Oct 26th, 3:30 PM - 5:00 PM

Vulnerability of passwords consisting of Numerical Repetitive Sequences in the WPA2 protocol

Genc Kelmendi
*University for Business and Technology - UBT, genckelmendi@gmail.com*

Edmond Hajrizi
*University of Business and Technology, ehajrizi@ubt-uni.net*

Follow this and additional works at: [https://knowledgecenter.ubt-uni.net/conference](https://knowledgecenter.ubt-uni.net/conference)

**Recommended Citation**
[https://knowledgecenter.ubt-uni.net/conference/2019/events/281](https://knowledgecenter.ubt-uni.net/conference/2019/events/281)

This Event is brought to you for free and open access by the Publication and Journals at UBT Knowledge Center. It has been accepted for inclusion in UBT International Conference by an authorized administrator of UBT Knowledge Center. For more information, please contact knowledge.center@ubt-uni.net.
**Vulnerability of passwords consisting of Numerical Repetitive Sequences in the WPA2 protocol**

Gene Gregor Kelmendi, Edmond Hajrizi

UBT – Higher Education Institution, Lagjja Kalabria, 10000 p.n., Prishtine, Kosovo

**Abstract.** Protocols that govern wireless security WPA2/WPA have been proven much more secure in comparison to their predecessor WEP. However, the human factor jeopardizes the solidity of cryptography by implementing passwords consisting of programmatically predictable numerical structures such as 1234512345, 11114444, 999888777 and so on. The methods presented in this paper are effective in decrypting such passwords within seconds using ordinary processor power. The prevalence of this vulnerable practice in the Prishtina (Kosovo) region is estimated to be 15.7% in 89 tested wireless routers. Under this study such types of passwords are termed and referred to as Numerical Repetitive Sequences or NRS. The paper defines NRS mathematically, identifies NRS types, composes formulas for calculating variations, presents algorithms to generate NRS, and proposes tools to implement attacks. The methods documented in this study should only be used for educational purposes.

**Keywords:** wireless password, network security, mathematical sequences, arithmetic progression, handshake attack

**Introduction**

Among the additional security that WPA/WPA2 protocols provided compared to the WEP predecessor, was the minimum password length requirement of 8 characters. This was intended to condition users to implement passwords more resistant to brute-force attacks. But by not validating the type of inserted characters, a security hole is left where the password can consist of repeatable characters and sequences, and therefore rendering them predictable. The length criterion alone cannot effectively avoid unsafe practices. A password consisting of 8 characters of the digit 5 is as unsafe as a password consisting of only one character of number 5. Similarly, a password of type “11223344” is as unsafe as a 4 character password of “1234”. Character replication or repetition can be easily simulated programmatically, and therefore does not increase the entropy and strength of the password. Passwords consisting of numbers only are particularly vulnerable to brute-force attacks. This is because the decimal base of numbers consist of only 10 elements (0-9), and the variations are significantly smaller when compared to alphabets which have a richer amount of elements. When these numeric passwords are also limited to certain templates, they even further endanger the security of the system. For example, passwords 1234512345, 11114444, or 999888777 are not only numerical by nature, but also suffer from replications and fixed progressions (incremental or decremental). Unfortunately, such passwords are often applied in the wireless domain. After taking a closer look, we realize that these types of passwords are essentially arithmetic sequences that are replicated for a given number of times, therefore for an easier
reference they will be referred to as Numerical Repetitive Sequences (NRS). In this paper NRSs will be mathematically defined, algorithmically generated, and implemented as a dictionary attack against the captured WPA handshake.

**NRS Definition**

Numerical Repetitive Sequences (NRS) is the numerical string of the length $l$ that replicates the finite arithmetic sequence of the form:

$$\{a_i\}_{i=1}^n with a_{i-1} + d$$

Where the first term $a_1$ and the upper limit $l$ meet the conditions

$$1 \leq n \leq 10; \text{Mod}(l,n) = 0$$

While distance $d$ is defined as

$$a_1, l, n, d \in \mathbb{N}^0 \mid 0 \leq a_1 \leq 9$$

Such as $d = 0$ if and only if $n = 1$, generating constant arithmetic sequence

$$d = 0 \iff n = 1$$

If $n < l$ arithmetic sequence or each term is replicated $l/n$ times successively, thus producing the numerical repetitive sequence with length $l$.

**Types of NRS**

**Definition 1.** If each term is replicated separately $l/n$ times, the sequence is labeled as internal sequence, and symbolically is denoted as $S_I$.

**Definition 2.** If entire sequence is replicated $l/n$, the sequence is labeled as external sequence, and symbolically is denoted as $S_E$.

**Definition 3.** The S sequence is a singular sequence if $n = l$ and is symbolically denoted as $S_S$. The singular sequence is neither internal nor external. Since the upper bound is equal to the length of the sequence $n = l$, neither sequence nor terms can be replicated.

**Definition 4.** The S sequence is a constant sequence if $d = 0$ and is symbolically denoted as $S_0$. The constant sequence is both an internal and an external sequence at the same time. Since $d$ is 0, it follows from equation (5) that the string has only one term. Thus since $\{a\} = a1$ the whole sequence and the corresponding terms are repeated $l$ times, satisfying the definitions 1 and 2. Primes greater than 10 have only constant sequences.

**Definition 5.** The S sequence is an incremental sequence if $d > 0$ is symbolically denoted as $S_+$. 

**Definition 6.** The S sequence is a decremental sequence if $d < 0$ is symbolically denoted as $S_-$. 

**Examples**
Analysis and Variations Formulas for NRS

Various lengths of sequences determine different values of variables $n, d$ and $a_1$ thus producing different variations for every length and consequently disables the application of classic formulas for calculating variations. In this analysis we will compile formulas that calculate separately the variations of constant, singular, and internal/external sequences, and summing the total sum at the end.

**Formulas for calculating variations of constant sequences**

Each length $l$ has 10 constant sequences because each number is integer with 1. It follows that for each length there are 10 constant variations, a variation for each value 0-9 replicated as long as the sequence is long, and this can be marked as:

$$|S_E| = 10 \quad (9)$$

Length consisting of prime numbers greater than 10 only have constant sequences,
which means their variation is always 10, as demonstrated by Example 5 and the following proof.

Proof. By definition prime numbers have no other factors except 1 and themselves. According to equation (2) in Chapter 2, the NRS definition limits the upper limit not greater than or equal to 10, consequently the only allowed upper limit for prime numbers remains to be number 1. When the upper limit is 1, the distance is 0 and it follows that the sequence is always constant with an amount of 10 variations.

**Formulas for calculating variations of singular sequences**

Given the equation (2) n cannot be greater than 10 and also the sequence is singular only if l is equal to n, then it logically follows that lengths greater than 10 have no singular sequences. Following formula calculates variations of singular sequences:

\[ |S_0| = 2(11 - l)^7 \quad |l| \geq 6 \quad (10) \]

From this formula we note that if l is greater than 10 the result will be negative (or zero), and the function denoted by the symbol + gets only the positive part, as it is also illustrated in the equation (16), where it follows that the set of singular sequences for the respective length is an empty set.

\[ |S_0| = \emptyset \iff l > 10 \quad (11) \]

**Formulas for calculating variations of internal and external sequences**

Values of n produce fixed variations regardless of the length of the sequence, the length simply determines how many times those variations need to be replicated. Thus, e.g. the upper limit n = 4 produces 48 internal and external sequences, even when the length l is 8 or 12 or any other length that is divisible to the limit n = 4. This demonstrates that NRS passwords are unsafe regardless of their length, and increasing the length does not effectively enrich the variation pool. Table 1 illustrates the variations given for each value of variables n with the potential to produce internal or external sequences.

<table>
<thead>
<tr>
<th>n</th>
<th>(S_I)</th>
<th>(S_E)</th>
<th>(S_T)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>135</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

The formula that calculates the internal and external variations for a given n is

\[ |S_{IE}| = 4 \left[ \sum_{d=1}^{n} 10 - d(n-1) \right] \quad (12) \]

Where k is the maximum distance supported by the given n, as defined by equation (4).
Total variations of NRS for a given length

The formula that calculates the total variations for a given length is:

\[
\bar{S}_r = \sum_{n \geq 1} \left[ 4 \sum_{l=1}^{n} 10 - d(n-1) \right] + 2(11-l)^+ + 10 \quad \forall \ n \mid l
\]

Example 4. Calculate the total variations for length 9.

\[
|S| = 4[10 - 1(3 - 1) + 10 - 2(3 - 1) + 10 - 3(3 - 1) + 10 - 4(3 - 1)] + 2(11 - 9)^+ + 10
\]
\[
= 4[10 - 2 + 10 - 4 + 10 - 6 + 10 - 8] + 2(2)^+ + 10
\]
\[
= 4[8 + 6 + 4 + 2] + 4 + 10
\]
\[
= 94
\]

Explanation 1. The formula conditions the upper limit to be divisible with \( l \). Thus, the factors of 9 are \{1,3,9\}. The first sigma starts from number 2 to \( l - 1 \) which is 8. From 2 to 8 is only one number that is proportional to 9, and it is 3. For upper limit 3 the second sigma is applied starting at distance 1 and is increased up to 4. Any increment greater than 4 it returns an empty set. For example, if \( k \) is 5 then \( 10 - 5(3 - 1) = 0 \). In this manner \( k = 4 \). After calculating the Sigma’s, the singular variations 4 and the constant variations 10 are added.

Example 5. Calculate the total variations for length 29.

\[
|S| = 2(11 - 29)^+ + 10
\]
\[
= 2(-18)^+ + 10
\]
\[
= 2(0) + 10
\]
\[
= 10
\]

Explanation 2. The formula conditions the upper limit to be a factor of \( l \). But, number 29 being a prime number does not have any factor within the definition of sigma (within the range \( n \geq 2 \) and \( l - 1 \)). As such first sigma does not apply, and consequently neither the second. The last part of the formula is replaced with actual values and is calculated. Since the result is negative –18, only the positive part is obtained which is 0 by using the following function from the reference [8]:

\[
f^+ = \frac{|f| + f}{2}
\]

Algorithm for the Generation of NRS List

The following is a function of the C# language that returns the entire NRS list for a given length:
Methodology of Attack

The methodology for decrypting NRS passwords of WPA and WPA2 wireless router consists of the following steps:

a) Generate the list of NRSs for lengths of 8 to 20 and is stored in a text file.
b) The victim wireless handshake is captured using the wifite tool.
c) The handshake is attacked by importing the text file in the pyrit

d) If the handshake is decrypted using one of the passwords generated as NRS, the attack will be considered successful.
The SPN password in Figure 1 successfully broken is 123123123. This is a password of length 9, with distance 1, external replication of value 3, initial term 1 and upper limit 3. We also note that 887 passwords were tried per second. The pyrit tool has completed the entire NRS list with 2090 combinations in 2.4 seconds. The CPU processor used for these attacks is the Core2 Duo P8600 @ 2.4 Ghz. Although it is a relatively old 2009 processor with a sub-average processor power it has managed to try NRS passwords up to 20 characters within 2-3 seconds.

The Prevalence of the Vulnerability

Statistics in the following table 2 are based on 89 handshakes captured in 5 different locations. Handshakes are tried to be decrypted with NRS list of the length from 8 to 20 with a total variation pool of 2090 passwords. Out of the 89 tested wirelesses, 14 NRS passwords were successfully extracted, accounting for approximately 16% of all wireless tested
The frequency of successfully decrypted NRS passwords is as follows: 12345678 (6 times); 123456789 (2 times); 0000011111 (1 time); 22223333 (1 time); 77778888 (1 time); 123123123 (1 time); 12341234 (1 time) and 777888999 (1 time).

<table>
<thead>
<tr>
<th>Loc. 1</th>
<th>Loc. 2</th>
<th>Loc. 3</th>
<th>Loc. 4</th>
<th>Loc. 5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount of handshakes</td>
<td>25</td>
<td>9</td>
<td>14</td>
<td>6</td>
<td>35</td>
</tr>
<tr>
<td>Decrypted</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Failed</td>
<td>22</td>
<td>9</td>
<td>12</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>Success rate</td>
<td>12%</td>
<td>0%</td>
<td>14.3%</td>
<td>50%</td>
<td>17.1%</td>
</tr>
</tbody>
</table>

Conclusion

The efficiency of NRS list is very high in the wireless domain. With a variation of only 2090 passwords it managed to decrypt the WPA / WPA2 handshake successfully in 15.7% of the cases. Additional studies and statistics are needed to verify this success rate in other international regions.

Acknowledgment. Acknowledgments to Prof. Dr. Edmond Hajrizi, Msc. Naim Preniqi and PhD.can. Besnik Qehaja.

References

10. Nipissing University, Finite Series Tutorial, 
   [http://calculus.nipissingu.ca/tutorials/finiteseries.html].
11. Khan Academy, Explicit formulas for arithmetic sequences | Algebra (article), 
   [https://www.khanacademy.org/math/algebra/sequences/constructing-arithmetic-sequence/a/writing-explicit-formulas-for-arithmetic-sequences].
12. Wolfram|Alpha, Wolfram|Alpha Examples: Mathematics, 
   [http://www.wolframalpha.com/examples/Math.html].
13. Wikipedia, Arithmetic Progression, 
   [https://en.wikipedia.org/wiki/Arithmetic_progression].
14. Dirk Van de moortel, Permutations - Variations - Combinations, 
   [http://users.telenet.be/vdmoortel/dirk/Maths/PermVarComb.html].
15. Wikipedia, Lagrange Interpolating Polynomial, 
   [https://en.wikipedia.org/wiki/Lagrange_polynomial].
16. Wikipedia, Positive and negative parts, 
   [https://en.wikipedia.org/wiki/Positive_and_negative_parts].
17. Armend Shabani, Strukturat Diskrete