Threats and Security Countermeasures

There are several ways to crack passwords such as algorithm analysis, brute-force attack, dictionary attack, rainbow attack, etc. But brute-force attack is one of the widely used methods of fully guesting password using a random approach[9]. According to Yasin and AbuAlrub [42], the best way to counter or decrease the possible rate of success of brute-force attacks is to understand its attack mechanism. Brute-force calculates every possible combination of ASCII charset size and password length that could be included in a password, which means short, simple and predictable combinations are quickly cracked in milliseconds.

2.4 Pseudorandom Number Generator

Pseudorandom number generator (PRNG) is designed to generate randomness. But as the word “pseudo” in PRNG suggests, it has the appearance of randomness, but eventually generates a predictable sequence of patterns. PRNGs are merely mathematical equations that use a seed value to generate random numbers, thus, at some point will cycle inherent from its deterministic source- the seed [22].

The structure of the raw PRNG has two stages: (1) the random number generator, and (2) sequence generator. The random number generator unit is given an initial value s_0 called seed. Typically, the user provides the seed as fixed or constant secret values. Then computed by some function produces “random” bit sequence called pseudorandom number sequence. Meanwhile, to calculate for the next seed s_n, the sequence generator increments the current state s_{n+1} then begins the next iteration until a full cycle called the period is repeated.

The mathematics behind PRNG evidently suggests that the sequence output can never exceed the entropy of its seed, which implies a short periodic sequence. However, if it would take hundreds of thousands of years for advanced computers to computationally repeat a period [26] or the PRNG is statistically provable [34], then we can safely assume that it is practically secure.

3. THE PROPOSED BBS-ECPRNG

The proposed generator implements the intractability of DLP and integer factorization problem (IFP) into one algorithm called BBS-ECPRNG. Full details and discussion about the proposed algorithm (concept diagram, notations, and algorithm) is published in [25]. The algorithm is written using Python legacy (Python 2.7) to generate the desired pseudorandom bit sequence.

4. METHODOLOGY

4.1 BBS-ECPRNG Code Implementation

The implementation of the proposed algorithm in Python is divided into three (3) major files:

4.1.1 BBSECPRNG Generator (BBSEC.py)

This file generates two (2) elliptic curve points then outputs the random sequence (in bits and characters).

4.1.2 Randomness Tests (randtest.py)

This file contains the 15 different NIST statistical tests downloaded from https://gerhardt.ch/random.php.

4.1.3 The Main Program (main.py)

This file contains the working interface that employs the BBSEC.py to generate the desired sequence and bit sequence length then subsequently tests the output sequence by calling the rantests.py program. The results are compiled in a CSV file.

4.2 WPA2 Password Sampling

This paper investigated the WPA2 password strength embedded in routers/ pocket wifi that are massively deployed in the Philippines. Three (3) of the largest ISPs in the country (codename IS1, IS2, and IS3) are covered in this comparison. These ISP companies directly supply their customers with wireless Internet routers/ pocket WiFi. During the experiment, we used one specific router/ pocket WiFi model for each ISP per operating system (OS). IS1 used AN5506-04, IS2 used Huawei B315s, and IS3 used ZTE MF65M model.

The generated bit sequence of BBS-ECPRNG is firstly subjected to statistical analysis using the 15 statistical tests recommended by NIST [29]. The BBS-ECPRNG full statistical results can found in [25]. Then the generated bit sequence of the algorithm is converted to eight (8) password characters using character sets ([a-z; A-Z] and [0-9]). Here, the approach is to extract the desired character length from the output bit sequence using binary to character conversion technique. We specify the most significant bit as the starting bit sequence, skipping non-ASCII or special characters during conversion. After generating the sample password (BE), we performed the brute-force attacks. BE was used as the new router password for IS1along each OS.

4.3 Simulation Plan and Penetration Testing Tools

In order to expose the strength of the WiFi passwords, Brute-force attack was employed. The attack has been performed in a Linux OS using Parrot Security 4.1. This security-testing platform is packed with a multitude of security and penetration testing tools such as Crunch and Aircrack software (https://www.parrotsec.org/). These tools were used to gain access to the network, and capture necessary packets to recover the keys.

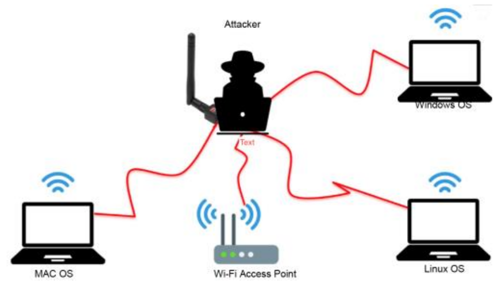

Figure 2. Network simulation plan

Figure 2 shows the network simulation plan comprised of a wireless router, attacker and three (3) laptop units of the same hardware specification. Each unit has different OS installed such as Windows (U1), Linux (U2), and MAC (U3) and all are connected using the WPA2 protocol. U1 used an Intel PRO wireless card, U2 used an Airport Extreme wireless card, and U3 used an Atheros AR9285 wireless network adapter. During the experiment, the AP, represented by the router, was approximately 10 meters away from the laptops in an open space. The WLAN's SSID was 'SOMEID' while the network's IP4 address was 192.168.175.0/24.

During the simulation test, attacks were executed in a shell script environment, represented by attacker with hat icon. Each of the platforms was penetrated using Aircrack-ng. In this experiment, we tested the WiFi password performance generated by the routers distributed by the three (3) largest ISP companies in the Philippines. Attack time (in hours) was recorded for every attack performed on each platform per ISP router. The simulation performed one trial on each platform per ISP's router.

5. RESULTS AND DISCUSSION

5.1 BBS-ECPRNG Sample Output

In order to generate the random bit sequence, the BBS-ECPRNG algorithm was coded in Python script in OSx. The user can enter the number of sequences, and the desired length for the bits and characters. The sample Python output is shown in Figure 3.

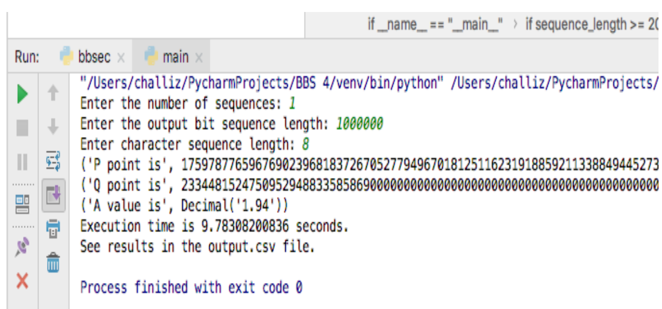

Figure 3. BBS-ECPRNG Python console output

5.2 Simulated Brute-Force Attack

Twelve experimental cases have been subjected to brute-force attacks. Each scenario have been examined and categorized into two (2) sets. The first set (case 1-9) used the default password embedded in the ISP routers running on the given OS, while the second set (case 10-12) used the BBS-ECPRNG generated password entered in the ISP router (IS1) running on the given sample of OS. The experimental cases are shown in Table 1.

Table 1. Password strength vs. Brute-Fore attack

Case	Legal User	Charset Size	Crack Time (hr)
1	IS1 on U1	(a-z, 0-9)	9.2
2	IS1 on U2	(a-z, 0-9)	9.6
3	IS1 on U3	(a-z, 0-9)	9.3
4	IS2 on U1	(A-Z, 0-9)	24.3
5	IS2 on U2	(A-Z, 0-9)	24.9
6	IS2 on U3	(A-Z, 0-9)	24.65
7	IS3 on U1	(0-9)	12.4
8	IS3 on U2	(0-9)	12.48
9	IS3 on U3	(0-9)	12.42
10	BE on IS1 on U1	(A-Z, a-z, 0-9)	604.8
11	BE on IS1 on U2	(A-Z, a-z, 0-9)	616.8
12	BE on IS1 on U3	(A-Z, a-z, 0-9)	607.2

As demonstrated in the last column, the keys were cracked easily in cases 1-3 compared to the result in cases 4-9. However, cases 10-12 were 25 times more difficult to crack than cases 4-6 regardless of the OS. This is because the number of ASCII character set size used in forming the WiFi passphrase in cases 10-12 is larger, compared to the other cases as seen in the third column, which indicates increase password security as reported in [12].

In the cases of IS1 and IS2, these routers may have used more charset size than IS3 for the default password, however, appears to follow a fixed pattern or composition that lowers its level of security and resistance to brute-force attacks [14]. One reason for this is the algorithm included right inside the router, which may have used a fixed string format starting from the Wi-Fi MAC address that Tsitroulis et al. [40] exposed.

It is also observed that U1 has the highest crack time as compared to its competitors. Mainly since different OS are impacted to differing degrees depending on how they implement the WPA2 protocol. However, when the BE generated password was used as the default password for U1, it exhibited an impressive attack performance result. This implies that a quick security improvement in routers could be achieved by simply replacing the weak password-generating algorithm by a statistically provable and secure key generator such as BBS-ECPRNG.

6. CONCLUSIONS

As previously discussed, the strength of BBS and ECPRNGs lie on the intractability of the IFP and the DLP, respectively. For these reasons, we propose BBS-ECPRNG algorithm as alternative to the algorithms embedded in routers. To prove its validity, distinct sequences of 10^6 bits each were generated, and tested based on the NIST statistical test suite standard. The test validates that the proposed BBS-ECPRNG is provably secure and statistically generates randomness, which is essential before practical applicability in various cryptographic applications.

This paper also ascertains the prevalence of insecurities in the default WPA2 passwords in routers particularly distributed in the Philippines. We propose the BBS-ECPRNG algorithm against router-based algorithms. BBS-ECPRNG produces truly random, unpredictable composition of WPA2 passwords that can slow or significantly decrease password-cracking success by 25 times more as compared to default WPA2 passwords generated by router-based algorithms in the market.

7. REFERENCES

[1] Mayank Agarwal, Santosh Biswas, and Sukumar Nandi. 2015. Advanced stealth man-in-the-middle attack in wpa2 encrypted wi-fi networks. *IEEE Commun. Lett.* 19, 4 (2015), 581–584. DOI:https://doi.org/10.1109/LCOMM.2015.2400443

[2] Aloul, F. A. 2010. Information security awareness in UAE: A survey paper. *Int. Conf. Internet Technol. Secur. Trans.* June (2010), 1–6.

[3] Frank Breitinger and Claudia Nickel. 2010. User Survey on Phone Security and Usage. *Biosig* May 2010 (2010), 139–144.

[4] Sonia Chiasson, Alain Forget, Robert Biddle, and P C van Oorschot. 2008. Influencing users towards better passwords: Persuasive Cued Click-Points. *Proc. 22nd Br. HCI Gr. Annu. Conf. People Comput. Cult. Creat. Interact.* (2008), 121–130. Retrieved from http://dl.acm.org/citation.cfm?id=1531514.1531531

[5] Cisco. 2015. Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. *Forecast Methodol.* (2015), 22. DOI:https://doi.org/1465272001663118

[6] Alisdair Faulkner. 2017. *ThreatMetrix Cybercrime Report Q1 2017*. Retrieved from https://www.threatmetrix.com/wp-content/uploads/2017/05/cybercrime-2017-q1-1493750698.pdf?_ga=2.49956064.716730045.1497157484-948371628.1497157484

[7] Ghanem, M. C. and Ratnayake, D. N. 2016. Enhancing WPA2-PSK four-way handshaking after re-authentication to deal with de-authentication followed by brute-force attack a novel re-authentication protocol. In *2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA)*, 1–7. DOI:https://doi.org/10.1109/CyberSA.2016.7503286

[8] Sarah Granger. 2002. The Simplest Security: A Guide To Better Password Practices. *Symantec Connect*. Retrieved from http://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices

[9] Aaron L-F Han, Derek F Wong, and Lidia S Chao. 2014. Password Cracking and Countermeasures in Computer Security: A Survey. *arXiv Prepr. arXiv1411.7803* (2014). DOI:https://doi.org/10.13140/2.1.2652.8329

[10] Changhua He and John C Mitchell. 2004. Analysis of the 802 . 11i 4-Way Handshake. *WiSe '04 Proc. 3rd ACM Work. Wirel. Secur.* (2004), 43–50.

[11] Anthony C Ijeh, Allan J Brimicombe, David S Preston, and Chris O Imafidon. 2009. Security Measures in Wired and Wireless Networks. (2009), 113–121.

[12] Philip G Inglesant and M Angela Sasse. 2010. The true cost of unusable password policies. *Proc. 28th Int. Conf. Hum. factors Comput. Syst. - CHI '10* (2010), 383. DOI:https://doi.org/10.1145/1753326.1753384

[13] Lazaridis Ioannis, Pouros Sotirios, and Veloudis Simeon. 2013. Vulnerability issues on research in WLAN encryption algorithms WEP WPA / WPA2 Personal. (2013), 40–46.

[14] Markus Jakobsson and Mayank Dhiman. 2013. The Benefits of Understanding Passwords. *Mob. Authentication SE - 2* (2013), 5–24. DOI:https://doi.org/10.1007/978-1-4614-4878-5_2

[15] Fh Katz. 2012. WPA vs. WPA2: Is WPA2 Really an Improvement on WPA? *2010 4th Annu. Comput. Secur. ...* (2012), 1–4. Retrieved from http://distributed-wpa-cracking.googlecode.com/svn-history/r306/trunk/papers/wpa_vs_wpa2.pdf

[16] Dušan Klinec and Miroslav Svítok. 2016. UPC UBEE EVW3226 WPA2 Password Reverse Engineering, rev 3. Retrieved July 1, 2017 from https://deadcode.me/blog/2016/07/01/UPC-UBEE-EVW3226-WPA2-Reversing.html

[17] Umesh Kumar and Sapna Gambhir. 2014. A literature review of security threats to wireless networks. *Int. J. Futur. Gener. Commun. Netw.* 7, 4 (2014), 25–34. DOI:https://doi.org/10.14257/ijfgcn.2014.7.4.03

[18] Fabian Lanze, a Panchenko, Benjamin Braatz, and Thomas Engel. 2014. Letting the puss in boots sweat: detecting fake access points using dependency of clock skews on temperature. *Proc. 9th ACM ...* (2014), 3–14. DOI:https://doi.org/10.1145/2590296.2590333

[19] Latha, P. H. 2014. Review of Existing Security Protocols Techniques and their Performance Analysis in WLAN. *Int. J. Emerg. Technol.* Comput. *Appl. Sci. (IJETCAS)* (2014), 162–171.

[20] Jakob Lell and Jörg Schneider. 2012. Insecure default WPA2 passphrase in multiple Belkin wireless routers. 7–8. Retrieved from https://www.agrs.tu-berlin.de/v_menue/advisories/wpa_default_passphrase/

[21] Eduardo Novella Lorente, Carlo Meijer, and Roel Verdult. 2015. Scrutinizing WPA2 Password Generating Algorithms in Wireless Routers. In *Proceedings of the 9th USENIX Conference on Offensive Technologies* (WOOT'15), 10. Retrieved from http://dl.acm.org/citation.cfm?id=2831211.2831221

[22] Volodymyr Lynnyk and Noboru Sakamoto. 2015. Pseudo random number generator based on the generalized Lorenz chaotic system. *Int. Fed. Autom. Control* (2015), 257–261. DOI:https://doi.org/10.1016/j.ifacol.2015.11.046

[23] Arati Mejdal. 2014. Making the Most of Social Media. *Chance* 27, 4 (2014), 28–30. DOI:https://doi.org/10.1080/09332480.2014.988953

[24] Mehdi Nasiri Noroozani and Hamid Reza Ebrahimi. 2014. The Study of Wpa and Wpa2 Algorithms in Wifi Technology. 3, (2014), 167–169.

[25] Ruji P. Medina. Bobby D. Gerardo Omorog, Challiz D. 2018. Enhanced Pseudorandom Number Generator based on Blum-Blum-Shub and Elliptic Curves. In *Proceedings - 2018 IEEE Symposium on Computer Applications and Industrial Electronics.*

[26] Christof Paar and Jan Pelzl. 2010. *Understanding Cryptography*. Springer-Verlag. DOI:https://doi.org/DOI 10.1007/978-3-642-04101-3

[27] Roberto Paleari and Alessandro Di Pinto. 2013. Multiple vulnerabilities on Sitecom devices. Retrieved from http://blog.emaze.net/2013/08/multiple-vulnerabilities-on-sitecom.html

[28] David P. Rosin. 2015. *Dynamics of Complex Autonomous Boolean Networks*. Springer International Publishing, Cham. DOI:https://doi.org/10.1007/978-3-319-13578-6

[29] Andrew Rukhin, Juan Soto, James Nechvatal, Smid Miles, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. 2010. A statistical test suite for random and pseudorandom number generators for cryptographic applications. *Natl. Inst. Stand. Technol.* 800, April (2010), 131. DOI:https://doi.org/10.6028/NIST.SP.800-22r1a

[30] Jessica Scarpati. 2009. Wireless security protocols: The difference between WEP, WPA, WPA2. 1–2. Retrieved from http://searchnetworking.techtarget.com/feature/Wireless-encryption-basics-Understanding-WEP-WPA-and-WPA2

[31] Schiller, J. and Crocker, S. 2005. Randomness Requirements for Security. DOI:https://doi.org/10.1007/978-94-017-0377-2_6

[32] Frederick T. Sheldon, John Mark Weber, Seong Moo Yoo, and W. David Pan. 2012. The insecurity of wireless networks. *IEEE Secur. Priv.* 10, 4 (2012), 54–61. DOI:https://doi.org/10.1109/MSP.2012.60

[33] Gagandeep Singh and Sukhvir Singh. 2014. IEEE 802 . 11 WLAN and Advancements : A Review. 3, 2 (2014), 32–39.

[34] Sobol' I. M., and Levitan, Y. L. 1999. A pseudo-random number generator for personal computers. *Comput. Math. with Appl.* 37, 4–5 (1999), 33–40. DOI:https://doi.org/10.1016/S0898-1221(99)00057-7

[35] Symantec Corporation. 2013. *Internet Security Threat Report 2013*. DOI:https://doi.org/10.1007/s10207-014-0262-9

[36] Talib, S., Clarke, N., and Furnell, S. 2010. An Analysis of Information Security Awareness within Home and Work Environments. In *International Conference on Availability, Reliability, and Security (ARES)*, 196–203. DOI:https://doi.org/10.1109/ARES.2010.27

[37] Dejan Tepsic, Mladen Veinović, and Dejan Uljarević. 2014. Performance evaluation of WPA2 security protocol in modern wireless networks. *Proc. 1st Int. Sci. Conf. - Sint. 2014* (2014), 600–605. DOI:https://doi.org/10.15308/sinteza-2014-600-605

[38] Hugh Thompson. 2013. The Human Element of Information Security. *IEEE Secur. Priv.* 11, 1 (January 2013), 32–35. DOI:https://doi.org/10.1109/MSP.2012.161

[39] Nik Thompson, Tanya Jane McGill, and Xuequn Wang. 2017. "Security begins at home": Determinants of home computer and mobile device security behavior. *Comput. Secur.* 70, (September 2017), 376–391. DOI:https://doi.org/10.1016/j.cose.2017.07.003

[40] Achilleas Tsitroulis, Dimitris Lampoudis, and Emmanuel Tsekleves. 2014. Exposing WPA2 security protocol vulnerabilities. *Int. J. Inf. Comput. Secur.* 6, 1 (2014), 93. DOI:https://doi.org/10.1504/IJICS.2014.059797

[41] Vandana Wekhande. 2006. Wi-Fi Technology : Security Issues. *Rivier Acad. J.* 2, 2 (2006), 1–17.

[42] Adwan Yasin and Fadi AbuAlrub. 2016. Enhance RFID Security Against Brute Force Attack Based on Password Strength and Markov Model. *Int. J. Netw. Secur. Its Appl.* 8, 5 (2016), 19–38. DOI:https://doi.org/10.5121/ijnsa.2016.8402

[43] Chen Zhang and Janet J Prichard. 2009. an Empirical Study of Cyber Security Perceptions , Awareness and Practice. *Issues Inf. Syst.* 10, 2 (2009), 242–248.

[44] Yulong Zou, Jia Zhu, Xianbin Wang, and Lajos Hanzo. 2016. A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. *Proc. IEEE* 104, 9 (2016), 1727–1765. DOI:https://doi.org/10.1109/JPROC.2016.2558521