Rogue AP Detection: The Tale of a Datagram and its Algorithms

Piyush Choudhary
Rajiv Gandhi Institute of Technology,
Mumbai University
Mumbai

Vineet Trivedi
Rajiv Gandhi Institute of Technology,
Mumbai University
Mumbai

ABSTRACT
Wireless LAN, in the current state of the world, has become ubiquitous. Therefore it is imperative to safeguard this technology, which otherwise could prove disastrous. Compromised WLANs, have the potential to leave the user susceptible to a plethora of unfavorable situations. In the following paper it is attempted to make wireless networks more easily secure by addressing one of the more commonly exploited technique of Rogue Access Points. This problem is tackled by articulating a method by which clients can recognize Access points to which they have previously connected. After a standard authentication procedure a packet exchange mechanism is used buttressed by a host of algorithms, selected randomly from an algorithm pool, which are run on selected packages on the client as well as the Access Point in order to completely obviate the possibility of a client connecting to a Rogue Access Point.

Keywords
Access Points, Rogue Access Points, Wireless sniffing, MITM, Traffic Monitoring.

1. INTRODUCTION
The foundation of the Rogue AP problem lies with the fundamental problem of inability of the client to authenticate the AP. This fact is so severely exploited that it enables majority of the rogue AP attempts to success.

The authentication phase specified in the Wired Equivalent Privacy (WEP) [1] and Wi-Fi Protected Access (WPA2) [2] protocol of the IEEE 802.11 standard, provides a robust method in order to make sure that only the legitimate clients are able to connect to the network. Though this being of paramount importance, is not complete. The IEEE 802.11 standard for any of the existing protocols, except the 802.1x RADIUS based authentication [2] [3], does not inherently provide any means, directly or indirectly, for the client to check if it is connecting to the correct Access Point.

There are several methodologies that been proposed to work around this shortcoming. These include,

- Involving the signal strength of the connected Access Points.
- Scanning the entire network, NMAP [4], to detect any rogue AP.
- Comparing MAC addresses.
- Pin-pointing the location of the Access Point.
- Using Temporal Traffic Characteristics.[5]

And many others.

However, to best of our knowledge, majority of these strategies either possess an easy work around or involve the use of an external component or intricate setup, which would prove as a deterrent to the not acquainted in the matter majority.

In this paper a method is suggested, which would not require any user participation. The client and the device would figure out each other's legitimacy by a packet exchange protocol. This method does not require any external components and would be barely noticeable to the user.

This paper follows up with the description of the problem, by explaining a common scenario and then moves on to explain the proposed solution. In the latter part before concluding, some numbers are crunched up, in order to display the complexity and the strength required to break the protocol.

1.1 Explaining a General Scenario of the existing problem
To describe the general common scenario, consider a rogue AP (RAP), a genuine AP (GAP) and a victim client. Initially the victim has successfully established a connection to the GAP in order to access the internet.

The attacker handling the RAP, has the final goal to get the client's data. So, first in order to create a Rogue AP, the details are copied off by monitoring the beacon frames of the GAP. This gives of the SSID and the BSSID of the GAP. This information is used by the attacker to create an RAP with the same details in order to deceive the victim.

Now, the RAP plans to intercept the communication originating from the victim, in a Man-In-the-Middle (MITM) [6] [7] fashion. The first step would be to disconnect the already established connection between the victim and the GAP. This is easily accomplished by either flooding the entire network of the GAP with deauthentication packets or place a targeted disconnection towards the client, which, either ways, severs the connection. At this state the client is looking to reconnect, which is when the RAP come into play.

The reconnection strategy uses the election algorithm among others to connect. This algorithm says better the signal strength higher the priority to connect to. This fact along with the inability of the client to connect to the GAP (which is still being bombarded with deauths), forces the victim to connect to the RAP. That's the end.

Since the client cannot identify the RAP as a fake, it readily connects to it considering it as a genuine AP. From this point on, the RAP can forward the data to the internet and monitor the data sent by the client indefinitely.

2. PROPOSAL

Following is the structure of the databases on the client and access point side respectively. These databases store information related to the clients and access points which include MAC addresses and packets. These databases are critical to the validation process.

Table 1. 'Validate' – Client Side

Sr. No.	AP MAC	Packet
1	AA:AA:AA:AA:AA:AA	11001001
2	BB:BB:BB:BB:BB:BB	10010111

Table 2. 'Validate' - Access Point Side

Sr. No.	AP MAC	Packet
1	AA:AA:AA:AA:AA:AA	11001001
2	BB:BB:BB:BB:BB:BB	10010111

Table 3. 'Operations'

Sr. No.	Algorithm
1	Reverse
2	Shift right by 2
3	2's Complement
4	Half XOR
5	Half Add
6	3/4th Subtract
7	Half Multiply
8	XOR 'PiyushVineet'
9	Multiply by no. of ones in data
10	Multiply by sum of algorithm numbers

There are 10 algorithms in table 'Operations'. The table will remain as specified above for all Access Points and devices, irrespective of manufacturers.

2.1 Client connects to AP

- Standard authentication phase.
- After the standard authentication process is over the client checks for the mac address of the AP in its table 'validate'.
    - If MAC address of AP is not found, it displays a message "New AP, Proceed?" Connection successful on selecting 'Yes' by User, and the client adds the mac address of the AP in its table 'validate'. Otherwise the connection is aborted and nothing is added to the table.
    - If MAC address is found, validation process follows.

Each packet received by the AP is stored in the database. When a new packet arrives with a greater Timestamp, the previous packet in the database against the same client MAC address is overwritten by this new packet. Similarly each packet sent by the client is stored in the database. When a new packet is sent with a greater Timestamp, the previous packet in the database against the same AP MAC address is overwritten by this new packet. At the time of disconnection the client and the AP store the last packet successfully sent by the client to the AP. The client stores this packet against the mac address of the AP in its table 'validate'. The AP stores the packet against the mac address of the client in its table 'validate'. As this packet will be encrypted any attacker sniffing the packets will not know its contents.

2.2 Validation Process

- Client generates a random number (a good example could be the Fortuna Pseudo Random Number Generator (PRNG) [8]) say 'x' ($4 \leq x \leq 12$). The client then further generates 'x' random numbers. The value of each of these random numbers is between 1 and 10 inclusive.
- The client then sends a packet containing these 'x' random numbers to the AP. The order of the random numbers should not change i.e. if the 'x' random numbers generated are 1, 6, 5, 9 then the client sends them in same order.
- These random numbers are to be matched, by the AP, with the serial numbers in the table 'Operations'. The corresponding algorithms are to be executed in succession on the data packet stored against the mac address of the client.
- If the numbers obtained are 1, 6, 5, 9 the AP checks the table 'Operations'. As we can see from table 'Operations' the algorithm 'Reverse', '$3/4^{th}$ Subtract' 'Half add' and 'multiply by no. of 1's' are against the serial numbers 1, 6, 5 and 9 respectively. These algorithms are then executed on the data in the packet stored against the mac address of the client, in sequence so as to maintain the order i.e. the algorithm against the serial number 1 is executed first followed by 6, 5 and 9.
- The client performs the same operations as the AP on the data in the packet stored against the mac address of the AP.
- The AP sends a packet containing the resultant data, to the client.
- The client then verifies this data with its own computed data. If the verification is not successful then an alert message, notifying the user of a potential rogue AP, is displayed.

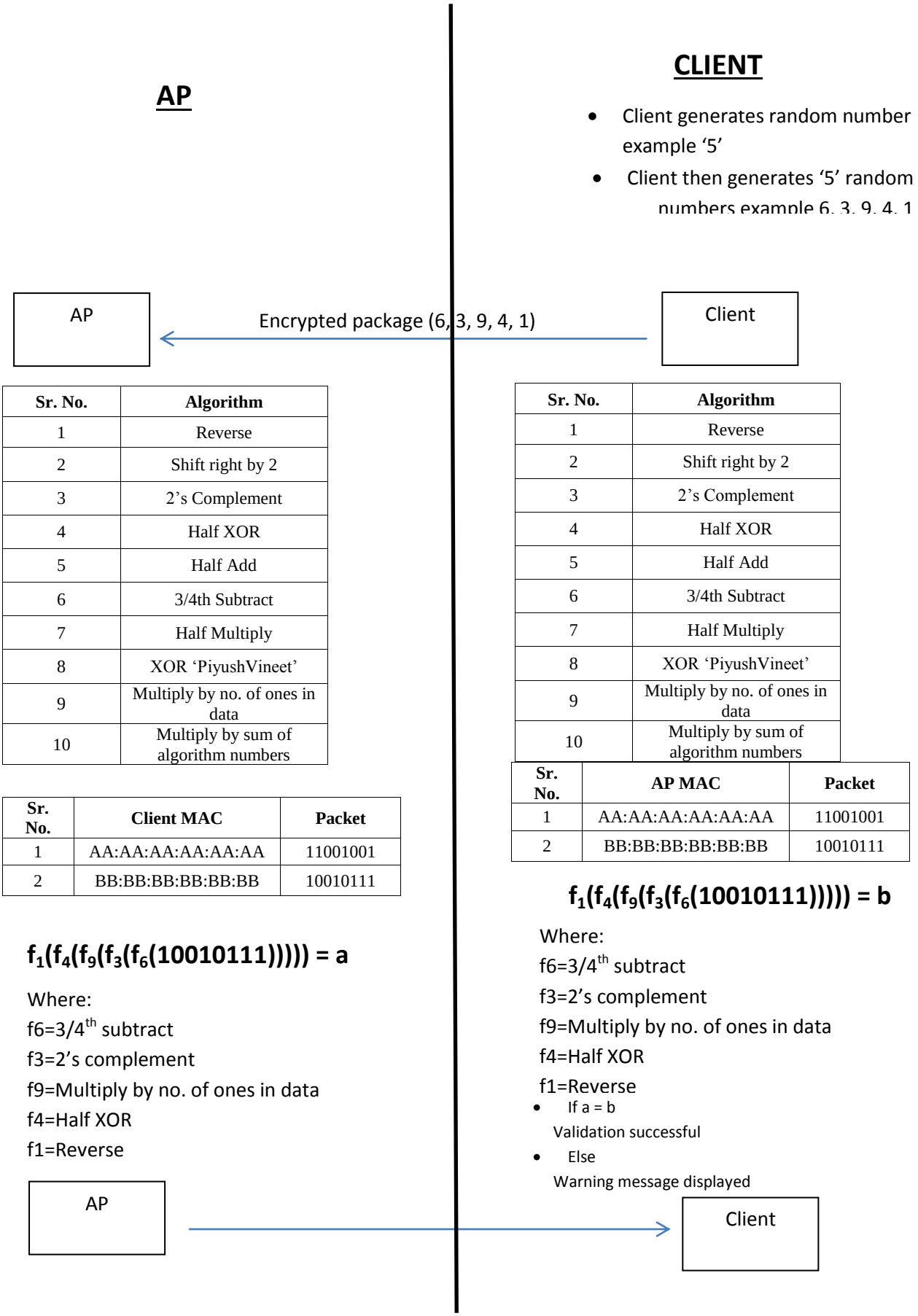

Fig 1: Flow chart for the packet exchange between the client and Access Point

3. RESULTS AND ANALYSIS

One of the methods to increase the security of any given system is to increase its randomness and hence increasing its complexity. By including the table 'Operations' we accomplish this objective.

As we can see there are 10 algorithms in table 'Operations' and the number of algorithms(x) to be executed on the Packet is: $4 \leq x \leq 12$

Total number of possibilities: $(10^4) + (10^5) + (10^6) + (10^7) + (10^8) + (10^9) + (10^{10}) + (10^{11}) + (10^{12}) = Y$

Where Y > 1 Trillion possibilities.

4. CONCLUSION

A new technique is presented to handle the rouge AP problem. This technique is not only more robust, but also doesn't hassle the end user much. By making use of the last packet successfully sent by the client to the AP, we simplify many of the problems regarding complexity in implementation put forth by many other methods. The fact that breaking the algorithm requires more than 1 trillion permutations, intimidates any attempts for trying the same against the algorithm. With the fact that it needs no special equipment, this method could be easily deployed for household as well as office purposes. Though we acknowledge that it is not impossible to counter it, which we don't think can ever be said, this method still presents a massive potential to be proved as one of the more secure techniques.

5. REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[2] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[8] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[9] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender.