ENHANCE RFID SECURITY AGAINST BRUTE FORCE ATTACK BASED ON PASSWORD STRENGTH AND MARKOV MODEL

Dr. Adwan Yasin[1] and Fadi AbuAlrub[2]

[1] Computer Science Department, Arab American University
Jenin - Palestine
[2] Computer Science Department, Arab American University
Jenin - Palestine

ABSTRACT

RFID systems are one of the important techniques that have been used in modern technologies; these systems rely heavily on default and random passwords. Due to the increasing use of RFID in various industries, security and privacy issues should be addressed carefully as there is no efficient way to achieve security in this technology. Some active tags are low cost and basic tags cannot use standard cryptographic operations where the uses of such techniques increase the cost of these cards. This paper sheds light on the weaknesses of RFID system and identifies the threats and countermeasures of possible attacks. For the sake of this paper, an algorithm was designed to ensure and measure the strength of passwords used in the authentication process between tag and reader to enhance security in their communication and defend against brute-force attacks. Our algorithm is design by modern techniques based on entropy, password length, cardinality, Markov-model and Fuzzy Logic.

KEYWORDS

RFID, brute-force attack, Markov-model, entropy, fuzzy logic.

1. INTRODUCTION

IN order to achieve security and privacy protection in the RFID system, we studied the RFID environment concerning how it works, its key components as well as threats and countermeasures of this technology in order to determine the attacks that still need for further research. [1][2]

Due to the limited size and cost of RFID systems, commonly used encryption techniques do not meet the desired security requirements. Thus; it is important to develop a new technique that enhances the security of RFID communication.

When the password is weak, it can be broken easily by hackers by using brute-force attack. The aim of this paper is to develop a new algorithm that generates strong passwords which can withstand brute-force attacks and tests the strength of generated passwords, and to integrate this algorithm with other authentication algorithms to enhance the security of RFID communication.

DOI: 10.5121/ijnsa.2016.8402

1.1 PROBLEM STATEMENT

Although most of the focus in RFID technology is on privacy, we should be more aware about information security issues in this technology. The weakness interest in information security in RFID technology makes it easy to access sensitive information that use this technology, which is an opportunity for manipulation, and theft of critical information through eavesdrop the communication between the tag and the reader. [1][3]

Due to the growing cost of existing encryption techniques, this will add to the cost of RFID systems which makes the design of new algorithms to ensure data security a challenge, and the unsafe communication channels which provide a non-secure environment for the exchange of information between the tag and the reader, it is necessary to protect these channels with a new low-cost encryption technology. And the use of non-cryptography based authentication with a random number doesn't provide enough security. [3][5]

In this paper, we will study different RFID attacks and the countermeasures for these attacks. We will also categorize the attacks and the countermeasure actions to select one of the most dangerous attacks that are not covered completely with security controls to enhance security on RFID based on the chosen attack.

2. RFID SYSTEM OVERVIEW

2.1 RFID TECHNOLOGY

RFID stands for Radio Frequency Identification that uses radio waves to transmit the identification as a unique serial number for an object wirelessly. RFID is deemed one of the most widespread technologies that use radio waves to track, classify, detect and uniquely identify a variety of objects (i.e., merchandise, people, and assets), it doesn't need line-of-sight scanning as it uses waves. RFID uses serial number to identify the objects with its full information; it stores the object on a microchip that is connected to an antenna (this combination called tags). [1]

2.2 RFID ENVIRONMENT

RFID environment consists of the following components as shown in Figure 1: [1]

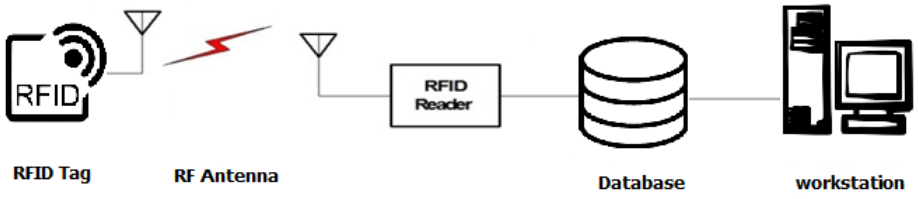

Figure 1. RFID System Components

1. **RFID Tags**

There are two main kinds of RFID Tags: [1]

- Passive Tags
- Active Tags which have larger memories up to 128 KB.[20]

2. **RFID Reader**
3. **RFID Middleware**
4. **Electronic Product Code (EPC):** A unique code for objects stored in RFID tags memory, matching the same functionality of barcode numbering scheme (UPC). This code is a 96-bit number as in Figure 2.

Figure 2. EPC Tags [21]

2.3 How Does RFID Work?

RFID tag store a unique identity code in its read-only or rewrites internal memory depending on type and application. The reader identifies the tag through the magnetic field frequencies. After authentication is done, the tag sends its unique serial and all information to the reader who, in turn, sends these data to the application side through middleware. The middleware is an interface between readers, tags, application and database. When tags send their identification to the reader and the system will match tags' code with the corresponding data existing in the database. The result determines if the next processing will be accepted or rejected as shown in Figure 3. [1]

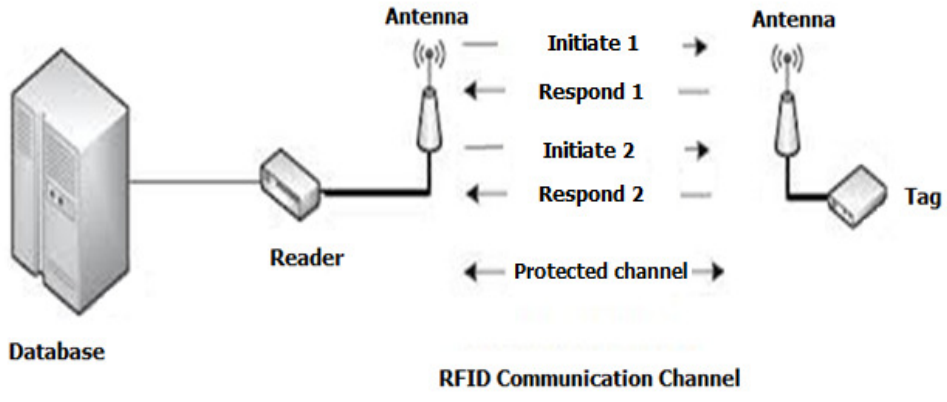

Figure 3. RFID Communications

2.4 RFID LAYERS COMMUNICATION

It's very important to understand the communication process between RFID components through the system. In order to do that, we should understand the OSI model for RFID system.

The OSI model is the model that is responsible for data communication through the system; it consists of logical layers that define the requirements of communication between tag and reader.

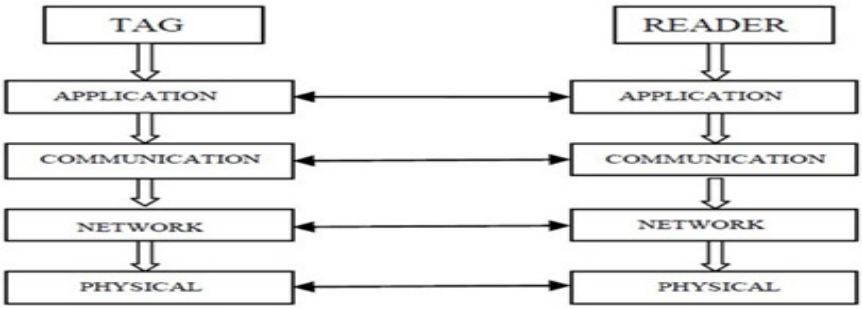

Figure 4. RFID Layers Communication [4]

3. RFID THREATS AND SECURITY ATTACKS

With the widespread usage of RFID, the big challenges to RFID applications in- creased, therefore; security and privacy protection become important. RFID applications are exposed to different types of malicious attacks ranging from passive eavesdropping to active interference. In order to understand RFID attacks; it's important to understand and summarize RFID weaknesses. This part explains and classifies the most important RFID attacks and their countermeasures. [4]

RFID threats come in these main layers: physical, network transport, application, Multilayer layer and Strategic Layer, classified based on their (integrity, confidentiality, and availability) as shown in Table 1. [7]

Table 1. Major RFID Threats [4]

Physical Layer	Network-transport layer	Application layer	Multilayer Attacks	Strategic Layer
Temporarily disabling tag and permanently disabling tag.	Eavesdropping(or Skimming)	Tag modification	Man in Middle Attack	Privacy Threats
	Spoofing		Replay	
	Cloning		Denial of Service Attack	
	Impersonation		Cryptography Attacks	
	Traffic Analysis			

4. RFID ATTACK CATEGORIZATION BY CIA

There are many attacks on RFID systems and these attacks can be classified based on integrity, availability and confidentiality as shown in Table 2.

Table 2. RFID Attack Categorization by CIA

	CIA principles		
RFID Threats	Confidentiality	Integrity	Availability
Eavesdropping Data	X		
Eavesdropping Transmission	X		
Spoofing	X	X	
Cloning	X	X	
Denial of Service			X
Tracking	X		
Impersonation		X	
Traffic Analysis	X		
Tag modification		X	
Man-in-the-middle		X	
Replay		X	
Cryptography	X	X	

5. RFID IMPACTS AND COUNTERMEASURES

Institutions put the necessary measures for achieving security and privacy. Risk management is one of the best means of calculating the risk in order to develop the necessary countermeasures for the risk prevention. Risks are calculated by assessing the threats as well as their impact on the institution, vulnerability and the likelihood. Note that the impacts of threats are related to CIA principle. In this section, we will discuss the effects of each threat and the appropriate countermeasure for each. [2]

It's important to analyze threat countermeasures when the organization determines the threats. We have already checked some other papers and calculated the number of countermeasures for each threat in order to focus our research on the threats that do not have adequate countermeasures as shown in Table 3. [2]

Reviewing the effects of RFID threats with CIA principle, we became able to determine which threat to focus on based on concerning relative importance of these threats for the institution. Depending on the results shown in Table 3, we chose cryptography attacks and focused on brute force attack due to the lack of its countermeasures. [2]

6. CRYPTOGRAPHIC ATTCAKS

In this attack, the attacker breaks these algorithms and tries to get the data that is stored in a tag. Brute force attack is mostly used in cryptography to break the encrypted algorithms.

6.1 BRUTE-FORCE ATTACK

Brute-force attack is an attack where software or tools are used to guess password and get access to sensitive data, in this attack, series of all possible passwords are sent in an

attempt to guess the used password and obtain access. To protect our data against this attack, the password used should be powerful enough. When the strength of password increased, it will take more time to guess it. In addition to using a strong password, powerful encryption can be used at the same time to provide a very high level of security for your data. [10][13]

There are many things to take into consideration in this kind of attack: a group of passwords to test, how fast the hacker can check whether potential passwords are valid or not, how long it would take the hacker to break the password as well as the possible rate of success for breaking a specific password.[11]

The feasibility of brute force relies on the domain of input characters of the password and the length of the password. Table 4 shows the number of possible passwords for a given password length and character set, as well as how long it would take to crack passwords of that type.[12]

When we create the password we use the following collections:

Numbers (0 to 9); that leaves us with 10 numbers. Characters (A-Z or a-z); 26 for upper-case letters and 26 for lower-case letters, so that the total is 52. Special Symbols (! @,., #, $, %, ,& , and more) that are about 32.

The formula used to count the number of combinations to try is: Total Combinations = $Possible\ character^{password\ length}$

Table 4. RFID Cracking Time

Password Consist of	Possible combinations	Max Time for Cracking the password
5 characters (3 lower case letters, 2 numbers)	36^5= 60,466,176	60,466,176 / 2,000,000,000 = 0.03 seconds
7 characters (1 upper case letter, 6 lower case letters)	52^7=1,028,071,702,528	1,028,071,702,528 / 2,000,000,000 = 514 seconds = approx, 9 minutes
8 characters (4 lower case letters, 2 special characters, 2 numbers)	68^8= 457,163,239,653,376	457,163,239,653,376 / 2,000,000,000 = 228,581 seconds = approx, 2,6 days
9 characters (2 upper case letters, 3 lower case letters, 2 numbers, 2 special characters)	94^9= 572,994,802,228,616,704	572,994,802,228,616,704 / 2,000,000,000 = 286,497,401 seconds = approx, 9,1 years
12 characters (3 upper case letters, 4 lower case letters, 3 special characters, 2 numbers)	94^12= 475,920,314,814,253,376,475,136	475,920,314,814,253,376,475,136 / 2,000,000,000 = 237,960,157,407,127 seconds = approx, 7,5 million years

7. RELATED WORK

The authors in paper [5] proposed an algorithm to ensure the security of authentication between RFID tag and reader without the need to use costly encryption techniques, and they are adopted in the design of the algorithm on matrix multiplication.

The algorithm assumes that each tag and reader stores a square matrix, and each of them stores one matrix and the inverse of the matrix which exist in the other end of the same size. The tag and the reader share the key K. With the knowledge that the selected key and matrices are generated randomly by both the tag and the reader. [5][6]

The algorithm description is shown in Figure 6.

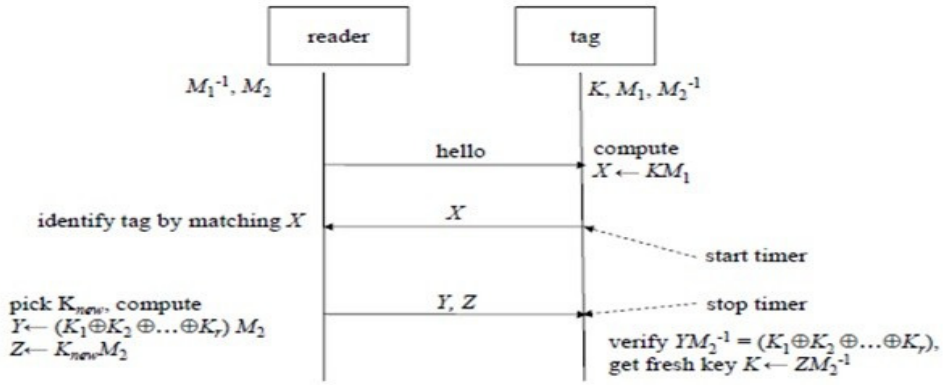

Figure 6. Secure Tag Identification Algorithms [5]

7.1 ALGORITHM LIMITATIONS

The proposed algorithm uses a key and matrix of size 24 bytes which is fairly small, and makes it possible for a hacker to guess each of the key and the matrix using a brute force attack. Generating multiple keys in each phase and store these keys in RFID tags consume their resources taking in consideration the limitation in their memory. [5][6]

The keys and matrices are generated randomly with small size and using brute-force techniques an intruder can easily guess these keys and matrices used in this algorithm. Data Security violates are strongly related to randomness. Weak random numbers and impairment in the authentication protocol allows any key of a cryptographic RFID to be found in a matter of seconds. [14]

Given that random numbers are used to secure our information, it will come as no surprise that the performance and characteristics of random number generators have a robust impact on security. To put it simply, attackers don't crack encryption, they rob or guess keys. Poor quality or insufficient quantities of random numbers have the

effect of creating that much easier, decreasing security to well below its designed level and making the overall system vulnerable. [15]

8. CONTRIBUTION

The algorithm used in paper [5] was taken and improved in this paper because it does not meet the security requirements necessary for the process of encryption between RFID tag and reader due to the use of weak key and matrices; which are small in size and lacks complexity giving by this a possibility for hackers to guess passwords and matrices using brute-force attack.

Thus, the idea here is to improve the previous algorithm by generating strong passwords according to the best practices so that they would be unbreakable and integrated into the former algorithm in order to increase its strength against brute-force attack.

The resilience of a password against brute-force attack can be determined based on three parameters cardinality, length, and entropy. The current minimum key length used in the previous algorithm is eight characters, so it should equal 32 characters or more.

And as an example by choosing a cardinality of 92 constructed as in Figure 7, and by using a password of 8 characters long, we calculated the entropy using equation number (1), and got a result of 52.4 bits entropy. This entropy measures the number of guesses the hacker need to break a password using a brute-force attack as follows: [17] [19]

$$Entropy = \log_2 C^L \qquad (1)$$

Where C is cardinality and L is password length.

8.1 WHAT MAKES FOR A SECURE RANDOM GENERATOR?

One of the important parameters to explore is entropy density. Entropy is a measure of the randomness of data. For a given throughput, lower entropy might result in keys that are less random, making them more vulnerable to hacking. [15]

8.2 ADDRESSING BRUTE-FORCE ATTACK USING STRONG PASSWORD

One of the effective methods to resist the brute-force attack is to use a strong password policy by reviewing the previous algorithm criticized earlier for using a randomly generated key and lacking strength. Strong passwords which are long and consist of symbols and letters help overcoming brute force attack. [16][6]

As there is no precise description of strong password, there are some rules to follow in order to ensure generating a strong password that can resist brute-force:

- Passwords must consist of twelve characters at least.
- Passwords should have both capital and small letters.

- Passwords must contain numeric characters.
- Passwords should include punctuation.

One of the methods used to resist brute-force attack is to delay it by increasing the number of failed attempts to break the password, with this delay we can have indications of an attack and, therefore; take the necessary measures to resist it. [16]

8.3 PASSWORD ENTROPY

Entropy is a quality indicator for passwords and high entropy can give a better quality. The entropy only establishes the boundary for the amount of guesses needed to crack the password. Thus we can estimate the number of attempts used by a hacker to guess the password. [17][19]

The entropy bits of a password measured in bits is:

- The base-2 logarithm of number of estimation to find the password.
- On average, an assailant should try half of potential passwords before finding the correct one.

For example a password that consists of 8 characters with upper and lowercase characters and numbers are given in the following equation:

$$E = log_2(62\text{^}8) = 47.6$$

By using this formula it's evident that increasing the length is more important than increasing the cardinality of a password. If we use the formula to test this by using the two passwords:"A13F=; 54d!" and"IhaveAOldHorse123". The first password, if we use Table 5, will have a cardinality of 92 and a length of 10, while the other password has a cardinality of 62 and a length of 17.

8.4 PASSWORD CARDINALITY

Entropy will show the passwords variation expressed as bits. It's calculated by a formula provided by Shannon (1948). Where C stands for the password cardinality which is the amount of different elements in a set by using the values in Table 5 for the cardinality and L stand for the length of the password and the formula is: [19]

Table 5. Cardinality

Symbols	Cardinality
a-z	26
A-Z	26
0-9	10
Special Characters e.g. +,V`~!@#$%^&* ()-_=;:'",<.>?	30

UC + LC + D + S = 26+26+10+30 = 92
UC + LC + D = 26+ 26 + 10 = 62
UC + LC + S = 26 + 26 + 30 = 82
UC + D + S = 26 + 26 + 30 = 66
..... *and so on*

Figure 7. Cardinality Values [26]

8.5 Measuring The Strength Of Passwords

Measuring the strength of password is one of the important means to ensure the security of passwords. Current means used to measure the strength of passwords do not provide sufficient accuracy due to applying simple rules. Based on what has been explained earlier, we have proposed a new way to measure the strength of passwords with high accuracy using Marko-models. [18][23]

There are many algorithms and websites that evaluate the strength of passwords but they are weak due to following simple rules such as requiring to use small and capital letters and symbols which lead to the generation of weak passwords. [18][23]

Entropy is one of the important measures used to measure the strength of passwords and evaluate their resistance against brute-force attack. It assesses the strength of password and it is approved by the National Institute of Standards and Technology (NIST). [23]

Strength of passwords can be defined with the extent of necessary power needed to guess and break passwords, thus; it is necessary to increase the strength of passwords in order to increase the time needed to break them and reduce the possibility of being guessed at all. From here we can define the entropy as the average number of possible passwords to guess in order to reach the correct password. [18][23]

A password checker function f(x) can be defined as follows: [18][23]

$$f(x) = -log(P(x)) \qquad (2)$$

Where P(x) is the probabilities.

Through this equation we can classify the password as"optimal" as the password has the same order both when using this equation and an optimal password guessing attack. Consequently we can classify the strength of passwords based on the time necessary to guess the password. [23]

Comparing the results to the extent of password strength through the use of three important sites to guess passwords which are: NIST, Google and Microsoft password checkers. With the knowledge that each of these sites has its specific methodology to assess the strength of passwords as shown in Table 6. [23]
Guessing is one of the standards used to determine the strength of the passwords by setting the time needed to break the passwords and recognize them. They assess the strength of passwords through computing the probabilities necessary to guess the a particular password then comparing the results with the previously mentioned password checker web sites, as shown in Table 6. [23]

Table 6. A Short List of Passwords as Scored by our Markov-Model [18]

Password	Ideal	Markov	NIST	MS	Google
password	9.09	9.25	21	1	1
password1	11.52	11.83	22.5	2	1
Password1	16.15	17.08	28.5	3	1
P4ssw0rd	22.37	21.67	27	3	1
naeemha	21.96	28.42	19.5	1	0
dkriouh	N/A	42.64	19.5	1	0
2GWapWis	N/A	63.67	21	3	4
Wp8E&NCc	N/A	67.15	27	3	4

8.6 ESTIMATING PASSWORD PROBABILITIES WITH MARKOV MODELS

Markov models proved its strength in the field of information security in general and password security in particular. [24]

The power of Markov models based on the extent of the accuracy of calculation of guessing passwords depending on well-known password corpus and generating an n-gram used to calculate the probabilities of new generated passwords. This helps us access to the most accurate results in the evaluation of of these models strength in guessing passwords depending on a large database of frequently used passwords. [24]

For example, suppose that we have D, E, and F. If training data shows that D is the most probably starting character, E is the character most likely to follow D, and F is the character most likely to follow E, then the first guess will be DEF. If the next-most-probable character to follow E is D, the second guess will be DED, and so on. [22]

Goal: Guess passwords and estimate the password probabilities from real password data (e.g. RockYou list). Several very large password datasets have been made publicly available through leaks: the Rockyou dataset, which contains a set of 32 million passwords. [18]

Markov assumption: these subjunctive probabilities can be approximated by a short history, e.g., for 3-grams (history 2):

$$P(password) = P(pa).P(s/pa).P(s/as).P(w/ss).P(d/sw)$$

8.7 Estimating Password Probabilities With Markov Models

By using the n-gram used by Markov-model, the likelihood of the following character in a string based on a prefix of length n. Hence for a given string c1,...,cm we can write: [24]

$$P(c1,\ldots,cm) = \prod_{i=1}^{m} P(ci|ci-n+1,\ldots,ci-1) \quad (3)$$

In n-gram we generate the counts of count(x1,..., xn), and the conditional probabilities can be computed as follows: [18]

$$P(ci|ci-n+1,..,ci-1)=count(ci-n+1,\ldots,ci-1,ci) / \sum_{x \in \Sigma} count(ci-n+1,..,ci-1,ci) \quad (4)$$

n-gram database	
Password	**Count**
aaaaa	17988
aaaab	340
aaaac	303
.....	
passa	1129
passb	225

p(w/pass) = count(passw)/count(pass*) = 97963/114218 0.86.

.....	
passw	97963
.....	
zzzzz	0

Figure 8. Conditional Probability Examples

For instance: To compute the probability of the (password) with n = 5 is calculated as shown below: ***P(password) = P(p)P(a/p)P(s/pa). . . P(d/swor)***
p(o/assw) = count(asswo)/count(assw) = 98450/101485 = 0.97.

And the probability is: P(password) = 0.0016, where the result of (password) from RockYou database is 0.0018. [18]

And familiarized them with Table 6 compared to the results to guess passwords using Markov model compared to the results from other famous password databases and deduce from the table over the accuracy of the results using Markov model. [23][24]

8.8 PROPOSED WORK AND DISCUSSION

Our proposed algorithm checks the strength of generated random password based on these parameters: length, entropy and cardinality and on an additional level of security using Markov model which measures password strength by calculating the "guess probability" (GP) for the password. Through this algorithm, we can put weak passwords in a black list to easily ignore them when created another time without repeating the previous calculations and this will save time.

Using a password of 8 characters long and a composite password which includes symbols, numbers,... etc. Let us suppose that the value of cardinality is 92, and by applying the entropy equation, we got a final result 52.4, and according to the password standards we note that short passwords can be guessed easily and thus it is necessary to increase the length of the password to be more than 32 characters according to RFID tag features and the password should be as complex as possible.

In our algorithm, we proposed a password with a length of 32 characters and an optimal password cardinality with a score of 92 and by applying the entropy equation we got a result 208.7, so that our conditions will be password length >=32, cardinality>=92, entropy >=208.7 and the "guess probability" (GP) should be over a specific threshold. The resulting score indicates the strength of generated passwords, if it is classified as weak passwords we put it in a black list to be ignored next time before check and we regenerate another password, while if it is classified as strong password we accept it.

Our proposed algorithm checks the strength of the generated password as shown in Figure 9. In our system we chooses an active tag and the reader generates a password based on our proposed algorithm which generates a password bigger than or equal to 32 bit and check if the password doesn't exist in the black list, it computes the cardinality depending on the structure of generated password. After that, the algorithm computes the entropy which is a measurement to estimate the strength of password to resist brute-force attack based on the value of cardinality and the length of generated password, then the algorithm computes the probability based on Markova model which indicates the strength of generated passwords, the proposed algorithm should be greater than a specific threshold. Finally, by aggregating the two results (entropy and guess probability) the final result shows the strength of password: If it is strong it will be used in the authentication process. If not, the algorithm will add it to a blacklist and regenerates a new password.

In order to aggregating the two results (entropy and guess probability), we suggest to use Mamdani Fuzzy logic model. This can be done by applying two-input one-output model which takes (entropy and guess probability) as input while the output will be the prediction result for their composition. In fuzzy logic, we classify the inputs into membership functions classified to (LOW, MEDIUM, HIGH) and the outputs can be

classified as (LOW, MEDIUM, HIGH) and through defuzzification, this model gives us a crisp values which determines the strength of password as shown in Figure 13. [25][26]
We also proposed a mechanism to fill the matrix based on Electronic Product Code (EPC), which contains a decimal and a hexadecimal code. In order to fill the matrix, we used the object class part and the serial number part and excluded zeros from both of them. Then converted the hexadecimal numbers to decimal format and filled all the decimal numbers to the matrix as shown in Figure 10.

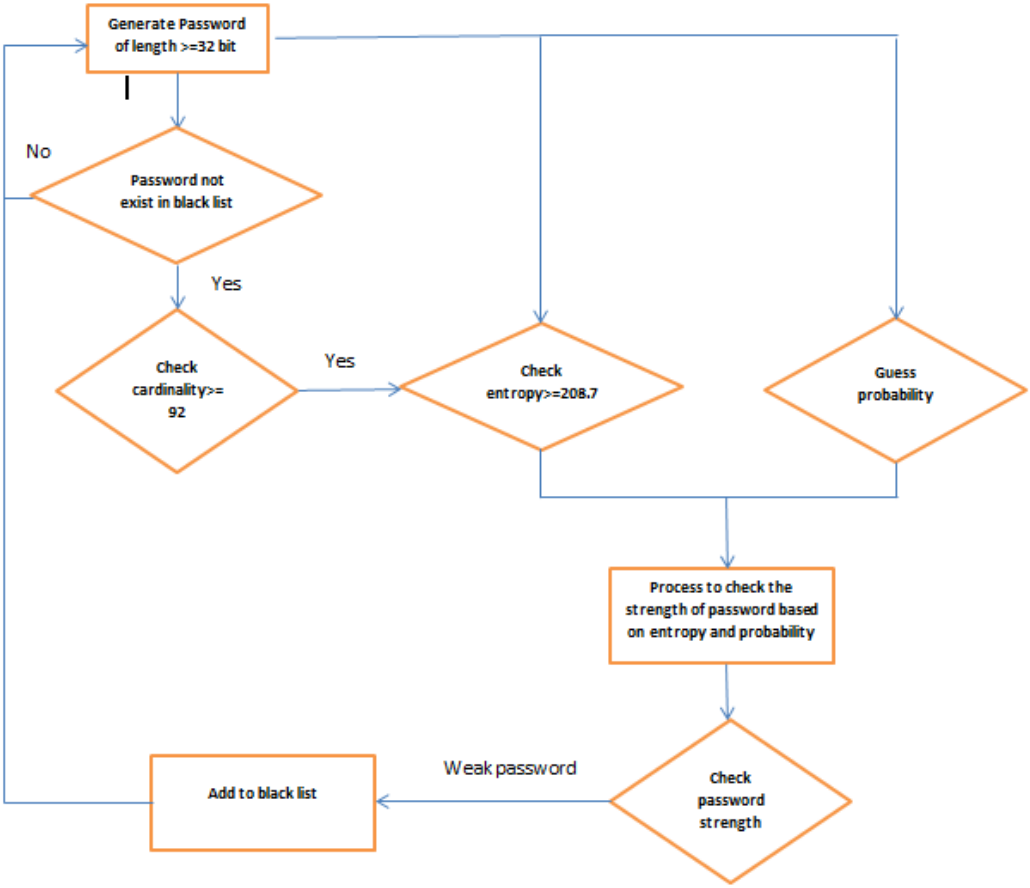

Figure 9. Proposed Algorithm

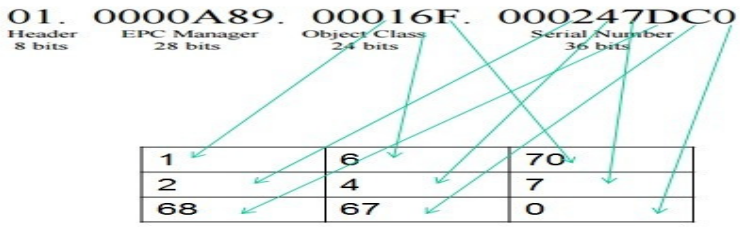

Figure 10. Proposed Mechanism to Fill the Matrix

8.9 ALGORITHM DESCRIPTION

Our enhanced algorithm enhances the security issues of the authentication process between RFID tag and reader. This algorithm requires that both tag and reader store two square matrices and share a key K as follows:

The tag: stores M1 and $M\,2^{-1}$, where $M\,2^{-1}$ is the inverses of M2.

The reader: stores M2 and $M1^{-1}$, where $M1^{-1}$ is the inverses of M1.

Database: stores information about the tags and each of the tags' information was indexed with a unique number X which equals to the multiplication of K and M1 (KM1).

The key K: is generated using our algorithm and a new key is generated for every identification session.

The matrices: M1, M2, M 1^{-1}, $M\,2^{-1}$, and the matrices are filled from EPC code as discussed before.

In the previous algorithm, both of the matrices and the key are generated randomly which still can be guessed through a brute-force attack. In our algorithm, on the other hand, we used a new algorithm to generate a strong key and a mechanism that fills the matrix making it more complicated and difficult for a hacker to guess.

The identification composes of two phases:

First phase: Tag identification. It happens when the reader send a SYNC message to the tag and starts a session, the tag then replies with ACKN message which is X=KM1, note that each tag is indexed in the database with a unique number X. While the tag replies, a timer starts, and when the reader extradites a unique number X, it communicates with the database through middleware to get the tag's information identified by X.

Second phase: the reader authentication. In this phase the reader tries to authenticate itself to the tag and through this process it generates a new key using our algorithm and sends it back to the tag. While the reader makes new authentication, it generates a new key and instead to send the whole keys back to the tag and to save the tag resources, the reader uses XOR to get a small one key instead of all generated keys and multiply the Knew with the matrix M2. And to obtain a new key the reader uses X*new* such that K*new* $\rightarrow$ *XnewM* 1^{-1}. Then the reader sends both Y and Z resulting from the multiplication of the new key with M2 to the tag which accepts and stores the new key in their memory to be used in a new identification and authentication phases, and at this time the tag stops the timer and the tag and so on.

In case of failure of the reader to be authenticated to the tag, the tag will stop the connection until reset and it can create one authentication at a session time.
The proposed algorithm is shown in Figure 11 and Figure 12.

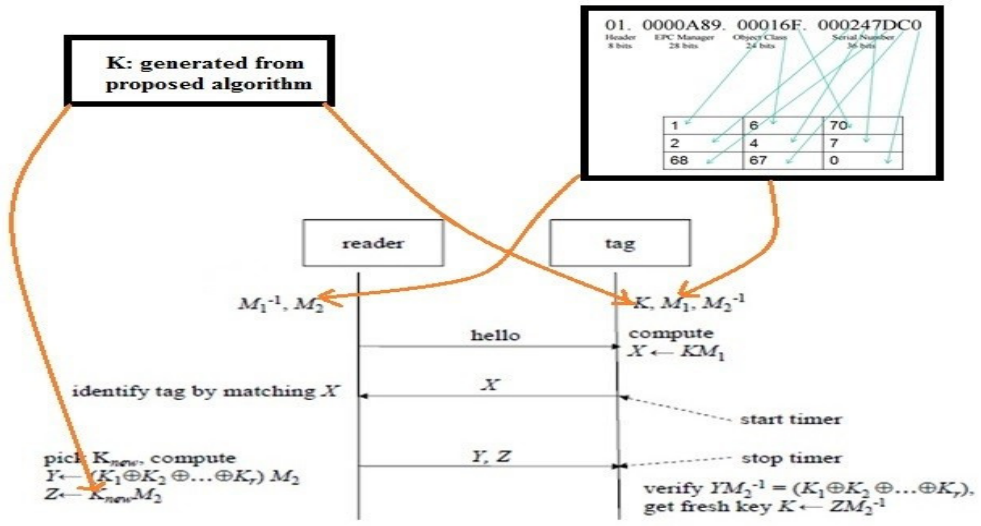

Figure 11. Propose Model for Secure Tag Identification Algorithm

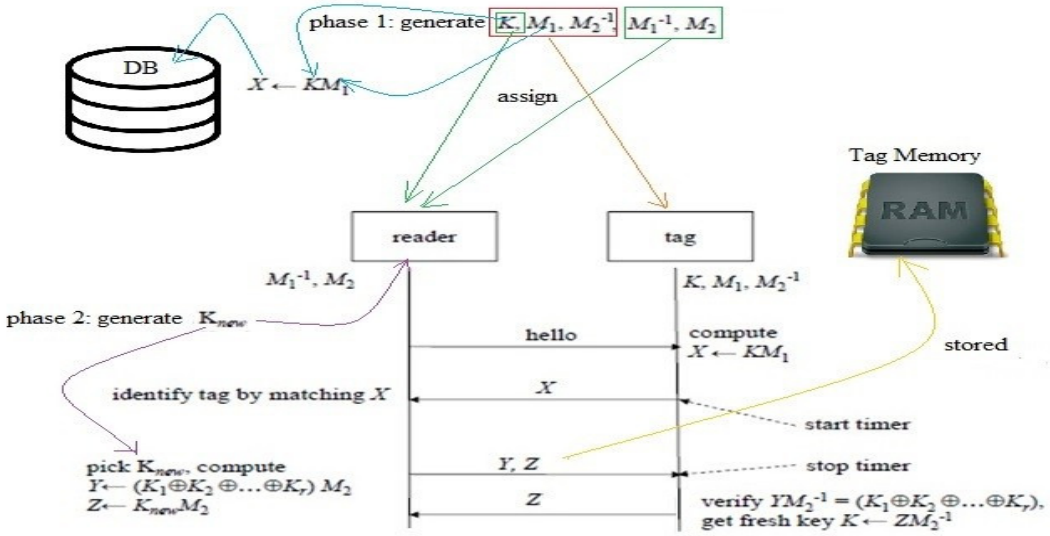

Figure 12. Proposed Secure Tag Identification Algorithm

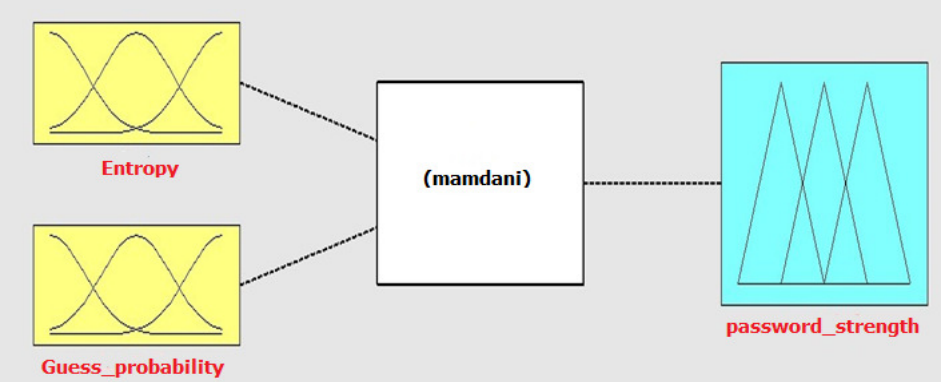

Figure 13. Mamdani Fuzzy Logic Model to Measure Password Strength

9. CONCLUSION AND FUTURE WORK

It can be positively concluded that our algorithm ensures RFID authentication security against brute-force attack. The algorithm prevents generated passwords that fail even a single condition. In order to be accepted, a password should pass all conditions of length, cardinality, and entropy and Markov probability. Our future work will be on the implementation part of the proposed algorithm by analyzing the results and comparing them with such algorithms.

Table 3. RFID Threats and Countermeasures for all Layers

	Countermeasures																		
	Cryptographic							Non-cryptographic					Others						
RFID Threats	Anonymous-ID Scheme	Public Key (Re-) Encryption	Hash Lock	Randomized Hash Lock	Hash-Chain Scheme	Pseudonym Throttling	Delegation Tree Authentication	Tag Killing	Tag Locking	Faraday Cage	Blocker Tag	RFID Guardian	Password Protection	Authentication	Active Jamming	secure channel	Time-based and counter-based	challenge and response mechanism	read protected memories	TOTAL
Eavesdropping Data	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X				15
Eavesdropping Transmission								X		X	X	X								4
Spoofing	X	X	X	X	X	X	X	X	X	X	X	X	X	X						14
Cloning	X	X			X	X	X	X	X	X	X	X	X	X					X	13
Denial of Service								X		X	X	X	X	X						6
Tracking	X	X		X	X	X		X		X	X	X								9
Impersonation													X	X						2
Traffic Analysis													X	X						2
Tag modification		X			X									X						3
Man-in-the-middle		X												X		X				3
Replay													X	X		X		X		4
Cryptography - Brute force attack		X												X		X				3
PRIVACY		X		X				X	X	X					X					6

REFERENCES

[1] Mandeep Kaur, Manjeet Sandhu, Neeraj Mohan and Parvinder S. Sandhu, (2011) " RFID Technology Principles, Advantages, Limitations & Its Applications", International Journal of Computer and Electrical Engineering, Vol.3, No.1, 151-157 ISSN: 1793-8163, February, 2011. DOI: 10.7763/IJCEE.2011.V3.306

[2] Nidhi Chauhan, (2014) " Vulnerability and Countermeasures of RFID System", International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Volume-2, Issue-9, September 2014

[3] Dong-Her Shih, Chin-Yi Lin and Binshan Lin, (2005) "RFID tags: Privacy and security aspects", International Journal of Mobile Communications Vol.3 (3):214-230. January 2005. DOI: 10.1504/ ijmc.2005.006581

[4] Gursewak Singh, Rajveer Kaur, Himanshu Sharma, (2013) "Various Attacks and their Countermeasure on all Layers of RFID System", International Journal of Emerging Science and Engineering (IJESE) ISSN: 23196378, Volume-1, Issue-5, March 2013

[5] Sindhu Karthikeyan and Mikhail Nesterenko , (2005) "RFID Security without Extensive Cryptography" , Conference Paper published 2005 in Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks - SASN '05. DOI: 10.1145/ 1102219.1102229

[6] RFID USER MANUAL. URL: https://datalocker.com/wp-content/uploads/2014/03/DL3-User-Manual.pdf (visited on 17/4/2016)

[7] Dushyant Kumar Sahu, Asit Xaxa,Atul Sahu , (2014) "CLASSIFICATION OF RFID THREATS BASED ON SECURITY PRINCIPLES", GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES [Sahu, 1(10): December, 2014] ISSN 2348 – 8034

[8] Ari Juels, (2006) " RFID Security and Privacy: A Research Survey", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 2, FEBRUARY 2006. DOI: 10.5121/ csit.2013.3526

[9] Vaibhaw Dixit , Harsh K. Verma , Akhil K. Singh, (2011) "Comparison of various Security Protocols in RFID",International Journal of Computer Applications (0975 8887) Volume 24 No.7, June 2011. DOI: 10.5120/2951-3965

[10] J. A. Cazier and D. B. Medlin, (2006) "Password security: An empirical investigation into e-commerce passwords and their crack times" Information Security Journal: A Global Perspective, vol. 15, no. 6, pp. 45–55, 2006. DOI: 10.1201/1079.07366981/46352.34.5.20061101/95107.article.1

[11] Melissa Walters, Erika Matulich," Assessing password threats: Implications for formulating university password policies", Journal of Technology Research

[12] Hsien-Cheng Chou1,Hung-Chang Lee..et al, (2013) " PASSWORD CRACKING BASED ON LEARNED PATTERNS FROM DISCLOSED PASSWORDS", International Journal of Innovative Computing, Information and Control Volume 9, Number 2, February 2013 ICIC International ©2013 ISSN 1349-4198 pp. 821–839

[13] Ms. Vidya Vijayan, Ms. Josna P Joy, Mrs. Suchithra M S," A Review on Password Cracking Strategies", International Journal of Research in Computer and Communication Technology ISSN(O) 2278-5841 ISSN(P) 2320-5156

[14] How does a weakness in a random number generator lead to a compromise of the entire cryptographic process. URL: http://security.stackexchange.com/questions/42327/how-does-a-weakness-in-a-random-number-generator-lead-to-a-compromise-of-the-ent (visited on 17/4/2016)

[15] Manber U, (1996) "A simple scheme to make passwords based on one-way functions much harder to crack", Computers & Security Journal, Volume 15, Issue 2, Pages 171-176. Elsevier. 1996. DOI: 10.1016/0167-4048(96)00003-x

[16] Bryan Sullivan, SPI Dynamics,"Preventing a Brute Force or Dictionary Attack: How to Keep the Brutes Away from Your Loot"

[17] Niklas Ekstrom,"PASSWORD PRACTICE The effect of training on password practice", Bachelor Degree Project in Computer Science

[18] Claude Castelluccia,..et al,"Adaptive Password-Strength Meters from Markov Models"

[19] Fatin G. Sabbagha and Robert H. Clarke, (2013) "MATCHING WEBSITE PASSWORD REQUIREMENTS STRENGTH WITH ROCKYOU PASSWORD SET", Electrical and Electronics Engineering: An International Journal (ELELIJ) Vol 2, No 4, November 2013

[20] Sarita Pais, Judith Symonds, (2011) "DATA STORAGE ON A RFID TAG FOR A DISTRIBUTED SYSTEM", International Journal of UbiComp (IJU), Vol.2, No.2, April 2011. DOI: 10.5121/ iju.2011.2203

[21] Matt Ward, Rob van Kranenburg, (2006) "RFID: Frequency, standards, adoption and innovation", JISC Technology and Standards Watch, May 2006

[22] Patrick Gage Kelley,..et al, (2012) "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms", San Francisco,Conference in 2012 IEEE Symposium on Security and Privacy (SP), pp. 523-537. IEEE, 2012

[23] Mudassar Raza, Muhammad Iqbal, Muhammad Sharif and Waqas Haider, (2012) " A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication", World Applied Sciences Journal 19 (4): 439-444, 2012 ISSN 1818-4952; © IDOSI Publications, 2012. DOI: 10.5829/idosi.wasj.2012.19.04.1837

[24] S. Vaithyasubramanian, A. Christy, D. Saravanan, (2014) "An Analysis of Markov Password Against Brute Force Attack for Effective Web Applications", Applied Mathematical Sciences,Vol. 8, 2014, no. 117, 5823 - 5830 HIKARI Ltd, www.m-hikari.com. DOI: 10.12988/ams.2014.47579

[25] Arshdeep Kaur, Amrit Kaur, (2012) " Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-2, May 2012. DOI: 10.1109/ icacea.2015.7164799

[26] Poonam , Surya Prakash Tripathi and Praveen Kumar Shukla, (2012) " Uncertainty Handling using Fuzzy Logic in Rule Based Systems", International Journal of Advanced Science and Technology Vol. 45, August, 2012

AUTHORS' PROFILES

Dr. Adwan Yasin PH.D Computer System Engineering /Computer security /Kiev National Technical University of Ukraine, 1996. Master degree in computer System Engineering- Donetsk Polytechnic institute, Ukraine, 1992. An associate professor former Dean of the Engineering and Information Technology Faculty of the Arab American University of Jenin, Palestine. Previously he worked as Chair Person of Computer Science Department- AAUJ, Assistant professor of the computer Science Department -The Arab American University- Palestine. Assistant professor of the computer & Information System Department, Philadelphia University- Jordan. Assistant professor of the Computer science department, Zarka Private University- Jordan. He has many publications in the fields of computer networking and security in different international journals. His research interests Computer Security, Computer Architecture and Computer Networks.

Fadi K M AbuAlrub received his BS in telecommunication technology from Arab American University in Palestine, Ramallah, in 2006. He is currently working with State Audit and Administrative Control Bureau as a software engineer and IT auditor since 2008, and currently pursuing his Master of Computer Science from Arab American University, Jenin, Palestine. His researches interest includes computer networking, information security, artificial intelligence, RFID technology and data mining.