

Received August 27, 2020, accepted September 9, 2020, date of publication September 28, 2020, date of current version October 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027325

Effective Wireless Communication Architecture for Resisting Jamming Attacks

AMANY ALSHAWI¹, PRATIK SATAM², FIRAS ALMOUALEM³,
AND SALIM HARIRI², (Senior Member, IEEE)

¹Communication and Information Technology Research Institute, King Abdulaziz City for Science and Technology, Riyadh 11442, Saudi Arabia

²Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA

³The Aerospace Corporation, El Segundo, CA 90245, USA

Corresponding author: Amany Alshawi (aalshawi@kacst.edu.sa)

This work was supported in part by the Air Force Office of Scientific Research (AFOSR) Dynamic Data-Driven Application Systems (DDDAS) under Award FA9550-18-1-0427, in part by the National Science Foundation (NSF) Research Projects under Grant NSF-1624668 and Grant NSF-1849113, in part by the National Institute of Standards and Technology (NIST) under Grant 70NANB18H263, and in part by the King Abdulaziz City for Science and Technology.

ABSTRACT Over time, the use of wireless technologies has significantly increased due to bandwidth improvements, cost-effectiveness, and ease of deployment. Owing to the ease of access to the communication medium, wireless communications and technologies are inherently vulnerable to attacks. These attacks include brute force attacks such as jamming attacks and those that target the communication protocol (Wi-Fi and Bluetooth protocols). Thus, there is a need to make wireless communication resilient and secure against attacks. Existing wireless protocols and applications have attempted to address the need to improve systems security as well as privacy. They have been highly effective in addressing privacy issues, but ineffective in addressing security threats like jamming and session hijacking attacks and other types of Denial of Service Attacks. In this article, we present an “architecture for resilient wireless communications” based on the concept of Moving Target Defense. To increase the difficulty of launching successful attacks and achieve resilient operation, we changed the runtime characteristics of wireless links, such as the modulation type, network address, packet size, and channel operating frequency. The architecture reduces the overhead resulting from changing channel configurations using two communication channels, in which one is used for communication, while the other acts as a standby channel. A prototype was built using Software Defined Radio to test the performance of the architecture. Experimental evaluations showed that the approach was resilient against jamming attacks. We also present a mathematical analysis to demonstrate the difficulty of performing a successful attack against our proposed architecture.

INDEX TERMS Denial of service (DoS), jamming attack, resilient communication system, software defined radio.

I. INTRODUCTION

Wireless communication has been used extensively owing to its simple deployment, low cost, and improved bandwidth capabilities. However, increased utilization of wireless communication has increased the frequency of cyberattacks. There are wireless conventions that emphasize issues of privacy, for example, the contribution by Keke *et al.* [1]. However, there is an essential need for solutions that offer feasible resilience against cyberattacks, such as jamming attacks, denial of service attacks (DoS), and session hijacking.

The associate editor coordinating the review of this manuscript and approving it for publication was Wenchi Cheng¹.

A preliminary version of this work was presented at the IEEE International Conference on Cloud and Autonomous Computing 2017 (ICCAC 2017) [2]. This study is an extension of the previous work to which the authors now provide additional insights through deeper and more detailed experimentation and analysis. Newer algorithms have been implemented to ensure that the proposed system could be utilized for a broader class of applications. Moreover, a dedicated section has been incorporated to describe the architecture's tolerance to attacks and analyze the performance and overhead. Finally, an entire section has been added to discuss certain complexity aspects of the architecture and mathematically prove that the proposed approach shows sub-quadratic

time behavior for some attack scenarios. Moreover, we have confirmed that the configuration time is linear with respect to the available redundant links and the attack probability. Therefore, our resilient wireless communications architecture has demonstrated evidence of efficient performance.

Our robust methodology is based upon the paradigm of a Moving Target Defense (MTD). MTD uses a diverse positioning mechanism. It randomly alters the position to constrain the exposure of existing vulnerabilities and present significantly fewer opportunities for successfully launched attacks to prevail [3], [4]. For example, we can variably readjust radio frequency communications parameters such as packet size, operating frequency, modulation scheme, and network address, to decrease the vulnerability of wireless communications. The selected methodology uses a Software Defined Radio (SDR) program for executing MTD algorithms [5]. We use two radio channels: an active channel, and the other a standby channel. The standby channel is used when an attack succeeds in damaging the active radio channel. A radio jamming attack is a DoS attack in which the attacker continuously transmits a signal that prevents communicating entities from using the services they need [6].

The remainder of this article is structured as follows. Section II discusses related works in addition to defining the technologies used to implement the solution, namely MTD and SDR. Section III provides a short background of the method. Section IV elaborates further upon the details of the execution approach. Section V exhibits our assessment of the robust methodology and explains how it can be used to tolerate large numbers of attacks. Section VI shows execution time measures and proves the sub-quadratic complexity of some elements in our approach that contribute to sustaining its temporal efficiency. Finally, Section VII concludes the paper and proposes the future direction for this research.

II. RELATED WORK

A. JAMMING ATTACK MECHANISMS

Jamming attacks are the most significant type of attack against wireless communication, and can be launched through different approaches [7], [8]. In the first approach, the attacker scans the traffic to detect the Start of Frame Delimiter (SFD) to prevent the receiver from getting the transmitted packets by jamming the channel. A pseudo number agreement between the sender and the receiver for the SFD generation can prevent this type of jamming attack. In the second approach, the radio channels are periodically scanned by the jammer. If the signal power is known, the jammer will then attack the channel. The sender should use frequency hopping to defend against such an attack further altering the functioning frequency linked with every phase. In the third approach, the attacker scans all communication channels for short intervals of time in a cyclic fashion aiming to determine the active communication channel. On detecting the active communication channel, the attacker jams the channel. Communication packets are split into smaller sizes to prevent

the attacker from having enough time to detect the message. In the fourth approach, a jammer sends short pulses via the channel, thus affecting all messages transmitted. Packet encoding prevents the attacker from executing this attack successfully.

Recent literature provides examples of several approaches for jamming attack detection and countermeasures. For example, a fault management scenario, in which, a fault occurs in a power network and alarm messages propagate along its nodes is analyzed in [9]. There are situations where some nodes cannot be reached because, among other possible reasons, malicious jamming attacks also occur. This leads to a situation in which other nodes do not receive the alarm message and possibly damage the nearby devices. For that reason, an approach is proposed for a Fault Detection, Isolation, and Service Restoration (FDIR) system whose main aim is to find the exact location of the fault, analyze it, and verify it. A related discussion, in the context of the power grid has been developed and a complete taxonomy of attacks including jamming has been presented in [10]. The authors discuss the obstruction of wireless sensor networks caused by jamming attacks in [11]. Using simulated scenarios and results, the significance of such obstructions could be determined.

The Maximum Attacking Strategy using Spoofing and Jamming (MASS-SJ) is presented in [12]. This attack strategy applies optimal power distribution to maximize the adversarial effects in order to interfere with the maximum number of signal channels. In the Request to Send (RTS) fake jamming attack, the attacker scouts the network; if the network is weak, the attacker will convey a false RTS to the access point, which responds with a Clear to Send (CTS) and reserves the bandwidth. In this process, the attacker stops receivers from transferring for the duration set by CTS. If the access point observes the grid, this type of attack might be averted [6].

B. MOVING TARGET DEFENSE

MTD is used to “make, assess, and set out strategies and mechanisms that are repeatedly changed, varied, and altered over a period of time. It enhances costs and difficulty for attackers and constrains the exposure to opportunities and risk of assaults, enhancing the resiliency of system” [13]. The following points describe the stages of an attack: 1) Reconnaissance: In this stage, the attacker gathers all the information from the target system environment. For wireless communication, this information includes the operating channel, modulation type, and bit rate; 2) Planning: During this stage, the attacker plans the attack and decides on a strategy based on the information obtained during the reconnaissance stage; 3) Execution: In the execution stage, the attacker carries out the attack on the target. MTD provides a resilient solution to network attacks by changing the configuration of the communications parameters at random time intervals. The application of MTD to wireless communications reduces the level of existing communication vulnerabilities that can be exploited by the attackers, in the event of a successful attack; however, this will last for a short period only. The attack

will fail after the communication parameters are changed, consequently ensuring communication resilience.

The notion of MTD is applicable at different stages of the hierarchy of structural design from the network level to the application level. The IP address can be randomly changed at the network level [14]. In this situation, both real and virtual IP addresses are used. The attacker cannot determine the origin of the packet as it utilizes the Virtual IP address (VIP). Before the packet arrives at the desired network, VIP is used throughout the system, and the receiver then converts the VIP to the Real IP address (RIP). To run each application, the MTD alters the implementation environment [15], [16].

In MTD, replication, diversity and random shuffling techniques are implemented to change the execution environment. The MTD approach can be generalized in order to create robust smart city services [17]. The robust practice uses significant elements at the level of resilient command and control applications as well as resilient communication services. All the communication and computation resources involved in this approach will be altered for conveying city services, making it highly challenging to penetrate; it would also be difficult to determine the properties of the communication links used. This would make it difficult to execute cyberattacks and would guarantee that all smart city facilities are running properly.

C. NETWORK CYBERSECURITY AND RESILIENCE

Existing cybersecurity precautions have failed to protect and secure network operations and services. The resilient network approach is one of the most promising approaches to mitigate any type of attack. In addition, network programmable systems can be applied to improve the system's robustness [18]. The "network programmability" property refers to a network's ability to alter its behavior according to existing system conditions. This results in the ability to eliminate attacks and ensure normal operations. Using programmable networks, the researchers in [18] achieved the resiliency to detect and protect against flash crowd attacks using programmable networks to detect and mitigate malicious attack. The detection mechanism is based on comparing the response traffic volume with the expected value. If the values are beyond the normal threshold, then an attack occurrence is declared. After confirming the attack, one of the following two solutions is used: The first is to change the direction of packets and to distribute them among other routers. The second solution is to release harmful packets along the routes of the attacker.

The development of highly resilient and secure network services became possible after the introduction of Software Defined Networking (SDN). Some researchers have mitigated Distributed DoS Attacks (DDoS) using the SDN strategy [19]. Using a pairwise key, other scholars have emphasized resiliency in wireless systems, which could be executed in three steps: initialization, direct key setup, and path key setup. According to the process, the essential properties of wireless systems security are integrity, authentication,

and confidentiality [20]. By contrast, the survivability requirements for wireless systems are reliability, availability, and energy efficiency.

Researchers are still interested in detecting new types of DDoS attacks. A model for a structural health monitoring network system has been proposed in [21]. The model is designed to protect against a flooding attack, a type of DDoS attack. The authors examined several network configurations, parameters, attack options, and scenarios. Based on their analysis, a new type of DDoS attack has been reported: Delayed Distributed Denial of Service attack (DDDoS). Jamal *et al.* [22] discuss the RTS attack, a form of DoS attack where malicious nodes reserve the medium unnecessarily for a prolonged period of time. Additionally, the authors provided a mitigation technique to restore network performance. Several modeling approaches for capturing the uncertainty in DoS attack strategies have been discussed in [23]. Special attention is paid to the Tail-Probability Based Failure Models (TPBFM) for describing the jamming attacks affecting wireless channels.

A resilient wireless network can also be developed through a channel hopping technique [20]. The resilience of this method relies on the channel frequency, randomness, and hopping rate. The resilience and performance of a wireless network are affected by the hopping time. Considering configuration time, the hopping time should not be set to a very small value. However, the duration should not be long, as that would provide the attacker with adequate time to determine the current vulnerabilities and launch a successful attack. Another approach to achieve resilience has been presented in [20], [24]. This is based on using multiple redundant and diversified routes to tolerate attacks when one of the routes is being compromised.

D. SOFTWARE DEFINED RADIO

The SDR approach is a reconfigurable radio built using Field Programmable Gate Arrays (FPGA). This implements different communication protocols and signal processing mechanisms using software rather than hardware. The radio includes a special mixer to change the signal into an Intermediate Frequency (IF) based on the Radio Frequency (RF) if the configuration requires the use of SDR in the Rx mode, or if it requires conversion from the IF mode to the RF mode. It comprises a converter for Digital-to-Analogue Conversion (DAC) and Analogue-to-Digital Conversion (ADC), and an FPGA or digital signal processor for processing signal commands produced from the software of SDR [5], [25]. The GNU-Radio toolkit [5], [26] is the most widely used SDR software environment. The GNU-Radio is an open source software utilized for implementing SDR algorithms. This radio supports the C++ and Python programming languages and provides tools for signal processing. We deployed the MTD algorithm to implement our resilient communication system so that the constraints of GNU-Radio modules can be randomly changed. This will be explained further in subsequent sections.

SDR has become significant because it enables different signal processing mechanisms and radio communication protocols to be applied using software instead of hardware techniques [5]. The alignment of the digital signal processor for DAC and ADC is a part of the SDR programming tools. Several previous studies have emphasized supporting SDR programmability to enhance the efficiency and usage of the radio spectrum [5], [27]–[29]. SDR can be used in cognitive radio to enhance the service quality for secondary receivers by assessing the features of every group and picking the ones that enhance user quality of service [28]. In our research, we take a complimentary approach by using the SDR programmability to design a resilient radio communication system that can continue to operate normally despite being attacked.

III. RESILIENT WIRELESS COMMUNICATION ARCHITECTURE

Let us suppose a scenario where we have only one configuration (which never changes) for the communication link. In such a scenario, an attacker will have sufficient time to inspect the communication channel as the communication link properties do not change. Hence, the attacker can determine the weakness and then perform a successful attack.

Our method is based on using the MTD technique to achieve resilient wireless communications [3], [15]. In this approach, the SDR function is used to change the properties of the communication link between two nodes. Therefore, a successful attack on our proposed wireless communications architecture will be a challenging task. Let us assume that the attacker inspects the communications channel when configuration A is active. The attacker would then require some time to identify the vulnerability, plan, and thereafter, launch an attack. However, the attack is thwarted because the time will not be sufficient if the properties of the communication link are quickly changed within short time intervals from configuration A to configuration B, and thereafter, from configuration B to configuration C, and so on. By considering this methodology, the communications channel attains new properties every time an attack is launched.

The programmability of the SDR technology is utilized in our methodology to randomly alter the communication link properties so that the attacker would not succeed in disturbing the wireless communication. The configurability of SDR is used to utilize the MTD method in radio communication, thus altering the communication link properties between two nodes, as depicted in Fig. 1. One or more of the channel properties could be changed based on the type of communication and level of protection required.

MTD organizes the receiver and transmitter modules so that they can function with varying packet lengths; the modulation and frequencies can also be randomly altered to avoid identification and therefore prevent assaults [14], [30]. Our research has used two radio links: the first is the standby channel, and the second is the primary channel [7]. If an attack affects the active radio channel, the system will use the

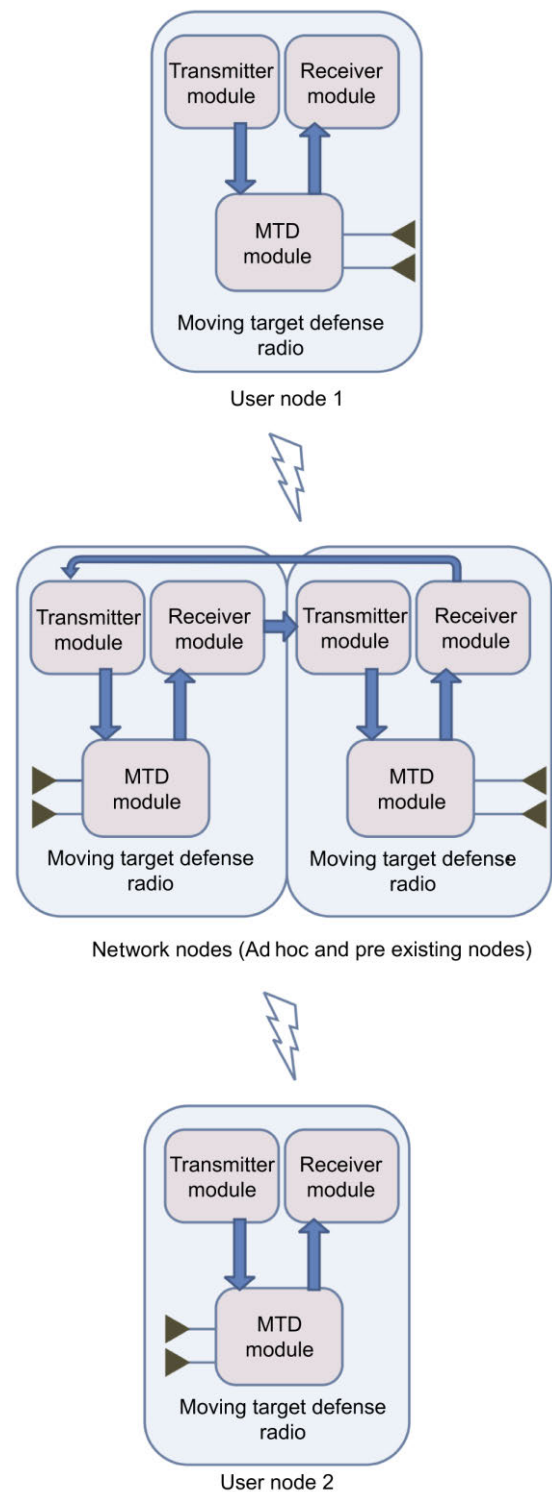


FIGURE 1. General architecture.

data delivered through the standby channel and consequently tolerate the attack.

An example of our approach deployed in a military tactical scenario is depicted in Fig. 2. In this example, two different radio channels are used for each link with respect

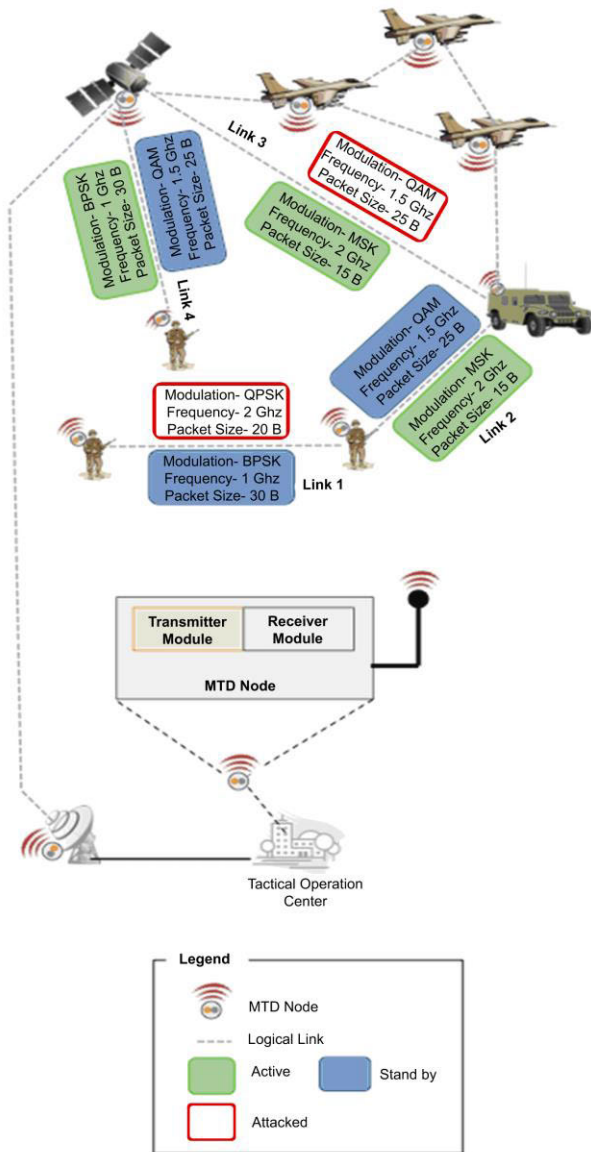


FIGURE 2. Example of a military tactical scenario system.

to packet size, signal frequency, and modulation. A 20-byte packet size, 2 GHz frequency, and Quadrature Phase Shift Keying (QPSK) modulation are used for the active channel of link 1, whereas a 30-byte packet size, 1 GHz frequency, and Binary Phase Shift Keying (BPSK) modulation are used for its standby channel. In this configuration, the system will tolerate an attack if the target is the active channel (red box), as the data provided by the standby channel (blue box) can be used to maintain the communication of link 1 during the attack.

IV. IMPLEMENTATION APPROACH

In this section, we describe our testbed which contains the transmitter, receiver, and the MTD module that will use SDR to randomly change the configuration during runtime [4], [5].

A. SOFTWARE DEFINED RADIO

Two SDR systems, a receiver and a transmitter were used in our testbed. On the transmitter side, a configuration program is deployed using GNU-Radio modules and Python. This program allows the user to automatically change the configuration for each communication cycle, as listed in Table 1. The same configuration table is utilized at the receiver side to configure the link after random intervals of time.

TABLE 1. Table of configuration.

Number of Configuration	Frequency	Modulation	Packet Length	Access Code
1	80 MHz	GFSK	1024 B	1000100010001 00010001000
2	120 MHz	GMSK	256 B	1000100010001 00010101010
3	90 MHz	GFSK	512 B	1000100010001 00010101111
4	130 MHz	GMSK	1024 B	1000100010001 00010001111
5	70 MHz	GFSK	256 B	1000100010001 11110001000

As discussed before, the resilient algorithm changes the configurations and the shuffling rate (reconfiguration time) automatically based on two parameters: 1) the key value; and 2) the iteration number. We utilized the Diffie–Hellman key exchange algorithm to maintain key confidentiality and prevent eavesdropping [34]. This key exchange algorithm uses symmetric encryption techniques. Both sides are required to agree on a base number (g) and a modulus (p). If a sender (Alice) and a receiver (Bob) want to exchange a key, both must use the same values (g) and (p), and they are both required to generate a random number. In this case, Alice’s random number is y , whereas Bob’s random number is x . Alice shall compute $(g^y \text{ mod } p)$ and send the result to Bob. Conversely, Bob shall compute $(g^x \text{ mod } p)$ and then send the result to Alice. Both Alice and Bob then compute the key value (Key) by finding $(g^{xy} \text{ mod } p)$ through multiplication of $(g^x \text{ mod } p)$ by $(g^y \text{ mod } p)$.

Configuration (i) signifies the type of configuration (configuration interval, packet length, and frequency) to be adopted during each iteration. For example, after computing the key (Key) on both sides, the communication process is started using the configuration; the reconfiguration time can be determined using the following two equations:

$$Config(i) = Key \cdot i \cdot \text{mod}N \quad (1)$$

$$Configuration_Time(i) = (Key + i) \cdot Tc \cdot \text{mod}N \quad (2)$$

where N is the number of configurations available, i denotes the communication cycle or iteration number, and Tc is the value for a predefined communication interval. Moreover, (1) shows that the configuration utilized defines the interval of reconfiguration. From (2), it is evident that the reconfiguration interval will change as the iteration number (i) and key value (Key) change. At this point it is important to note

how crucial the selection of the Diffie–Hellman algorithm modulus p is because it determines the keys to be used in our proposed procedure. In our testbed, we made $p = N$ (the number of configurations available). The specific values considered for N will be presented in Section V. However, in Section VI, we can see that the values for N can grow in an arbitrary way, i.e., by increasing the security strength (based on the Diffie–Hellman algorithm), and without affecting our proposal’s time complexity.

Fig. 3 illustrates the structural design and operations of the implemented resilient communications system.

Fig. 4 depicts the underlying algorithm. Initially, in the ResilientCommunicationService_SDRProcedure, the system starts with an operation message including a special key for defining the reconfiguration and configuration time denoted by T_c . The Diffie–Hellman key exchange algorithm is used in steps 3 to 9 to prevent from getting identified [14], [31]–[33]. The configuration is defined by each end-user and the timeframe to initiate the process of communication (Steps 10 to 12).

In the ResilientCommunicationService_Transmitter algorithm, each message is transferred via two channels at the transmitter side (i.e., standby and active channels), and each one of them uses a different configuration. On the receiver side, the ResilientCommunicationService_Receiver algorithm receives the data and interprets it via the standby and active channels.

B. JAMMING ATTACKS

On our own test bed, we experimentally demonstrated that the approach is feasible for tolerating multiple attacks. A jamming attack was introduced by transmitting a jamming signal to prevent the system from accurately receiving transmitted data. In the testbed on channel 3, the attacker launches a continuous signal with a frequency of 90 MHz using Gaussian frequency shift keying (GFSK) modulation, and a 512-packet length, as listed in Table 1. The jamming signal interfered through the transmitting channel configuration and transmitted a signal of attack to corrupt the data on the active channel. The system then immediately switched the data communication operations to the standby channel that was not affected by the attack as it used a different configuration.

V. ARCHITECTURE’S TOLERANCE TO ATTACKS

To analyze the overhead and performance of our proposed architecture, we will present the resilience of the approach analytically. At first, we present how to compute a successful attack probability by using the reconfiguration time. Second, we demonstrate the manner in which the probability value is changed by the reconfiguration time based on different time slots.

A. SUCCESSFUL ATTACK PROBABILITY

In our approach, we change the radio link configurations randomly to make it extremely difficult for attacks to occur.

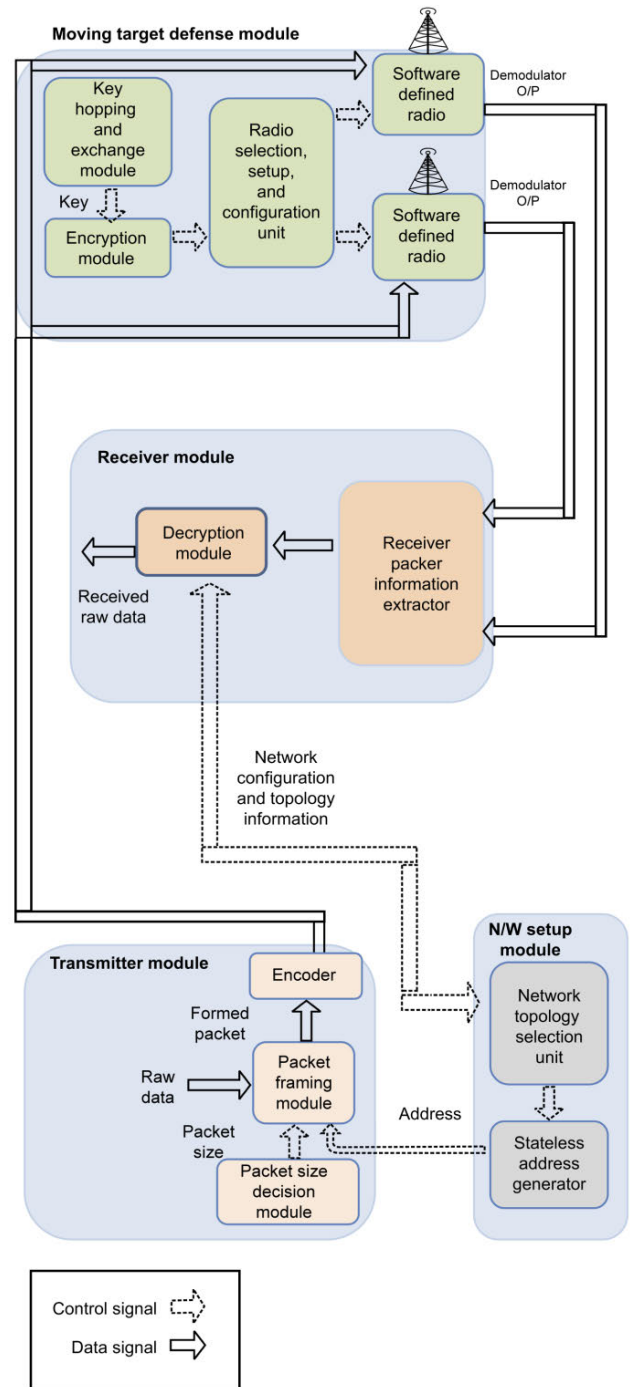


FIGURE 3. Resilient communication system architecture.

We need to set the parameter numbers for random selection to calculate a successful attack probability between reconfigurations. Table 2 lists the configuration parameters available for the random selection using SDR links and the number of options for each configuration. For instance, with a channel spacing value of 25 kHz, the operating value of the frequency ranges from 225 to 400 MHz. There are approximately 7,000 channels available in this case; and therefore, we can

```

Input: g: Base number
         p: Modulus
         N: Number of configurations available.
         {c1, c2, ..., cN}: Set of configurations available.
         Tc: Predetermined communication interval.
Output: None
Procedure ResilientCommunicationService_SDRProcedure
             (g, p, N, {c1, c2, ..., cN}, Tc)
1.   i = 1 // Communication cycle or iteration number
2.   While(true) do
       // Starting of Diffie-Hellman Key Exchange
       // Procedure
3.     y = getAliceRandomNumber( )
4.     x = getBobRandomNumber( )
       // Alice computes  $g^y \text{ mod } p$  and sends to
       // Bob.
5.     modulusAlice = getAliceComputedModulus( )
6.     SendTo(Bob, modulusAlice)
       // Bob computes  $g^x \text{ mod } p$  and sends to
       // Alice.
7.     modulusBob = getBobComputedModulus( )
8.     SendTo(Alice, modulusBob)
9.     key =  $g^{xy} \text{ mod } p$ 
       // Ending of Diffie-Hellman Key Exchange
       // Procedure
10.    config(i) = key · i · mod N
11.    configuration_time(i) = (key + i)·Tc·mod N
12.    StartCommunicationProcess(Tc, config(i),
        configuration_time(i), {c1, c2, ..., cN})
13.    i = i + 1
14.  End-of-while
End-of-procedure
    
```

```

Input: N: Number of configurations available.
         {c1, c2, ..., cN}: Set of configurations available.
         Tc: Predetermined communication interval.
Output: None
Procedure ResilientCommunicationService_Transmitter
             (N, {c1, c2, ..., cN}, Tc)
1.   While(there is data to be transmitted) do
2.     for each configuration a,be{c1,c2,...,cN} do
3.       if(Tc ≠ 0) then
         // Send message MSG over Active Link using
         // configuration a.
4.       SendMessage(MSG, Active_Link, a)
         // Send message MSG over Standby Link
         // using configuration b.
5.       SendMessage(MSG, Standby_Link, b)
6.       else
         // The Receiver and the Transmitter switch
         // to the standby communication link
         // according to the initial key exchange.
7.       TransitionStandbyLinkToActiveStatus( )
8.       SetupNewStandbyLink(c(i+1))
9.     End-of-if
10.  End-of-for
11.  End-of-while
End-of-procedure
    
```

FIGURE 4. Resilient communication system algorithm.

```

Input: N: Number of configurations available.
         {c1, c2, ..., cN}: Set of configurations available.
         Tc: Predetermined communication interval.
Output: None
Procedure ResilientCommunicationService_Receiver
             (N, {c1, c2, ..., cN}, Tc)
1.   While(there is data to be received) do
2.     for each configuration a,be{c1,c2,...,cN} do
3.       if(Tc ≠ 0) then
         // Receive message MSG over Active Link
         // using configuration a.
4.       MSG = ReceiveMessage(Active_Link, a)
         // Receive message MSG over Standby Link
         // using configuration b.
5.       MSG = ReceiveMessage(Standby_Link, b)
6.       else
         // The Receiver and the Transmitter switch
         // to the standby communication link
         // according to the initial key exchange.
7.       TransitionStandbyLinkToActiveStatus( )
8.       SetupNewStandbyLink(c(i+1))
9.     End-of-if
10.  End-of-for
11.  End-of-while
End-of-procedure
    
```

FIGURE 4. (Continued.) Resilient communication system algorithm.

TABLE 2. Number of possibilities.

Configuration	Possibilities
Frequency range	In the military range, we have 7000 channels.
Modulation scheme	We have at least six modulation schemes such as ASK, FSK, PSK, QAM, MSK, and OOK
Packet length	In our research we use: 128 B, 256 B, 512 B, 1024 B, and 2048 B
Access code	In our research, we use 24-bit length

select any one of them for each reconfiguration. Moreover, for each reconfiguration, there are six different modulation schemes available to select from, including Phase-Shift Keying (PSK), Frequency-Shift Keying (FSK), Amplitude-Shift Keying (ASK), Minimum-Shift Keying (MSK), On-Off keying (OOK), and Quadrature amplitude modulation (QAM).

For our experiment, we used the following packet length values: 2048 B, 1024 B, 512 B, 256 B, and 128 B. For synchronization, we selected the access code of the GNU-Radio as the transmitter. The receiver should have an identical access code value to receive the data correctly. Additionally, a 24-bit length access code was used in our implementation, which gave us a total of 224 options. In this case, the probability of selection for each code is 1 in 16,777,216.

A successful attack probability ($\Pr(A_s)$) depends on the number of possible configurations the attacker could try for each reconfiguration. Therefore, the probability in this case depends on the length of the reconfiguration interval because increasing the reconfiguration interval would enable

the attacker to perform additional attempts. Thus, $Pr(A_S)$ can be mathematically represented as:

$$Pr(A_S) = \frac{T_c}{Mod \cdot Freq \cdot Len} \quad (3)$$

T_c is the configuration interval, Mod is the number of available modulation schemes, $Freq$ is the number of available frequency channels, and Len is the number of possible packet lengths. For simplicity, we have excluded the number of possibilities for access codes in (3). We have also assumed that the reconfirmation time of the attacker is 1 ms.

As an example, the probability of a successful attack based on (3) is 0.0095 if our reconfiguration interval is 2000 ms, as depicted in Fig. 5. The figure clearly shows that the probability of a successful attack is less than 0.01 when the reconfiguration interval is less than 2 s. The probability is 0.1 if the time is less than 21 s. We can reduce the probability to approximately zero by considering a shorter reconfiguration interval.

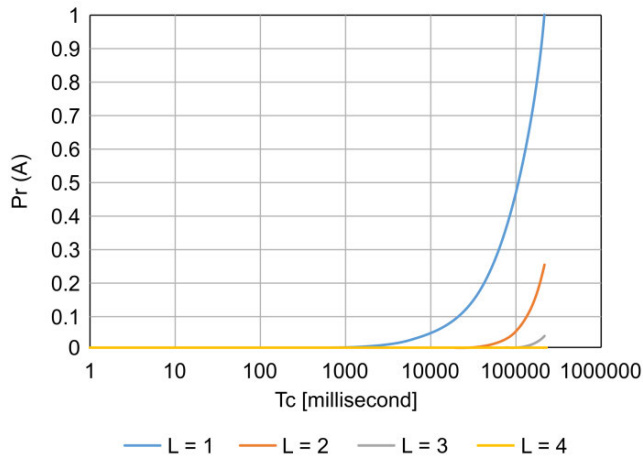


FIGURE 5. Successful attack probability with reconfiguration time T_c and multiple values for L .

We can use diversified and redundant communication links to enhance the resiliency of our system. In this scenario, we calculate the probability of a successful attack using the following equation:

$$Pr(A_L) = \left(\frac{Pr(A_S)}{L} \right)^L \quad (4)$$

where L represents the number of links being used redundantly. For instance, in the case of four redundant links being used to transmit data, the probability of a successful attack would be zero for the chosen reconfiguration time.

B. PROBABILITY OF A SUCCESSFUL ATTACK WITH SLOTTED RECONFIGURATION TIME

To further increase the resilience of the approach used in this study, we have distributed the time into consecutive slots. Here, we present the computation of the average probability during a time slot for a successful attack.

We divided the configuration interval (T_c) into multiple time slots (T_s) by assuming that the attacker would change the configuration for every time slot as he would be required to make a new attempt for each slot. The probability of a successful attack is presented as P ; the number of combinations for frequencies, packet lengths, and modulation schemes is N ; and the number of attacked time slots is represented by M .

Two possible attack scenarios are considered here. A random combination or serial combinations could be utilized by the attacker for every attempt. For each scenario, two different types of attacks are simulated: a jamming attack, where the used channel is jammed by the attacker and the configuration is altered when the user discovers the attack; and a scanning attack, where the entire channel is scanned by the attacker to gain access to the data while the end-user is unaware of the attack. In the scanning attack scenario, the configuration remains the same until the end of T_c .

C. RANDOM COMBINATIONS

1) RANDOM SCANNING ATTACK ANALYSIS

For the random scanning attack, the attacker uses a random configuration for every attempt regardless of the combinations used during past attempts. If the attack succeeds, this success lasts until there is a change in the configuration. For example, if the attack becomes successful during the first time slot, then P is equivalent to T_s/N . However, if the attack fails on the first attempt, then the average probability for the second time slot becomes $(N - 1)/N \cdot 1/N \cdot (T_s - 1)$.

In this regard, the projected probability of the attack is calculated as follows:

$$E [P_{random-scan}] = \frac{\sum_{i=1}^{T_s} \left(\frac{N-1}{N} \right)^{i-1} \left(\frac{1}{N} \right) (T_s - i + 1)}{T_s} \quad (5)$$

where i denotes the communication cycle or iteration number. Fig. 6 illustrates the probability of a successful random scanning attack for different combinations of time slots.

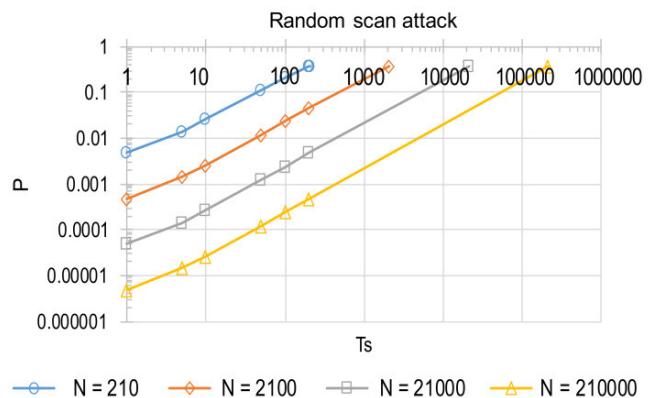


FIGURE 6. Probability of a successful random scanning attack using different combinations of time slots.

2) RANDOM JAMMING ATTACK ANALYSIS

In the random jamming attack, a random combination is utilized by the attacker for each attempt. However, because a legitimate user is capable of detecting the jamming attack after a successful attempt, the configuration is immediately changed for the next time slot. Equation (6) can be used to find the probability of a successful attack, as in this scenario, the attacker must randomly guess the configuration of each time slot as depicted in Fig. 7.

$$E[P_{random-scan}] = \frac{1}{N} \tag{6}$$

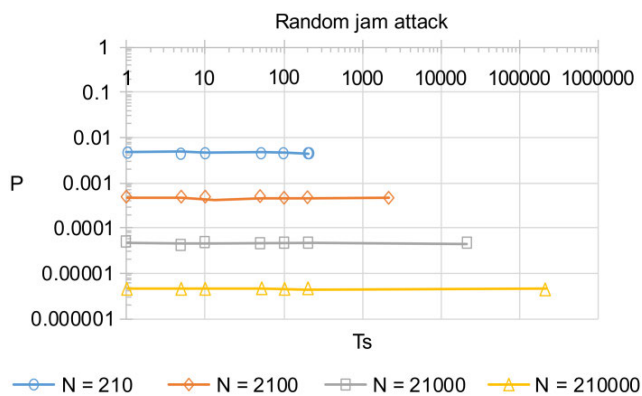


FIGURE 7. Probability of a successful random jamming attack using different combinations.

D. SERIAL COMBINATIONS

In this type of attack, the attacker attempts to use new configurations that have not been used before. For example, if the first combination attempted by the attacker used a frequency of 130 MHz, a packet length of 128 B, and GFSK modulation, then the second combination would use a frequency of 900 MHz, a packet length of 512 B and Gaussian minimum shift keying (GMSK) modulation.

1) SERIAL SCANNING ATTACK ANALYSIS

As the attack cannot be detected by the user, there would be no change in the configuration until the end of T_c . As the attack attempts are serial, the expected probability for the attack can be calculated using the following equation and the results are depicted in Fig. 8.

$$E[P_{serial-scan}] = \frac{\sum_{i=1}^{T_s} \left(\frac{1}{N}\right) (T_s - i + 1)}{\frac{T_s}{2}} = \left(\frac{1}{N}\right) \left(\frac{T_s + 1}{2}\right) \tag{7}$$

2) SERIAL JAMMING ATTACK ANALYSIS

We can use (5) to calculate the probability of success for the expected serial jamming attack. The length of the consistent success line is denoted by L_s , whereas N denotes the time interval of $L_s = N \cdot E[P_{serial-scan}]$. Moreover, the duration of the failure is represented as L_f , which is equivalent

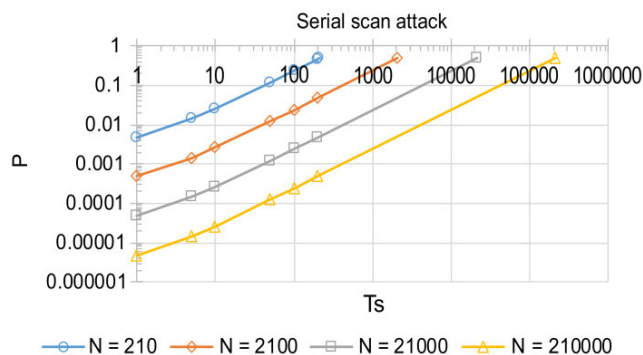


FIGURE 8. Probability of a successful serial scanning attack with different combinations.

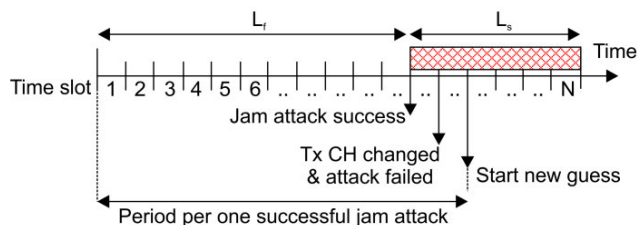


FIGURE 9. Diagram of a serial jamming attack.

to $N - L_s = N - N \cdot E[P_{serial-scan}] = N \cdot (1 - E[P_{serial-scan}])$, as illustrated in Fig. 9.

During the first time slot, immediately after the failed time duration L_f if a successful attack attempt takes place, this leads to a change in the configuration of the sender whereas the attacker will still be using the configuration from the previous channel that resulted in a successful attack. Hence, one significant successful attack occurs in the $L_f + 2$ time slot, which is calculated using the following equation:

$$E[P_{serial-jam}] = \frac{1}{L_f + 2} = \frac{1}{N(1 - E[P_{serial-scan}]) + 2} \tag{8}$$

Fig. 10 depicts the probability of a successful jamming attack for a combination of several numbers. It is evident that previously analyzed attacks do not distinguish random attacks from serial attacks. However, at this point, we are dealing with different types of attacks. For the sake of completeness, we have presented results of such attacks in their respective

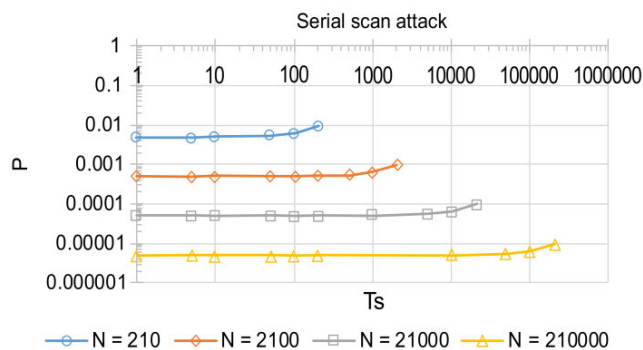


FIGURE 10. Probability of a successful serial jamming attack.

subsections, and Figures 7 and 10. Nevertheless, it is also important to mention the effective presence of differences in the probabilities associated with these attacks. To provide evidence, a difference has been calculated for one case ($N = 210$), which is illustrated in Fig. 11. In the following section we will see the presence of this behavior in the context of the execution times required by our algorithm for dealing with such attacks. These are the interesting features of our proposal as it provides evidence of a uniform behavior under different types of attacks.

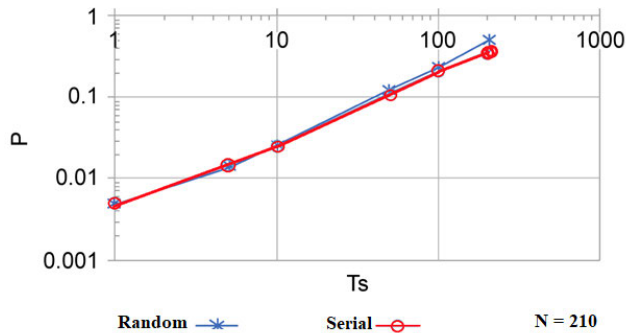


FIGURE 11. Serial vs. random scan attack.

Verification of all analytical results has been conducted through simulations that used the same parameter values as those we used in our analysis. There is less than 1% difference between the analytical results and the simulations.

VI. SOME COMPLEXITY ASPECTS

We conducted several experiments and they all indicate that our algorithm is capable of tolerating attacks targeting one of the communication links by using redundant diversified links that are not affected by the attack. Fig. 12 depicts an instance for the considered playground of the experiments dealing with the robustness against radio attacks for the random combinations’ scenario. The overhead and performance of the approach with consideration for execution time is summarized in Table 3. For our methodology, the overhead results from the key exchange and the random variations in the communication channels.

Subsections VI.A to VI.D discuss the determination of some complexity aspects starting from the equations presented in Section V. The main idea is to provide some theoretical elements that support the aspects regarding the execution time of our proposal under several attack scenarios.

TABLE 3. Overhead.

Execution time (in s)		Overhead percentage (time)
without resilience	with resilience	with resilience
100	110	10%
150	164	9%
200	216	8%

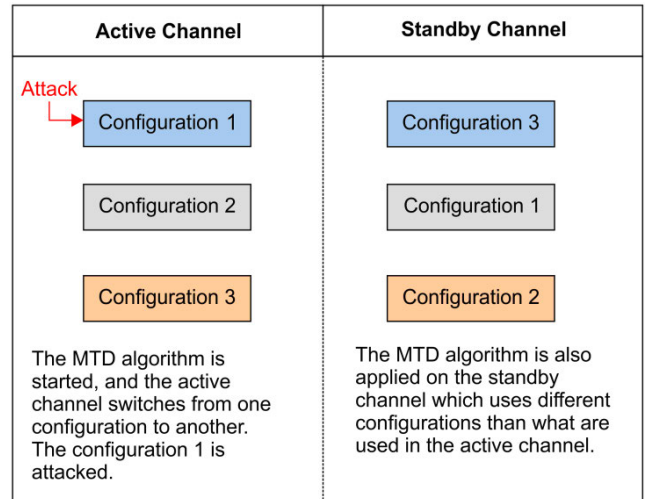


FIGURE 12. Scenario: Robustness against radio attack.

A. ABOUT THE REQUIRED CONFIGURATION TIME

The perspective we are going to follow now is based on the fact that it is possible to determine an appropriate configuration time in terms of the probability of an attack and the number of redundant links. Configuration time is a preponderant aspect to be considered determining the execution time of our approach. In more formal terms, the discussion in this section regards how configuration time behaves; that is, increases in the attack probability and the number of used redundant links. The probability of a successful attack, $Pr(A_L)$, is given by (3) and (4):

$$Pr(A_L) = \left(\frac{Pr(As)}{L} \right)^L = \left(\frac{1}{L} \cdot \frac{Tc}{Mod \cdot Freq \cdot Len} \right)^L \quad (9)$$

L describes the number of used redundant links. Note that specific values for Mod (the number of available modulation schemes), Freq (the number of available frequency channels), and Len (the number of possible lengths) describe the configuration of a specific communication system [34], as mentioned earlier in Section V. Therefore, they can be unified into a constant k given by $k = Mod \cdot Freq \cdot Len$. The aforementioned equation is then solved for Tc , and owing to constant k , from an asymptotic point of view can be despised, we directly obtain the following new equation with its corresponding restrictions:

$$Tc(Pr(A_L), L) = L Pr(A_L)^{\frac{1}{L}} Pr(A_L) \in (0, 1], \quad L > 0 \quad (10)$$

Therefore, the configuration interval Tc is described as a function of the successful attack probability, $Pr(A_L)$, and the number of used redundant links, L . Fig. 13 depicts the graph for $Tc(Pr(A_L), L)$. Fig. 14 depicts some contour lines for (10). In both cases, we consider $0 < Pr(A_L) \leq 1$.

Let us suppose $0 < Pr(A_L) \leq 1$, while the number of links being used redundantly increases, or formally,

$L \rightarrow \infty$. Then, we have:

$$\lim_{L \rightarrow \infty} Tc(Pr(A_L), L) = \lim_{L \rightarrow \infty} L Pr(A_L)^{\frac{1}{L}} = \infty \quad (11)$$

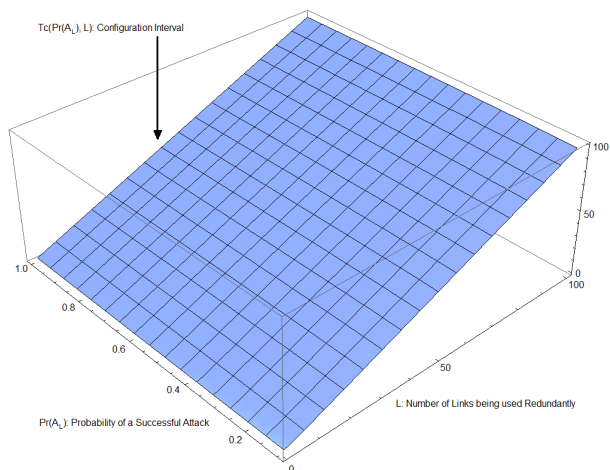


FIGURE 13. Configuration time T_c in terms of probability of a successful attack and the number of links being used redundantly.

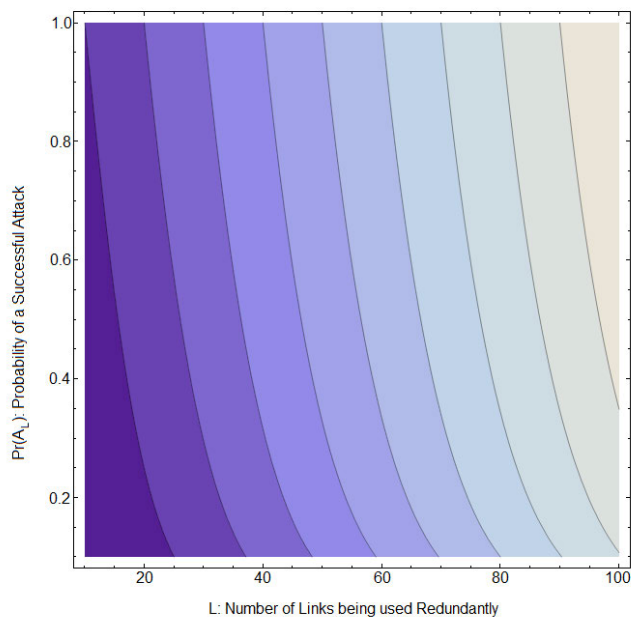


FIGURE 14. Some contour lines for configuration time T_c in terms of probability of a successful attack and the number of links being used redundantly.

Such a limit is consistent with the practice because as the number of links being used redundantly increases, the configuration time T_c will also increase. This is also consistent with the behavior of the algorithms shown in Section IV. A visual inspection of Fig. 13 and Fig. 14 suggests a linear relationship for the configuration time T_c and a direct and unique dependence on the value of L (number of links being used redundantly). This observation is formally verified by evaluating the following limit by assuming $0 < \text{Pr}(A_L) \leq 1$ and $L \rightarrow \infty$:

$$\lim_{L \rightarrow \infty} \frac{T_c(\text{Pr}(A_L), L)}{L} = \lim_{L \rightarrow \infty} \frac{L \text{Pr}(A_L)^{\frac{1}{L}}}{L} = 1 \quad (12)$$

Therefore, the linear function L is tight bound for the configuration time $T_c(\text{Pr}(A_L), L)$. It has been verified from an asymptotic point of view that the linearity of the required reconfiguration time can be calculated using the following equation:

$$T_c(\text{Pr}(A_L), L) = \Theta(L) \quad (13)$$

The important observation here is that the configuration time behaves linearly with respect to these parameters: 1) attack probability, and 2) the number of links being used redundantly. As mentioned earlier, the configuration time contributes to the execution time of our approach, and the fact that we have identified its linear complexity enhances the notion of the time efficiency of our approach, and more specifically, a low temporal efficiency in some of its conforming elements. Furthermore, the practical application of (10) lies in the sense that for a specific communication system, configuration time can be properly adjusted by considering the expected attack probability and the number of available redundant channels.

B. RANDOM SCANNING ATTACK TIME COMPLEXITY (RANDOM COMBINATIONS)

Let $E_{upper}[P_{random-scan}]$ be an upper bound for (5) obtained by substituting the term $(N - 1)/N$ by 1, then:

$$\begin{aligned} E_{upper}[P_{random-scan}] &= \frac{\sum_{i=1}^{T_s} \left(\frac{1}{N}\right) (T_s - i + 1)}{T_s} \\ &= \frac{1}{N} \left(\frac{T_s}{2} + \frac{1}{2}\right) \end{aligned} \quad (14)$$

Here, i is the number of iterations in our proposed algorithm, T_s is the number of time slots in which the whole configuration time T_c has been distributed, $E[P_{random-scan}]$ is an attack probability (5), and N is the number of combinations for frequencies, packet lengths, and modulation schemes.

Now, we solve for T_s from (14). Therefore, we get:

$$T_s = 2N \cdot E_{upper}[P_{random-scan}] - 1 \quad (15)$$

Suppose $E_{upper}[P_{random-scan}]$ is the same probability given for (5), that is:

$$E_{upper}[P_{random-scan}] = E[P_{random-scan}] \in (0, 1] \quad (16)$$

Moreover, by assuming $E_{upper}[P_{random-scan}]$ as a constant term, we find that (15) computes an upper bound for the time slots T_s required for a fixed attack probability for the random scanning attack, as described in Subsection V.C.1. Clearly, T_s is also expressed in terms of combinations for frequencies, packet lengths, and modulation schemes.

Now, to have a more precise idea about the algorithm's time complexity both sides of (15) should be multiplied by the number c of executed steps in each one of the algorithm's iterations $i = 1, 2, 3, \dots, T_s$. As previously observed, our algorithm depends on some computations such as the Diffie-Hellman Key Exchange procedure which in turn depends on the Discrete Logarithm Problem. As mentioned in

Section III.A, the Diffie–Hellman algorithm requires as input a value p corresponding to a considered modulus. Choosing a right modulus impacts on the execution time; however, as pointed out by several researchers, $O(p^{1/2})$ is the commonly accepted time for solving such a problem [35]. Thereafter, suppose $c = p^{1/2} = N^{1/2}$ and $E_{upper}[P_{random-scan}]$ is a fixed constant, then by a straight calculation on the right side of (15), the time complexity for dealing with a random scanning attack, $T_{random-scan}$, is given by:

$$T_{random-scan} = O\left(N^{3/2}\right) \quad (17)$$

C. SERIAL SCANNING ATTACK TIME COMPLEXITY (SERIAL COMBINATIONS)

We solved for T_s from (7), and obtained:

$$T_s = 2N \cdot E[P_{serial-scan}] - 1 \quad (18)$$

Equation (18) computes the time slots T_s required for a fixed attack probability for the serial scanning attack, as described in Subsection V.D.1, which depends on the number N of combinations for frequencies, packet lengths, and modulation schemes. Both sides of (18) are multiplied by the number c of executed steps in each one of the algorithm's iterations $i = 1, 2, 3, \dots, T_s$. Then, such as was done in the previous subsection, let us suppose $c = p^{1/2} = N^{1/2}$ and $E[P_{serial-scan}]$ is a fixed constant, then the time complexity for dealing with a serial scanning attack, $T_{serial-scan}$, is given by:

$$T_{serial-scan} = O\left(N^{3/2}\right) \quad (19)$$

D. SERIAL JAMMING ATTACK TIME COMPLEXITY (SERIAL COMBINATIONS)

By substituting (7) into (8), we obtain:

$$E[P_{serial-jam}] = \frac{2}{2N - T_s + 3} \quad (20)$$

Now, if we solve for T_s from (20), then, we obtain:

$$T_s = 2N + 3 - \frac{2}{E[P_{serial-jam}]} \quad (21)$$

Equation (21) computes the time slots T_s required for a fixed attack probability for the serial jamming attack, as described in Subsection V.D.2. Both sides of (21) are multiplied by the number c of executed steps in each one of the algorithm's iterations $i = 1, 2, 3, \dots, T_s$. Then, again if we assume $c = p^{1/2} = N^{1/2}$ and $E[P_{serial-jam}]$ is a fixed constant, the time complexity for dealing with a serial jamming attack, $T_{serial-jam}$, can be obtained as:

$$T_{serial-jam} = O\left(N^{3/2}\right) \quad (22)$$

VII. CONCLUSION

This article presents a resilient wireless communication architecture based on MTD modules. Using SDR in wireless communication enables us to dynamically program the radio network and provide two differentiated yet redundant channels. The active channel is the primary channel used for

data transmission whereas the standby channel is utilized in case the active channel is successfully attacked. Each configuration channel is randomly changed after every reconfiguration. The configuration link is defined with properties such as modulation type, packet size, and link frequency. The resilience of this approach has been experimentally validated, and the probability of a successful attack has been quantified. Our assessment demonstrated that the presented resilient methodology can tolerate a wide range of wireless attacks including jamming attacks. Moreover, by properly setting the reconfiguration time and shuffling change rate, we can reduce the probability of successful attacks to approximately zero. The proposed approach exhibited linear time behavior in some of its elements. In this specific study, we confirmed that the configuration time is effectively linear in terms of the available redundant links and attack probability. Moreover, we proposed three time complexity bounds for the cases dealing with random scanning, serial scanning, and serial jamming attacks. In all these three cases, the algorithms' complexity is sub-quadratic with respect to the number of combinations for frequencies, packet lengths, and modulation schemes. Therefore, our resilient wireless communications architecture has shown evidence of efficient performance. Future research might consider the possibility of utilizing our approach of resilient/redundant wireless communications for industrial control systems, specifically for the communication between controllers and actuators, or for wireless sensor-controllers.

REFERENCES

- [1] K. Gai, K.-K.-R. Choo, M. Qiu, and L. Zhu, "Privacy-preserving content-oriented wireless communication in Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3059–3067, Aug. 2018, doi: 10.1109/JIOT.2018.2830340.
- [2] F. Almoualem, P. Satam, J.-G. Ki, and S. Hariri, "SDR-based resilient wireless communications," in *Proc. Int. Conf. Cloud Autonomic Comput. (ICCAC)*, Sep. 2017, pp. 114–119.
- [3] V. Casola, A. De Benedictis, and M. Albanese, "A moving target defense approach for protecting resource-constrained distributed devices," in *Proc. IEEE 14th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2013, pp. 22–29.
- [4] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for moving target defense network protection," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.
- [5] C. Y. Chen, F. H. Tseng, K. D. Chang, H. C. Chao, and J. L. Chen, "Reconfigurable software defined radio and its applications," *J. Appl. Sci. Eng.*, vol. 13, no. 1, pp. 29–38, 2010.
- [6] M. Acharya and D. Thuente, "Intelligent jamming attacks, counterattacks and (counter) 2 attacks in 802.11 b wireless networks," in *Proc. OPNET-WORK Conf.*, Washington DC, USA, 2005, pp. 1075–1081.
- [7] A. D. Wood, J. A. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2007, pp. 60–69.
- [8] Y. Guan and X. Ge, "Distributed secure estimation over wireless sensor networks against random multichannel jamming attacks," *IEEE Access*, vol. 5, pp. 10858–10870, 2017.
- [9] S. A. Naseem, R. Uddin, O. Hasan, and D. E. Fawzy, "Probabilistic formal verification of communication network-based fault detection, isolation and service restoration system in smart grid," *J. Appl. Log.-IFCoLog J. Logics Appl.*, vol. 5, no. 1, pp. 321–365, 2018.
- [10] L. Chhaya, P. Sharma, G. Bhagwatikar, and A. Kumar, "Wireless sensor network based smart grid communications: Cyber attacks, intrusion detection system and topology control," *Electronics*, vol. 6, no. 1, p. 5, Jan. 2017.

[11] Ž. Gavrić and D. Simić, "Overview of DOS attacks on wireless sensor networks and experimental results for simulation of interference attacks," *Ingeniería e Investigación*, vol. 38, no. 1, pp. 130–138, Jan. 2018.

[12] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu, "Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2431–2439, Sep. 2017.

[13] E. O. O. T. President, N. Science, and T. Council. (Dec. 2011). *Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program*. [Online]. Available: <http://www.whitehouse.gov/>

[14] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proc. HotSDN*, Helsinki, Finland, 2012, pp. 127–132.

[15] G. Dsouza, S. Hariri, Y. Al-Nashif, and G. Rodriguez, "Resilient Dynamic Data Driven Application Systems (rDDAS)," in *Proc. Int. Conf. Comput. Sci.*, 2013, pp. 1929–1938.

[16] C. Tunc, F. Fargo, Y. Al-Nashif, S. Hariri, and J. Hughes, "Autonomic resilient cloud management (ARCM) design and evaluation," in *Proc. Int. Conf. Cloud Autonomic Comput.*, Sep. 2014, pp. 44–49.

[17] J. Pacheco, C. Tunc, and S. Hariri, "Design and evaluation of resilient infrastructures systems for smart cities," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Sep. 2016, pp. 1–6.

[18] L. Xie, P. Smith, M. Banfield, H. Leopold, J. P. G. Sterbenz, and D. Hutchison, "Towards resilient networks using programmable networking technologies," in *Active and Programmable Networks* (Lecture Notes in Computer Science), vol. 4388. Berlin, Germany: Springer-Verlag, 2005, pp. 83–95.

[19] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 52–59, Apr. 2015.

[20] O. Erdene-Ochir, M. Minier, F. Valois, and A. Kountouris, "Toward resilient routing in wireless sensor networks: Gradient-based routing in focus," in *Proc. 4th Int. Conf. Sensor Technol. Appl.*, Jul. 2010, pp. 478–483.

[21] K. Mazur, B. Ksiezopolski, and R. Nielek, "Multilevel modeling of distributed denial of service attacks in wireless sensor networks," *J. Sensors*, vol. 2016, Jun. 2016, Art. no. 5017248.

[22] T. Jamal, M. Alam, and M. M. Umair, "Detection and prevention against RTS attacks in wireless LANs," in *Proc. Int. Conf. Commun., Comput. Digit. Syst. (C-CODE)*, Mar. 2017, pp. 152–156.

[23] A. Cetinkaya, H. Ishii, and T. Hayakawa, "An overview on denial-of-service attacks in control systems: Attack models and security analyses," *Entropy*, vol. 21, no. 2, p. 210, 2019.

[24] Y. Z. Jembre and Y.-J. Choi, "Distributed and jamming-resistant channel assignment and routing for multi-hop wireless networks," *IEEE Access*, vol. 6, pp. 76402–76415, 2018.

[25] R. Muzammil, M. S. Beg, and M. M. Jamali, "A dynamically reconfigurable transceiver for software defined radio," *Int. J. Comput. Appl.*, vol. 76, no. 17, p. 8887, 2013.

[26] F. Ge, C. J. Chiang, Y. M. Gottlieb, and R. Chadha, "GNU radio-based digital communications: Computational analysis of a GMSK transceiver," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.

[27] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[28] M. Timothy Masonta, M. Mzyece, and N. Ntlatlapa, "Spectrum decision in cognitive radio networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1088–1107, 3rd Quart., 2013.

[29] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," in *Proc. IEEE Int. Workshop Mobile Multimedia Commun. (MoMuC)*, Nov. 1999, pp. 3–10.

[30] M. Mohsin and R. Prakash, "IP address assignment in a mobile ad hoc network," *Computer*, vol. 33, no. 7, pp. 49–55, Jul. 2000.

[31] Y. Qian, K. Lu, and D. Tipper, "A design for secure and survivable wireless sensor networks," *IEEE Wireless Commun.*, vol. 14, no. 5, pp. 30–37, Oct. 2007.

[32] K. Weniger, "PACMAN: Passive autoconfiguration for mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 507–519, Mar. 2005.

[33] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[34] F. Almoaleem, "SDR-Based Resilient Wireless Communications," M.S. thesis Dept. Elect. Comput. Eng., Univ. Arizona, Tucson, AZ, USA, 2017. [Online]. Available: <http://hdl.handle.net/10150/625342>

[35] I. F. Blake and T. Garefalakis, "On the complexity of the discrete logarithm and Diffie–Hellman problems," *J. Complex.*, vol. 20, nos. 2–3, pp. 148–170, 2004.



AMANY ALSHAWI received the M.S. degree in computer information systems from the University of Miami and the Ph.D. degree in information technology from George Mason University. She is currently the Founding Director of the King Abdulaziz City for Science and Technology (KACST) Women Program, an initiative to increase the role of KACST women in local, regional, and international science and technology domains. In 2014, she was among the leading researchers who established the National Center for Cybersecurity Technology, KACST. She joined KACST, in 2011, where she is leading the Cryptography Research Group and the National Center for Electronics and Communication. She serves as a Board Member in multiple associations, among them are the ArabWIC Saudi Chapter and the Princess Nora University Distinguished Saudi Women Prize. In 2016, she founded the IEEE WIE Riyadh Affinity Group and is still chairing the committee. Prior to joining KACST, she held multiple positions at Prince Sultan University—College for Women, among them an Assistant Professor with the CIS Department, a Cooperative Training Program Coordinator, and a Research Center Director.



PRATIK SATAM received the M.S. and Ph.D. degrees from The University of Arizona, in 2015 and 2019, respectively. He is currently a Research Assistant Professor with The University of Arizona. His current research interests include autonomic computing, cybersecurity, cyber resilience, secure critical infrastructures, and cloud security.



FIRAS ALMOUALEM received the master's degree in electrical and computer engineering from The University of Arizona. He currently works as a Communication and Network Systems Engineer at The Aerospace Corporation.



SALIM HARIRI (Senior Member, IEEE) received the M.Sc. degree from The Ohio State University, in 1982, and the Ph.D. degree in computer engineering from the University of Southern California, in 1986. He is currently a Professor with the Department of Electrical and Computer Engineering, The University of Arizona, where he is also the Director of the NSF Center for Cloud and Autonomic Computing. His current research interests include autonomic computing, cybersecurity, cyber resilience, secure critical infrastructures, and cloud security.