A Survey of Man In The Middle Attacks

Mauro Conti, *Senior Member, IEEE*, Nicola Dragoni, and Viktor Lesyk

***Abstract*—The Man-In-The-Middle (MITM) attack is one of the most well known attacks in computer security, representing one of the biggest concerns for security professionals. MITM targets the actual data that flows between endpoints, and the confidentiality and integrity of the data itself. In this paper, we extensively review the literature on MITM to analyse and categorize the scope of MITM attacks, considering both a reference model, such as the open systems interconnection (OSI) model, as well as two specific widely used network technologies, i.e., GSM and UMTS. In particular, we classify MITM attacks based on several parameters, like location of an attacker in the network, nature of a communication channel, and impersonation techniques. Based on an impersonation techniques classification, we then provide execution steps for each MITM class. We survey existing countermeasures and discuss the comparison among them. Finally, based on our analysis, we propose a categorisation of MITM prevention mechanisms, and we identify some possible directions for future research.**

***Index Terms*—Man-In-The-Middle (MITM) attack, MITM defence techniques, MITM classification, security.**

I. Introduction

TODAY, almost each aspect of our life may be associated with the usage of Internet or cellular networks. For instance, we use online home banking, online entertainment and shopping, social networks, and so on. All these online services store or transfer user's sensitive information, which represents a key target for hackers. Besides individuals, hackers target enterprises and organisations, leading to big economical loss. In this new world of "people and things always connected" by means of the Internet, it is very common to daily read about successful attacks to connected things and online services. One of the most successful attacks is known as Man-In-The-Middle (MITM), which results in gaining control over end-users' transferred data.

The name *Man-In-The-Middle* is derived from the basketball scenario where two players intend to pass a ball to each other, while one player between them tries to seize it [1]. MITM attacks are sometimes referred to as *bucket brigade attacks* or *fire brigade attacks* [1]. Those names are derived from the fire brigade operation of dousing off the fire by passing buckets from one person to another, between the water source and the fire. MITM attack is also known as: Monkey-in-the-middle attack, Session hijacking, TCP hijacking, TCP session hijacking [1].

To the best of our knowledge, the term *Man-In-The-Middle attack* was firstly mentioned by Bellovin et al. in [2] with reference to [3]. After that paper, the term MITM has become a reference attack in the security community, counting an increasing number of citations every year. To mention only a few, in Verizon's data investigation report [4] and in [5], researchers showed that MITM attack is one of the most common type of security attacks. Frankel et al. [6] described MITM attack as one of the major threats against network security. Such publications alongside with previously specified awarenesses clearly show that MITM attack has become more and more important and widespread, in principle being able to affect every online interaction.

Nevertheless, today there is no publication which gives an extensive overview of the MITM attack for each Internet layer. Efforts have been done to describe the problem within one specific protocol, like MITM attacks on Address Resolution Protocol (ARP) [7], or within a specific technology like Bluetooth [8]. Also, there are surveys which do not go sufficiently into details of each MITM attack, but provide a partial coverage of the attack's topology. For example, in [9], authors proposed categorisation of the MITM attack, which do not cover all known attacks. Also, researchers did not provide execution steps of attacks, but rather gave abstract description of them. Moreover, in prevention mechanisms, authors listed solutions without any explanation of used approaches.

In the MITM attack, the common scenario involves: two endpoints (victims), and a third party (attacker). The attacker has access on communication channel between two endpoints, and can manipulate their messages. The MITM attack can be visualised as shown on Figure 1. In particular, victims try to initialise secure communication by sending each other public keys (messages M1 and M2). Attacker intercept M1 and M2, and as a return sends its public key to the victims (messages M3 and M4). After that, victim1 encrypts its message by attacker's public key, and sends it to victim2 (message M5). Attacker intercepts M5, and decrypts it using known private key. Then, attacker encrypts plaintext by victim2's public key, and sends it to victim2 (message M6).

Manuscript received September 20, 2015; revised February 18, 2016; accepted March 27, 2016. Date of publication March 29, 2016; date of current version August 19, 2016. The work of M. Conti was supported by a Marie Curie Fellowship funded by the European Commission under the agreement PCIG11-GA-2012-321980. This work is also partially supported by the EU TagItSmart! Project H2020-ICT30-2015-688061, the EU-India REACH Project ICI+/2014/342-896, the TENACE PRIN Project 20103P34XC funded by the Italian MIUR, and by the projects "Tackling Mobile Malware with Innovative Machine Learning Techniques," "Physical-Layer Security for Wireless Communication," and "Content Centric Networking: Security and Privacy Issues" funded by the University of Padua. The work of V. Lesyk was supported by an Erasmus Mundus Scholarship funded by the European Commission for the NordSecMob project (Masters in Security and Mobile Computing), and the Internationalisation project of the M.Sc. in Computer Science of the University of Padua.

M. Conti is with the Department of Mathematics, University of Padua, Padua 35131, Italy (e-mail: conti@math.unipd.it).

N. Dragoni and V. Lesyk are with DTU Compute, Technical University of Denmark (DTU), Kgs. Lyngby 2800, Denmark, and also with the Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Örebro, Sweden.

Digital Object Identifier 10.1109/COMST.2016.2548426

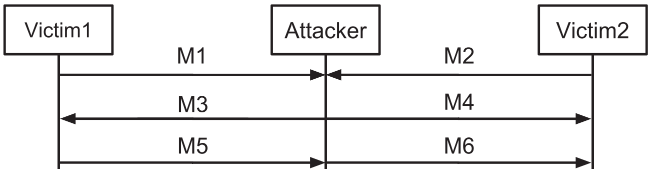


Fig. 1. Exchanged messages in a typical MITM attack.

As a result, the attacker has convinced both victims that they use secure channel, but in reality it has access to all encrypted messages.

A. Methodology

We used bottom-up approach in order to get the better understanding of the current status of the MITM attack. Firstly, we reviewed almost all literature that mentions MITM attack, which was published no earlier than 2000. Then we started to classify articles, papers, books, based on used protocols, and theirs contribution (such as new cryptographic prevention method, or new detection approach). Later, we found that some approaches are modifications of more older one, so we extended scope by including more older literature. At this point we started to focus on the most mentioned the attacks, and prevention methods, and based on them we created main categorisation of the MITM attack.

B. Contribution of the Paper

In this paper, we provide a thorough survey of the MITM attack with focus on the OSI model, and on specific mobile networking technologies, i.e., GSM and UMTS. We chose directly this scope, since classes of the MITM attack correlate with layers of OSI model; GSM is one of the most spread network, which covers more than 90% of the world population [10], and was not designed to be MITM resistant; UMTS is a good example of technology evolution with legacy support. Further, in paper we classify MITM attacks based on several parameters, namely: location of an attacker in the network, nature of a communication channel, and impersonation techniques. Next, we use the impersonation techniques classification as a reference classification and we go into details for each category providing attack algorithms and categorising prevention mechanisms.

To the best of our knowledge, this is the first extensive study of the MITM attack. When we compare our work with the existing body of research, we see the following. Clark et al. [11] executed one of the most significant surveys of defence schemes against SSL/TLS MITM attack. Authors reviewed the spectrum of issues concerning trust model between certified authorities and browsers. Similarly, in [12]–[14], researchers carried out small studies of detecting and defeating mechanisms of SSL/TLS MITM attack. Saxena et al. [15] collected proposals, which prevent MITM attack on GSM and UMTS networks. Thus, there is no previous work in the literature that covers MITM attacks across each layer of the OSI model, classifies MITM, and categorises MITM defence approaches. Our work fills this gap, by providing MITM attack survey over the period 1992-2015.

C. Organization

The rest of this paper is organised as follows. Section II overviews MITM attack and introduces three different MITM classifications, namely based on location of an attacker in the network, nature of a communication channel, and impersonation techniques. Sections III, IV, V, and VI focus on each subclass of the impersonation techniques classification, that we use as a reference classification throughout the paper. In particular, Section III describes spoofing attacks and its connection to MITM, discusses ARP, DNS, DHCP, IP spoofing-based MITM attack, and provides algorithms of their execution. Section III lists prevention mechanisms against ARP, DNS, DHCP, and IP spoofing. Section IV gives a brief introduction of the SSL/TLS protocol, presents SSL/TLS MITM attack, describes defence solutions against SSL/TLS MITM. Section V discusses the BGP protocol and the IP hijacking attack, overviews BGP MITM attack and provides BGP MITM prevention mechanisms categorisation. Section VI gives a short description of GSM, UMTS networks, False Base Station (FBS) attack, introduces FBS spoofing-based MITM attack, and lists defence approaches against FBS spoofing-based MITM. Finally, Section VII draws some conclusions on the MITM attack and proposed solutions, providing also a brief insight on future research.

II. Man-In-The-Middle attack

Man-In-The-Middle (MITM, also abbreviated in the literature as MIM, MiM, MitM, or MITMA) is a kind of attack where a malicious third party secretly takes control of the communication channel between two or more endpoints. The MITM attacker can intercept, modify, change, or replace target victims' communication traffic (this distinguishes a MITM from a simple eavesdropper). Moreover, victims are unaware of the intruder, thus believing that the communication channel is protected.

MITM attack can be executed in different communication channels such as GSM, UMTS, Long-Term Evolution (LTE), Bluetooth, Near Field Communication (NFC), and Wi-Fi. The attack targets are not only the actual data that flows between endpoints, but also the confidentiality and integrity of the data itself [16]. In particular, MITM attack aims to compromise [5], [17]:

- Confidentiality, by eavesdropping on the communication.
- Integrity, by intercepting the communication and modifying messages.
- Availability, by intercepting and destroying messages or modifying messages to cause one of the parties to end communication.

We can identify at least three ways of characterising MITM attacks, leading to three different categorisations:

1) MITM *based on impersonation techniques*.
2) MITM *based on the communication channel in which the attack is executed*.
3) MITM *based on the location of attacker and target in the network*.

The first classification (MITM *based on impersonation techniques*) divides approaches according to how attackers convince

TABLE I
MITM ATTACK ON DIFFERENT LAYERS OF OSI MODEL AND TYPES OF CELLULAR NETWORKS

		MITM Attacks
OSI Layer	Application	BGP MITM, DHCP spoofing-based MITM, DNS spoofing-based MITM
	Presentation	SSL/TLS MITM
	Transport	IP spoofing-based MITM
	Network	
	Data Link	ARP spoofing-based MITM
Cellular networks	GSM	FBS-based MITM
	GSM/UTMS	

victims that they are valid endpoints. In our opinion, this criterion allows to fully focus on the attack itself and this is the main reason we will follow this classification in the rest of the paper. Moreover, it is useful to collect weak and strong sides of algorithms across different communication channels. Thus, researchers can observe which methods are better against MITM attacks, and then apply such methods to update weak protocols, and create new MITM resistant algorithms. From this perspective, MITM attacks may be divided into:

- **Spoofing-based MITM** is an attack in which the attacker intercepts a legitimate communication between two hosts by the means of spoofing attack, and controls transferred data, while hosts are not aware of a middle man existence. In some cases (e.g., DNS spoofing), attacker spoofs devices between victims, in other cases (e.g., ARP spoofing), the attacker spoofs directly victim's devices.
- **SSL/TLS MITM** is a form of active network interception, where the attacker inserts itself into the communication channel between two victims (usually victim's browser and the web server). Then attacker establishes two separate SSL connections with each victim, and relays messages between them, in a way such that both are unaware of the middleman. This setup enables the attacker to record all messages on the wire, and even selectively modify the transmitted data.
- **BGP MITM** is an attack, which is based on the IP hijacking, but where attacker makes stolen traffic to be delivered to the destination. Thus, traffic goes through attacker's Autonomous Station (AS), where it can be manipulated.
- **False Base Station based (FBS-based) MITM** is an attack, where third party forces victim to create connection with a fake Base Transceiver Station (BTS), and then, using it, attacker manipulates victims' traffic.

Another way to categorise MITM attack is to consider the *communication channel in which attack is executed*. In Table I, we list MITM attacks across OSI layers and cellular networks. Each layer enforces different approaches to provide security. Nevertheless, neither of them is free from MITM attacks.

On Blackhat Conference, Ornaghi et al. [9] presented last classification - *based on the location of attacker and target in the network*. They divided MITM attacks in: *local area network* (attacker and target are in a same network), *from local to remote (through a gateway)* (attacker and target are in a different networks, attack is executed through a gateway), and *remote* (attacker and target are in a different networks, attacker uses remotely controlled services for attack execution).

In their work, authors considered STP mangling as MITM attack, but it is not real MITM, since the attacker is able to receive only *unmanaged* traffic [9]. Also, we consider port stealing as a part of the ARP spoofing attack; ICMP redirection and IRDP spoofing as ICMP spoofing, which in its turn is a part of IP spoofing attack; traffic tunnelling and route mangling as a DNS spoofing and IP hijacking (BGP MITM). The following list concludes this classificaiton:

- *Local area network*: ARP/DNS spoofing-based MITM.
- *From local to remote (through a gateway)*. ARP/DNS/DHCP/IP spoofing-based MITM.
- *Remote*. DNS spoofing-based MITM, BGP MITM.

The remaining part of the paper is organised according to the classification of MITM based on impersonation techniques. Thus, next four sections will cover each subclass of that categorisation. Last section VII will sum up and discuss the main contribution of the paper.

III. SPOOFING-BASED MITM ATTACK

In this section, we describe spoofing attack and its connection to the MITM attack. Then, we overview ARP, DNS, DHCP, IP spoofing-based MITM attacks (sections III-A, III-C, III-E, III-G). Finally, based on our categorisation, we list prevention mechanisms against these attacks (sections III-B, III-D, III-F, III-H, respectively).

Spoofing attack is an impersonation of a device or a user in the network by a malicious party. Spoofing attack is used as an opening for other attacks, such as DoS, MITM, or session hijacking attacks [18]. There are several types of spoofing attack that malicious parties can use: ARP spoofing, DNS spoofing, DHCP spoofing, IP spoofing. In all types of spoofing, attackers use same protocols' weakness - a lack of source and destination messages authentication.

Some papers [18]–[29], describe spoofing as the first step or one of steps for the execution of a MITM attack. Other papers [30]–[33] name spoofing as an equivalent of the MITM. However, in [34]–[40] authors define *spoofing-based MITM attack* as a spoofing attack, which leads to the MITM attack. In this paper we will stick to the last naming.

Spoofing-based MITM is an attack in which the attacker intercepts a legitimate communication between two hosts by the means of a spoofing attack, and controls transferred data, while hosts are not aware of a middle man existence.

In our work, we review ARP, DNS, DHCP, and IP spoofing, where each of them leads to the spoofing-based MITM attack. In the following sections, we go into details of each attack, and describe prevention mechanism against them.

A. ARP Spoofing-Based MITM Attack

Network devices use ARP protocol to map network addresses to Media Access Control (MAC) addresses. ARP is crucial in LAN communications, because each frame that leaves a host must contain a destination MAC address [25]. ARP is a trusted protocol and was not designed to cope with malicious hosts [41].

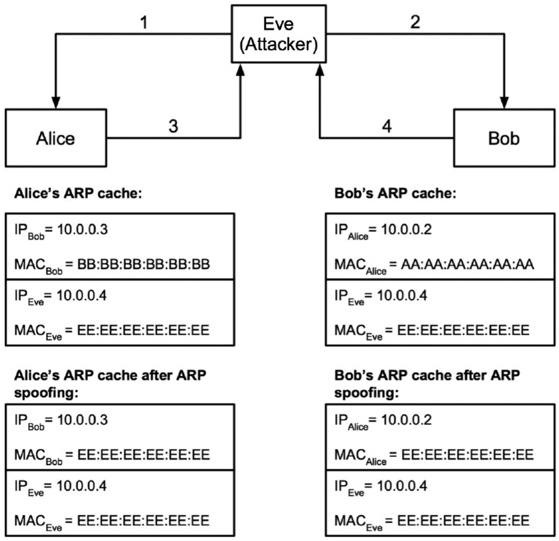

Eve
(Attacker)
1
2
Alice
3
4
Bob
Alice's ARP cache:
IP_{Bob}= 10.0.0.3
MAC_{Bob} = BB:BB:BB:BB:BB:BB
IP_{Eve}= 10.0.0.4
MAC_{Eve} = EE:EE:EE:EE:EE:EE
Bob's ARP cache:
IP_{Alice}= 10.0.0.2
MAC_{Alice} = AA:AA:AA:AA:AA:AA
IP_{Eve}= 10.0.0.4
MAC_{Eve} = EE:EE:EE:EE:EE:EE
Alice's ARP cache after ARP spoofing:
IP_{Bob}= 10.0.0.3
MAC_{Bob} = EE:EE:EE:EE:EE:EE
IP_{Eve}= 10.0.0.4
MAC_{Eve} = EE:EE:EE:EE:EE:EE
Bob's ARP cache after ARP spoofing:
IP_{Alice}= 10.0.0.2
MAC_{Alice} = EE:EE:EE:EE:EE:EE
IP_{Eve}= 10.0.0.4
MAC_{Eve} = EE:EE:EE:EE:EE:EE

Fig. 2. ARP spoofing-based MITM attack.

By modifying victims' local ARP cache table (adding, updating cache entries), the attacker can associate a malicious host's MAC address with IP of a target host. Consequently, the attacker can launch DoS attack, perform MITM attack and gain access to confidential information [22]. ARP spoofing attack may be divided into two types [42]: cheating the gateway, and cheating the host of the internal network.

When a host requires to communicate with another host in the same network, whose MAC address is unknown, it broadcasts an ARP Request to all hosts inside the network. Only the host with the announced IP is expected to issue a Reply, which includes its MAC address. However, when ARP cache is managed in a dynamic mode, cache entries can be easily fabricated by forged ARP messages, since proper authentication mechanism is missing [43]. At the same time, the source machine saves the IP to MAC entry in its local cache, so the next time communication can be speeded up, by avoiding the broadcasts.

ARP is a stateless protocol and has a lack of security in caching system. A number of papers [27], [28], [30], [34], [38], [44] showed in practice how to use those weaknesses for achieving MITM attack. Suppose, we have next network: the attacker Eve (IP = 10.0.0.4, MAC = EE:EE:EE:EE:EE:EE), victim Alice (IP = 10.0.0.2, MAC = AA:AA:AA:AA:AA:AA), and victim Bob (IP = 10.0.0.3, MAC = BB:BB:BB:BB:BB:BB). The next steps are necessary to complete ARP spoofing-based MITM attack (see Figure 2):

1) Eve sends an ARP Reply message to Alice, which says that IP: 10.0.0.3 has MAC address: EE:EE:EE:EE:EE:EE. This message will update Alice's ARP table.
2) Eve also sends an ARP Reply message to Bob, which says that IP: 10.0.0.2 has MAC address: EE:EE:EE:EE:EE:EE. This message will update Bob's ARP table.
3) When Alice wants to send a message to Bob, it will go to Eve's MAC address EE:EE:EE:EE:EE:EE, instead of Bob's BB:BB:BB:BB:BB:BB.
4) When Bob wants to sends a message to Alice, it will also go to Eve.

B. ARP Spoofing Defence Mechanisms

Abad et al. [22] carried out one of the most significant ARP security surveys. They analysed the schemes for detecting and preventing ARP spoofing attacks, and specified requirements for an ideal solution. Oh at. al. [43] showed new defence approach, and made comparison of previously proposed methods. Most of schemes may be classified from two perspectives: way of implementation (cryptographic, voting-based, hardware), and location of solution (server-based, and host-based). In Table II, we combine two categorisations and compare solutions. In next sections, we list ARP spoofing prevention mechanisms, focusing on schemes that prevent ARP from MITM attacks.

1) Detection of ARP Spoofing: Carnut et al. [19] proposed an architecture based on switched networks, which does not require special software to be installed on the network hosts. Instead, approach delegates the task of detection to one or more detection station. Their experiments showed that the architecture detects ARP attacks properly, without generating false positives (alarms that turn out to be not a part of attacks). However, attackers could hide behind the volume traffic and remain undetected for reasonably long periods. Also, authors have not made their software widely available.

ARP-Guard [58] and ARPDefender system [59] are commercial products, which use a sensor-based architecture to detect and localise several internal network attacks, including ARP spoofing. Solutions analyse the information from the LAN and Simple Network Management Protocol (SNMP) sensors, and in case of attack the management system alerts administrators.

Arpwatch [60] is an open source program that helps to monitor Ethernet traffic activity in the network and maintains a database of Ethernet/IP address pairings. When suspicious (IP, MAC) pairing change occurs the network administrator receives an alert. This tool is very lightweight and widely available, but it depends on the network administrators' ability to differentiate non-malicious events and ARP spoofing attacks, and to take appropriate measures when the attack occurs. Hou et al. [61] proposed similar approach, in which they added inspection module, and developed Intrusion Detection System (IDS) Snort.

Belenguer et al. [62] introduced low-cost embedded IDS, which is able to detect and prevent ARP spoofing attacks automatically and efficiently, but requires to be plugged into a switch or a hub. Device has two firmware variants that offer different functionality: reactive and proactive. In reactive mode the IDS scans all packets flowing through the network. In proactive the IDS periodically refreshes the IP to MAC mappings of all active hosts to repair them. Also, authors pointed out that IDSs (e.g., ARP-Guard [58], ARPDefender system [59], Arpwatch [60], Snort [61]) tend to generate a great amount of false positives. Moreover, IDSs' ability to detect ARP spoofing is limited [63], as they may not be able to detect all of the attack forms.

Ramachandran et al. [18], [26] proposed a scheme for detecting ARP spoofing through the mismatch of the ARP Request/Response packets and the (IP, MAC) pairs of TCP SYN. The destination (IP, MAC) pairs in the TCP SYN packets is the same as reported in the ARP message, only in case of

TABLE II
COMPARISON OF ARP SPOOFING PREVENTION APPROACHES

Approach	Category	Protocol	Concerns/Problems
S-ARP [45]	C	ARP*	Performance overhead, is not feasible for a wireless network.
Gouda et al. [46]	C/S	ARP*	Single point of failure, DoS.
CLL [47]	C	ARP, DHCP	Requires unnecessary time to select the appropriate algorithm (authors were considering management of large scale Ethernet network) [48].
Goyal et al. [20]	C	ARP*	Single point of failure.
TARP [24]	C/S	ARP	Performance overhead.
P-ARP [49]	C	ARP	DoS, slows down the network throughput to an unacceptable level.
Demuth et al. [50]	S	ARP	Does not work with wireless nodes.
Kwon et al. [51]	S	ARP	Uncompleted.
Ortega et al. [25]	S/HA	ARP	Uncompleted.
Pansa et al. [52]	S/H	ARP, DHCP*	Single point of failure, DoS, DHCP compatibility.
Trabelsi et al. [23]	H	ARP	Difficult to decide the level of trustiness and importance of each host.
Philip et al. [32]	H	ARP	No static IP addresses support, works only with Linksys routers.
Anticap [53]	H	ARP	Applicable only to UNIX systems.
WinPcap [54]	H	ARP	Vulnerable to ARP packet spoofing using RAW socket [55].
ASA [43]	C/H	UDP, ARP	UDP authentication.
MR-ARP [35]	V/H	ARP	DoS.
EMR-ARP [39]	V/H	ARP	Requires too much computational time.
GMR-ARP [40]	V/H	ARP	Additional overhead.
DAI [56], [57]	HA	ARP	Partially works with wireless nodes, can only be used with certain switch types.

C—Cryptography, S—Server-Based, H—Host-Based, V—Voting; HA—Hardware, *-Protocol Should Be Modified.

the spoofing attack data differs. However, if ARP DoS attacks are continuously generated to probe the network, this approach would create a heavy traffic on the LAN.

Carnut et al. [19] suggested another method, which delegates the detection of ARP spoofing to a specialised detection/test stations. Later, Trabelsi et al. [64] improved their approach. In both cases detection works in two phases: detection of hosts with enabled IP packet routing, and detection of ARP cache poisoning attack. In the first phase, test station generates trap ICMP echo Request packets. If host sends back the received trap ICMP packets, then it has enabled IP packet routing, and considered suspicious. In the second phase, test host analyses traffic generated by a suspicious host to identify whether or not it has performed ARP spoofing against the target host in the network.

Kalajdzic and Patel [36] proposed a system, where instead of relying on the test host, each host performs detection by itself. They introduces two methods for detection ARP spoofing: reverse ARP poisoning with active IP probing and IP probing with CAM table poisoning. However, methods are week against certain MITM attacks, and detection accuracy depends on the size of probing.

Barbhuiya et al. [65] suggested another method, which checks the integrity and authenticity of ARP Replies using a combination of digital signatures and one time passwords. The proposed IDSs run on hosts and ensure the genuineness of the (IP, MAC) pairings by an active verification mechanism. The IDS sends verification messages to probe requests upon receiving ARP Requests and ARP Replies. Verification messages include digital signatures, which help to identify ARP origin.

Recently, Song et al. [66] introduced a detection scheme for ARP spoofing attack by the means of a routing trace, named DS-ARP. The detection module periodically keeps the ARP cache table under surveillance and checks changed items. Once a change in the ARP cache table is identified, the DS-ARP determines whether an ARP spoofing attack has taken place through a routing trace. If the attack occurred then the protection module converts the (IP, MAC) pairs' information back to the valid states. It prevents ARP spoofing attacks by changing the link type from a dynamic state to a static state.

2) Cryptographic Solutions: S-ARP [45] is a backward compatible extension to ARP that relies on public-key cryptography to authenticate ARP Replies. All hosts create public and private key pairs during the initial contact with the network, and send them with signed certificates to the Authoritative Key Distributor (AKD). Using public-key cryptography anyone can identify whether the transmitted request is from a valid user. Thus, the ARP spoofing attack can be prevented.

Gouda et al. [46] proposed another approach that resolves the security problems of MAC and IP pairs within the Ethernet. The scheme consists of a security server, which uses two protocols Invite-accept and Request-reply to connect all hosts inside the network. Each host checks the IP and MAC addresses from the security server, thereby enhancing security. However, the solution is not practical, as it requires to change the ARP protocol implementation of every host. Another disadvantage is that the secure server represents a single point of failure in the network, and becomes an obvious target for DoS attacks.

Goyal et al. [20] presented enhanced S-ARP with lower computational cost. The scheme is based on a combination of digital signatures, one time passwords, and hash chains. To avoid additional computations, solution allows to use same digital signature several times over multiple ARP Replies, during a predefined time period. To authenticate client to the untrusted server, solution uses Lamport Hashes [67] (one time password system), which enhances security significantly in comparison to S-ARP. Although, since proposal still uses AKD it has single point of failure.

Lootah et al. [24] introduced another method T-ARP to reduce the computational cost of S-ARP by employing the concept of tickets (centrally generated (IP, MAC) address mapping attestation). This approach uses a Local Ticket Agent

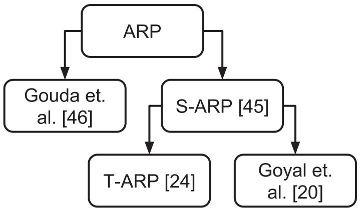


Fig. 3. Cryptographic protocols evolution.

(LTA), and a Key Management Server (KMS) to issue a public key to obtain the (IP, MAC) pair from the ticket. This approach is backward compatible with existing ARP, but it is susceptible to replay attacks and adds performance overhead. Figure 3 shows evolution of cryptographic approaches.

Another solution is CLL [47], a layer 2/3 security extension, which provides authentication and confidentiality to the hosts in the LAN by safeguarding all layer 2 traffic including ARP and DHCP handshakes. CLL employs public key cryptography to identify all hosts based on their (IP, MAC) address pairs. Hosts authenticate each other, exchange cryptographic parameters, and negotiate symmetric session keys to protect their following unicast packets with a message authentication code and an optional cipher.

P-ARP [49] is a new scheme, which is based on ARP authentication, but with few changes: P-ARP uses in addition magic number, nonce, and hash function HMAC, which produces the authentication data. In order to hide the target IP address in the APR Request message, hash function creates nonce, and HMAC values. The defect is the ineffectiveness against ARP DoS attacks, and solution slows down the overall network throughput to an unacceptable level in practice.

3) Voting-Based Solutions: Nam et al. [35] proposed MR-ARP, which is the first voting-based ARP spoofing resistant protocol. If an ARP Request or Reply message arrives and declares a new MAC address for an IP address, MR-ARP queries the corresponding machine (machine with old MAC address), and checks if new IP address is still used by that machine (this process is based on Antidote's [68] method, which is later discussed). In case when the MR-ARP machine receives the ARP Request or Reply declaring an (IP, MAC) mapping for a new IP address, it requests neighbour machines to vote for the new IP address. For this mechanism, the voting can be fair only when the voting traffic rates of the responding machines are almost the same. This condition can be satisfied in the Ethernet, but may not be valid in the 802.11 network, due to the traffic rate adaptation based on the signal-to-noise ratio (SNR).

Name et al. tried to overcome the limitation of MR-ARP by introduction of EMR-ARP [39]. The new protocol improves the voting procedure through the incorporation of computational puzzles, besides authors also achieved:

- mitigation of ARP spoofing attacks in wired or wireless LAN;
- backward compatibility with existing ARP;
- minimal infrastructure upgrade cost;
- incremental deployability;
- automatical configuration for newly joining nodes.

On the other hand, EMR-ARP requires too much computational time from devices. Improvements have been proposed in 2013 by means of the GMR-ARP [40] protocol, namely:

- increased the fairness of voting compared to MR-ARP, due to dropping too early Reply packets and determining voting related parameters analytically;
- decreased voting traffic overhead, which is lower than in previous voting protocols (MR-ARP, EMR-ARP);
- overcomed the condition of the computational power, by avoiding the computational puzzle;
- in contrast to MR-ARP, GMR-ARP can protect upgraded machines when the wired nodes and wireless nodes coexist in the same subnet.

However, this approach would create additional overhead in wireless mesh networks, especially in large-scale, since the voting requests are issued in broadcast [69].

4) Hardware Solutions: Some switches implemented a feature called Dynamic ARP Inspection (DAI) to provide additional security. DAI safeguards the network from many commonly known ARP spoofing-based MITM attacks [57]. It ensures only valid ARP Requests and Responses are forwarded. The Ethernet switch monitors the validity of the received ARP packet based on the trusted (IP, MAC) mapping database. However, this database is either manually managed or dynamically managed through DHCP snooping (see Section III-F). Also, this approach may not be effective, if the ARP spoofing occurs among the wireless nodes connected via the same Access Point (AP) [56].

5) Server-Based Solution: Antidote [68] is a non-cryptographic solution, which attempts to prevent ARP spoofing by contacting and giving a higher priority to the previous owner of a given IP address in the case of MAC conflicts. However, Antidote cannot prevent spoofing for a new IP addresses if a malicious ARP Reply arrives first [45].

Kwon et al. [51] proposed similar to Gouda et al. [46] approach. They suggested to use an agent, which retrieves genuine (IP, MAC) pairs from a host and forwards them to the manager to construct reliable (IP, MAC) mappings. Then the manager implements monitoring based on the mappings and licensed hosts.

Demuth et al. [50] introduced a solution, which requires installation of three additional servers in the network: a gateway protection, client protection, and server protection. Gateway protection creates vaccines for a valid (IP, MAC) mappings. These vaccines are passed to the network endpoints using client protection. The server protection plays role of monitor, and checks for any active ARP spoofing. In case when it founds the attack, the client protection provides vaccines to users in the network. The drawback of the proposal besides the maintenance of three additional machines is that it can not work with the wireless networks.

Ortega et al. [25] proposed method that can be used to prevent ARP spoofing attacks in small office LANs. The scheme consists of a server and a switch with the OpenWrt firmware installed. The server listens to ARP Requests in the network and responds with an (IP, MAC) mapping from the ARP cache, while switch blocks all ARP messages. However, authors have not described how the server collects the correct (IP, MAC) mappings, so that it may generate correct Replies to the incoming ARP Requests.

6) Host-Based Solutions: Tripunitara et al. [70] proposed a middleware host-based approach of asynchronous detection and prevention of ARP spoofing attacks. The solution depends on duplicates in order to detect the attacks. Thus, if host is already spoofed, or under DoS attack, system will not secure it. Moreover, the solution is not practical, as it requires Streams based protocol stack [71] and changes on all hosts in the network.

Anticap [53] is a kernel patch for various UNIX-based operating systems that prevents ARP spoofing attacks. The solution rejects ARP updates, which contain a MAC address that differs from the current table entry for that IP address. Still Anticap does not work in dynamic DHCP-enabled networks and is available for a limited number of operating systems.

Trabelsi et al. [23] introduced a stateful ARP cache management mechanism based on a fuzzy logic. The stateful ARP cache, unlike existing and stateless ARP cache, does not update ARP cache entries upon receiving ARP Requests. Instead, fuzzy logic controller aggregates various host properties to detect malicious hosts.

Philip et al. [32] showed how to prevent ARP spoofing in wireless LAN by implementing the defence mechanism in the AP. The basic idea is as follows: the AP constructs a list of correct (IP, MAC) address mappings by monitoring DHCP ACK messages or referring to the DHCP leases file. Then AP blocks all ARP packets with a false mappings based on the constructed list. The drawbacks include that it does not support hosts that have static IP addresses, and it was designed to work only with Linksys routers.

Pansa and Chomsiri [52] proposed a revision of the DHCP definition. They suggested to include authentication of network devices and inclusion of mapping information between IP and MAC addresses. Although, flawless in concept, it would cause serious compatibility problems, since DHCP's legacy.

WinPcap [54] is a library that captures and filters ARP packets. When ARP Response packet is received and the cache requires the update, it firstly compared against the correct (IP, MAC) address pairs. In case when they differ, network administrator will get warning about spoofing attack, and update will not occur.

Agent-based Secure ARP cache management (ASA) [43] is a host-based, cryptographic approach, which essentially blocks the unauthenticated exchange of ARP Requests and Replies among the hosts. ASA filter blocks all of ARP messages inside the network, but instead uses datagram protocol (UDP). Hosts send UDP packets, which contain (IP, MAC) pairs encrypted by a symmetric key. Then, local ASA agents update the ARP cache table. The proposed technique poses no compatibility problems, and overcomes ARP spoofing weaknesses. Also, authors pointed out that authentication of the UDP packet exchange will make system more secure.

C. DNS Spoofing-Based MITM Attack

DNS is a hierarchical naming system for the Internet based on an underlying client-server architecture [72]. The main function of a DNS server is to perform url-resolution (a process of translating a url or a domain name). For instance, *portalen.dtu.dk* into a physical IP address, such as 130.225.90.135. DNS servers and domain names are organised hierarchically in terms of top-level domains, subordinate, and lower-level domains, respectively *dk*, *dtu*, *portalen* in our example.

One of the most prominent and dangerous attacks against DNS is DNS spoofing [73], which is executed via cache poisoning (DNS spoofing quite often named as DNS poisoning). DNS service uses cache system for improving performance, but it has various weak sides. DNS spoofing results in storage by DNS resolver the invalid or malicious mappings between symbolic names and IP addresses. DNS spoofing may be categorised into [74], [75]:

1) hijacking or sniffing packets in the process of query response (between the recursive DNS and authoritative DNS);
2) cache poisoning through the birthday attack [76];
3) hacking on authorised DNS.

Further in this section, we discuss only first category of the DNS spoofing attack, since second category is fully cryptographic problem, and third category may include number of methods.

To execute DNS spoofing, attackers need to override local DNS's routing entries with fraud data, which will lead victim to use rogue server. DNS renew its cache depending on TTL of entries, thus from time-to-time it asks other DNSs for updates. Attacker may use this for the execution of the DNS spoofing attack. There are two methods of poisoning the cache:

- insert rogue DNS server in the network, which will produce fraud data (may lead to spreading fraud data not only to target DNS, but further to neighbour DNSs);
- send fake DNS answer right before real DNS will send valid one (by default DNS accepts first answer, and discards further automatically [77]).

Let us consider the next example (based on Kaminsky method [78]): we have the network of two name servers, attacker Eve, and victim Alice. Also, let us suppose that local DNS do not have the address query, which is asked for, and the Authoritative Name Server (ANS) is deployed by the attacker. Then to execute DNS spoofing Eve should perform the next steps (see Figure 4):

1) Eve generates a query for the DNS server to fetch the IP of a specific website (dtu.dk).
2) Local DNS server do not have the address, then query is forwarded to the ANS to resolve it.
3) The ANS reply with fake information (IP of dtu.dk is 204.2.22.23), optionally adding few more fields. Local DNS saves data into cache for the time TTL.
4) Alice generates a query to find the IP address of the dtu.dk.
5) Compromised DNS respond with 204.2.22.23, and Alice will be directed to attacker's server.

At this point, Eve has access only on one endpoint, which uses her rogue server. To extend attack to DNS spoofing-based MITM, the attacker should implement on rogue server phishing website or service, which will communicate with the original one.

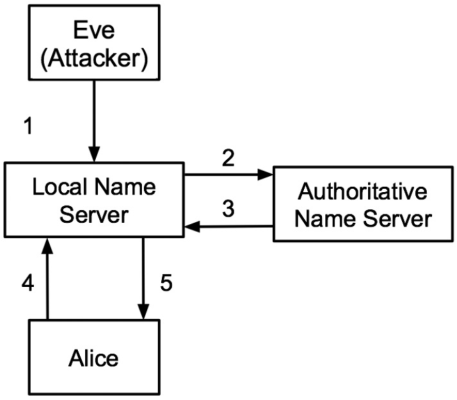


Fig. 4. DNS spoofing attack.

Our example shows scheme with insertion of rogue DNS in the network, but let say that Eve is not owner of the ANS. Then she can send spoofed message to local DNS right after it requests for data from the ANS. Thus, local DNS will overwrite caches and will block answer from valid ANS, as a protection from replay attack.

D. DNS Spoofing Defence Mechanisms

Herzberg et al. [79] performed a critical study of the prominent defence mechanisms against DNS spoofing, and proposed taxonomy of them. Based on their study and categorisation, in the following sections we provide updates to state of the art, by considering unmentioned methods. Section III-D1 shows overview of the DNS forgery detection, Section III-D2 describes entropy increasing mechanisms, and Section III-D3 presents cryptographic solutions. Recently, Bai et al. [37] suggested a new way of defence, which we propose to consider as a new category *artificial neural networks*, see Section III-D4.

1) Detection of DNS Spoofing: Anax [80] is a real-time solution that uses machine learning techniques. Testing results showed very low false positive rate (0.6% of all new resource records), and a high detection rate (91.9%). Cache Poisoning Detection System (CPDS) [75] is another real-time scheme, which has lower dependency, higher security, and applicability in comparison to Anax.

Other approaches make use of collaborative peer-to-peer (P2P) networks. The basic ideas behind these methods are similar to ARP voting-based protocols (see Section III-B3). Approaches validate authenticity of a DNS Responses, by distributing the DNS requests across hosts in the network, or by consulting a set of trusted peers, and then the majority to gives an answer.

Park et al. presented one of the first solutions named CoDNS [81]. Later, Poole et al. [82] showed method based on it, but with improved security. However, Kaminsky introduced new type of DNS cache poisoning [78], and demonstrated that both proposals are not strong enough. Sun et al. [83] presented DepenDNS protocol, but Alfardan et al. [84] discovered that it does not protect against the DNS spoofing attack. Another approach is DoX [21], where cache updating is based on the trust between resolvers.

2) Entropy Increasing Mechanisms: Entropy increasing mechanisms are solutions, which add more randomness to DNS packets, in order to entangle the injection of invalid DNS Responses. During the last years several proposals have been published, the most notably: Source Port Randomisation (SPR) [85], source/destination IP address Randomisation (IPR), DNS 0x20 encoding [86], WSEC-DNS [87], DNS-cookies [88]. Nevertheless, production of significant extra entropy to DNS requests will not protect from the DNS spoofing, the attacker may still be able to poison *high value* domain names [89]. Therefore, it still exists the need of alternative or additional technique of defence.

3) Cryptographic Solutions: DNS cryptographic solutions are similar to previously described ARP cryptographic solutions (see Section III-B2), they also use cryptography as a main countermeasure against spoofing attacks.

TSIG [90] is a cryptographic solution that involves authentication based on shared secret keys installed at both endpoints (requester and server), and one way hashing. SIG(0) [91] is another proposal that uses public key authentication, where the public keys are stored in DNS as KEY Resource Records (RRs), and a private key is stored at the signer. SIG(0) in comparison to TSIG involves relatively expensive asymmetric operations, while TSIG uses keyed hash authentication codes, which are relatively inexpensive to compute. Recently, Shrivastava et al. [92] published more detailed comparison of symmetric key usage for DNS.

DNS Security extensions (DNSSEC) [93] addresses the cache poisoning vulnerability in DNS. It provides data integrity and origin authenticity via cryptographic digital signatures over DNS resource records [94]. All messages from DNSSEC servers are digitally signed by public keys. By checking the signature, a DNSSEC resolver is able to validate if the information originated from a legitimate server, and if it identical to the data on the authoritative DNS server [95]. However, both servers and resolvers must use the DNSSEC protocol to maintain data origin authenticity, and integrity. Also, DNSSEC does not provide protection from buffer overruns, DoS attacks, nor provide confidentiality [96]. Moreover, DNSSEC can not be actually deployed in a short period of time [29].

Bernstein proposed DNSCurve [97] as an alternative to DNSSEC. The solution uses high-speed elliptic curve cryptography, and simplifies the key management problem that affected DNSSEC. DNSCurve implements per-packet encryption for all DNS queries, while DNSSEC does not. Moreover, DNSCurve's algorithm for public-key encryption is much faster and has smaller keys than the RSA algorithm used in DNSSEC. Anagnostopoulos et al. [98] gave detailed comparison of two methods.

S-DNS [83] is a proposal that is based on Identity-Based Encryption (IBE) [99] key management scheme. S-DNS is a simple security solution with low computation and communication overheads. It targets the different DNS query interaction models from iterative, recursive, and caching schemes. The method decreases the success probability of DNS spoofing and provides a backward compatibility.

4) Artificial Neural Network Solution: Recently, Bai et al. [37] described an Artificial Neural Networks (ANN) defence

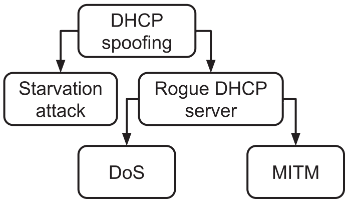


Fig. 5. DHCP spoofing classification.

against DNS spoofing attacks. Researchers managed to construct a 3-layered ANN: input, hide, output. The input consists of three parameters: ANSWER, AUTH, ADD, which present: number of authority RRs, answer RRs, and additional information. The network output layer corresponds to 10 possible categories, which predict the reliability of packets from the least reliable (0) to the most reliable (10). Packets below category 5 are considered forged, and will be discarded, while the others are considered right and are accepted. After training, the neutral network generates weights for these DNS Response packet fields. After testing configurations the performance testing showed excellent results. The average identifying ratio is 98% for both valid and forged packets.

E. DHCP Spoofing-Based MITM Attack

DHCP is a protocol that provides network configuration parameters with newly connected hosts. Parameters include IP address, subnet mask, default gateway, DNS server, and leased time. DHCP provides a client/server structure in which 4 DHCP packets exchanged between DHCP server and hosts to assign above parameters automatically.

DHCP plays an important part in the network management. However, DHCP has a number of well known security concerns, in particular:

- DHCP does not have authentication of DHCP messages' origins. On one hand, DHCP clients can not guarantee that they are connected to a trusted DHCP server. On the other hand, DHCP server is not able to ensure that it communicates with a legitimate client.
- Each DHCP message is transmitted in a cleartext [100].

Figure 5 shows classification of DHCP spoofing [47].

Execution of DHCP spoofing-based MITM attack is possible via the rogue DHCP server. The attacker tries to reply to a DHCP Request faster than the legit DHCP server of the local network. Let us consider the following example: network consists of the Victim host, Legitimate DHCP Server, Rogue DHCP Server, and two switches S1, S2. When victim connects to the network next communication occurs (see Figure 6):

1) Client broadcasts DHCP Discovery.
2) Rogue Server sends DHCP Offer (Unicast).
3) Client broadcasts DHCP Request.
4) Rogue Server sends DHCP ACK (Unicast).

As addition, attacker may run DoS attack on the Legitimate DHCP Server to ensure that Victim host will not get answer from it. Another option is to run DHCP starvation attack (when attacker exhausts the allocated pool of IP addresses offered by the valid DHCP server, so that new host machines fail to obtain IP addresses [101]). At this point attacker may provide three misconfigurations for the host [102], where each will lead to the MITM attack:

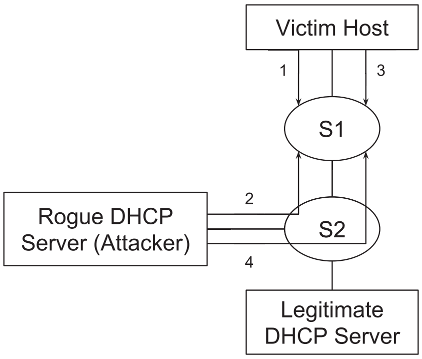


Fig. 6. Rough DHCP server attack.

1) Wrong Default Gateway.
2) Wrong DNS server.
3) Wrong IP Address.

F. DHCP Spoofing Defence Mechanisms

Most recently, Dinu et al. [103] surveyed solutions that aim to make DHCP more secure, and presented their own approach. Based on their work, in Table III we propose our classification of DHCP spoofing prevention methods. All presented solutions are cryptographic and have been categorised according to their classes of cryptography and support of legacy.

There is one more solution, which we consider as a separate hardware category - DHCP snooping. It is a security feature that acts like a firewall between untrusted hosts and trusted DHCP servers [113]. DHCP snooping blocks DHCP responses from ports that do not have DHCP servers associated with them [114]. Moreover, DHCP snooping works together with DAI (see Section III-B4).

G. IP Spoofing-Based MITM Attack

IP is the primary protocol in the Internet, which operates at the network layer of the OSI model. It has the task of delivering packets from the source host to the destination host solely based on the IP addresses in the packet headers. IP defines packet structures that encapsulate the data to be delivered. It also defines addressing methods that used to label the datagram with source and destination information. IP uses a connectionless model, meaning there is no information regarding the transaction state, which is used to route packets in a network. Moreover, IP specifies no method for validating the authenticity of a packet's source. This implies that the attacker could forge the source address to be any it desires.

IP spoofing-based MITM is an attack where a malicious party intercepts a legitimate communication between two non-malicious parties. The malicious entity controls the flow of communication and can eliminate or alter the information sent by one of the original participants, without the knowledge of either of original endpoints. To achieve such results, attackers

TABLE III
COMPARISON OF DHCP SPOOFING PREVENTION APPROACHES

Approach	Category	Concerns/Problems
Komori et al. [105]	S	Legitimate hosts must register in advance, adds additional message flow, hard to manage for large number of hosts.
Duangphasuk et al. [100] (SDSS)	S	Additional communication between the DHCP client and authentication service.
De et al. [106]	S	Additional communication between DHCP server, DHCP client, and authentication server.
Ju et al. [107]	S/RFC	The authors did not describe how the random value (the number, which used by the server and client to compute the session key) is determined.
Hornstein et al. [108]	S/RFC	Additional communication between the DHCP server and Kerberos server.
Wong et al. [109]	A/RFC	Additional flow with trusted server.
Glazer et al. [110]	A/RFC	Was not implemented and authors did not discuss how the digital certificates are transmitted between the DHCP client and server, given the fact that a digital certificate size can easily exceed a DHCP packet length.
Dinu et al. [103]	A/RFC	Increases chances of a DoS attack.
Jerschow et al. [47]	A	Requires unnecessary time to select the appropriate algorithm [48]. (Authors were considering management of large scale Ethernet network.)
Aura et al. [111]	A	The paper does not present the way used to transmit the public key or the digital certificate of the server.
Duangphasuk et al. [100] (SDDC)	A	Approach does not take into account the fact that a digital certificate size could exceed the DHCP message size.
Shue et al. [112]	A	Additional communication between the DHCP client and authentication service.

A—Asymmetric/Public-Key Cryptography, S—Symmetric/Private-Key Cryptography, RFC—Based on DHCP Authentication Standard [104].

can use a number of IP spoofing techniques, which may be classified as follows [115]:

- **Blind and Non-Blind spoofing.** The difference between these two types is that in Non-Blind spoofing the attacker is in the same subnet as a victim, which opens opportunity to sniff on sequence and acknowledgement numbers. Blind spoofing requires an attacker to firstly send requests to a network, and then be able to analyse the transmission sequence. Both methods usually are used for DDOS attacks, but also executed as a data mining phase for IP spoofing-based MITM attack.
- **ICMP spoofing.** IP uses ICMP to send one-way messages to implement various error-reporting, feedback and testing capabilities. ICMP has Redirect messages, which are typically used to notify routers of a better route. These messages may be abused for execution of the MITM attack, since ICMP does not provide authentication mechanisms. The attacker spoofs ICMP Redirect messages to route the victim's traffic through its router, where it can be eavesdropped and modified.
- **TCP Sequence-Number prediction.** TCP is a connection-oriented protocol, which means that before communication begins a connection link should be build. This is achieved using three-way handshake: SYN, SYN-ACK, ACK. TCP uses sequence numbers for data acknowledgement. These numbers help protocol to reduce data loss and determine out-of-order packets in same way ensure reliability. Numbers are generated in pseudo-random way in a manner known to both parties. The idea of Sequence-Number Prediction is to find the algorithm of sequence numbers generation, and then use that knowledge to intercept an existing session (often referred to as *Hijacking an authorised session* attack).

H. IP Spoofing Defence Mechanisms

IPSec and Ingress Filtering have been the main mechanisms against IP spoofing in the last decade. Ingress Filtering filters the inbound packets at the border router of the AS, and generally enabled on the edge devices. It can be deployed using two techniques: Access Control List (ACL) and unicast Reverse Path forwarding (uRPF). ACL contains a set of rules, based on which packets will be filtered [116]. uRPF reversely uses the forwarding table for filtering. This technique will work correctly, if forwarding paths are symmetric, but under route asymmetry uRPF may drop valid packets. Ingress Filtering described in more details in Best Current Practices (BCPs) [117], and [118]. Other similar technique is Egress Filtering, which checks outbound packets.

TABLE IV
IP SPOOFING DEFENCE APPROACHES

Approach	Location	Method
Ingress [118]	R	Filtering-on-path
Egress [117]	R	Filtering-on-path
DPF [122]	R	Filtering-on-path
IDPF [123]	R	Filtering-on-path
SAVE [124]	R	Filtering-on-path
BASE [125]	R	Filtering-on-path/Cryptography
Passport [126]	R	Filtering-on-path/Cryptography
SPM [127]	R	End-to-end auth
HCF [128]	H	End-to-end auth
AIP [129]	R&H	End-to-end auth/Cryptography
Stack Pi [130]	R&H	End-to-end auth

R—Router-Based, H—Host-Based, S&H Server and Host Based.

Yao et al. [119] proposed next classification: filtering-on-path (mechanisms filter packets before they aggregated at victim network), and end-to-end authentication (mechanisms validate the source address at the other end of communication). In [120], [121], researchers divided prevention mechanisms based on the location: host-based, router-based, and based on both hosts and routers. In further sections we stick with the second classification. In Table IV, we extend previous categorisations by introduction of cryptography category for solutions, which use encryption or hash functions.

1) Router-Based Solutions: Distributed Packet Filtering (DPF) [122] is a method that filters packets based on the binding between prefix/flow and interface. It checks whether packets have traveled from their source to destination in an unexpected route, in that case they are dropped. Inter-Domain Packet Filter (IDPF) [123] is an extension of the DPF concept. It builds inter-domain filtering rules based on the valley-free feature of inter-domain routing and the BGP announcement filtering rules.

Source Address Validation Enforcement (SAVE) [124] is a new protocol for networks, which propagates valid network prefixes along the same paths that data packets will follow. Routers, which are on the paths, construct filters by using prefixes and paths information. BGP Anti Spoofing Extension (BASE) [125] is a combination of marking and filtering approach, which is based on BGP Update messages. BASE is similar to SAVE, however instead of treating the incoming interface, as the incoming direction, each BASE router marks packets with a unique key and uses the key as the incoming direction.

Spoofing Prevention Method (SPM) [127] is a mechanism, which inserts AS tag into data packet, and validates the bindings between address prefix and AS tag at the destination AS. For each packet that arrives at the destination, routers verify the key in the packet.

Packet Passport system (Passport) [126] is based on symmetric cryptography and hash algorithms. According to [131] solution can not provide protection against spoofing, if the origin AS does not deploy it.

2) Host-Based Solutions: Hop Count Filtering (HCF) [128] checks the validation of source prefix based on the binding between prefix and hop count value. The method produces significant number of false negatives, since the forged network and the attacker can be located on the same distance from the destination. Moreover, HCF can be bypassed by the attackers, who modify the initial TTL value.

3) Host and Router Based Solutions: Stack Path identifier (Pi) [130] is a reactive scheme, where each router uses IP identification field for marking. Such method will guarantee that the packets travelling along the same path will have same marking. Although, if number of attacks increases, it is more possible that any given Pi mark will receive some attack packets, which will force scheme to drop all valid packets.

Accountable Internet Protocol (AIP) [129] is a cryptographic mechanism, in which each address of the Internet is a combination of the hash values of the public keys of the domain and the host. Network host can validate the correctness of the address by running a challenge-response protocol, where a nonce is sent to the domain or host requesting a digital signature of the nonce with the private key.

IV. SSL/TLS MITM Attack

In this section we give a brief introduction of the SSL/TLS protocol (Section IV-A). Then, we describe SSL/TLS MITM attacks (Section IV-B), and based on our classification, we overview prevention mechanisms against SSL/TLS MITM (Section IV-C).

A. SSL/TLS Protocol

Secure Socket Layer (SSL), as well as its successor, Transport Layer Security (TLS), is an encryption protocol designed to provide secure communication and data transfers over the Internet. Both protocols were created for setting up a secure channel between two communicating parties: a client and a server, or between two clients.

Netscape Communications developed SSL in 1990s. Later, in 1999 IETF TLS working group introduced first version of the TLS, which is based on the SSL 3.0. In fact, the version field of the TLS 1.0 protocol is 3.1 [132], and was considered originally as an update of the SSL. Both SSL and TLS have a lot in common, as a result, in some papers [133]–[135] researchers use SSL and TLS terms interchangeably. SSL/TLS protocols provide the main services as follows [136]:

- authenticate users and servers, to ensure that data is sent to the correct destination;
- encrypt data to prevent data filch during transmission;
- maintain data integrity during transmission to ensure data is not changed.

The SSL/TLS protocols establish session in the way of combining four protocols: Record protocol (ensures that connection is confidential and trustworthy), Handshake protocol (responsible for negotiating session variables between clients through a handshaking process), Change Cipher Spec protocol (instantiates the newly negotiated connection state) and Alert protocol (notifies an entity of an error). The X.509 certificate provides public key in order to prevent attacks on it. Certificates are produced by third parties, named Certificate Authorities (CAs). If the signature of the certificate can be traced back through a certificate validity chain (named certification path) up to a trusted CA, then communicating party assumes that the certificate is valid.

This certificates validation method is the de facto standard in the Internet, and it has been considered "secure" for decades [12]. In the last decade, particular attention has been given to X.509 Public Key Infrastructure (PKI) security. There are hundreds of CAs, where every single one has the ability to issue trusted certificates to any website on the Internet. In case one of trusted CAs is compromised (has issued a replacement certificate that contains a rogue key), it will undermine the whole idea of SSL/TLS security.

Most recently, Holz et al. [137] carried out empirical analysis of X.509 PKI, discussing weak sides, and proposing improvements to it. Georgiev et al. [138] presented an in-depth study of SSL connection authentication in non-browser software, and concluded that SSL certificate validation is completely broken in many critical software applications and libraries. Huang et al. [13] presented the first analysis of forged SSL certificates in the wild, which gave statistics about real usage of forged certificates. During the study it was discovered that 0.2% of real-world connections were substituted with unauthorised forged certificates. Soghoian et al. [139] based on weaknesses of X.509 PKI described compelled certificate creation attack.

B. SSL/TLS MITM Attack

The security guarantees offered by SSL/TLS depend on the validation of the certificate. Consequently, one of the attack's objectives is to hijack, or to forge the certificate. Karapanos et al. [14] proposed the following categorisation of the SSL/TLS MITM attack.

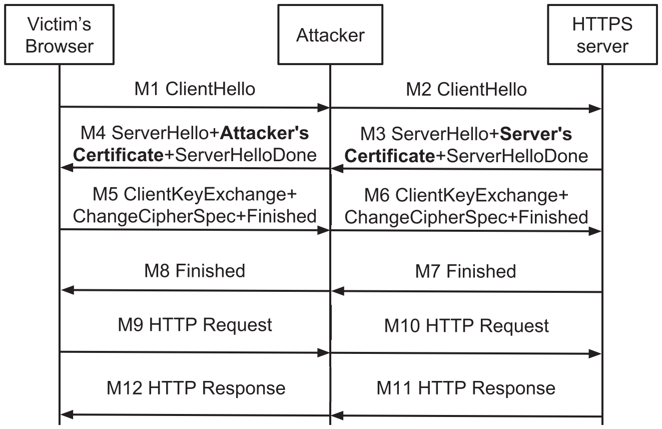

Victim's Browser
Attacker
HTTPS server
M1 ClientHello
M2 ClientHello
M4 ServerHello+**Attacker's Certificate**+ServerHelloDone
M3 ServerHello+**Server's Certificate**+ServerHelloDone
M5 ClientKeyExchange+ ChangeCipherSpec+Finished
M6 ClientKeyExchange+ ChangeCipherSpec+Finished
M8 Finished
M7 Finished
M9 HTTP Request
M10 HTTP Request
M12 HTTP Response
M11 HTTP Response

Fig. 7. Example of SSL/TLS MITM Attack.

- MITM+certificate:
  - (i) Attacker holds a valid certificate to the target web server. This case is possible if the attacker compromises a CA, or is able to force it to issue such certificate.
  - (ii) Attacker holds an invalid certificate. In this scenario the attacker may succeed if the victim will ignore the security warnings, which is a common phenomenon [140].
- MITM+key: attacker has a private key to legitimate server.

While we are not aware of incidents involving server key compromise, let us consider example of SSL/TLS MITM attack with illegitimately generated certificate (see Figure 7). Suppose, network consists of victim's browser, HTTPS server, and attacker (positioning itself in between). First of all, the attacker intercepts SSL/TLS ClientHello message from the browser (message M1), and responds to it using invalid certificate (message M4). If the victim accepts the forged certificate (ignores the security warnings of the browser), the attacker finishes connection setup (messages M5 and M8). In parallel, attacker establishes its own SSL/TLS connection with the server (messages M2, M3, M6, M7). As a result, the attacker has two active SSL/TLS connections: with victim's browser, and with the server. At this point, attacker relays all encrypted messages between them (messages M9, M10, M11, M12). In the middle, attacker decrypts messages from the client, and then re-encrypts them before sending to the server and vice-versa. Therefore, the attacker can access private information of the victim, or even modify it. Cases when attacker has valid certificate or private key are implemented in a similar scheme. In messages M4 and M5 attacker should not use its certificate, but a valid one instead.

C. SSL/TLS Defence Mechanisms

A number of proposed solutions which aim to defend SSL/TLS from MITM attack uses third-party entities that provide various benefits: protection of the first connection to a new domain, scalable attestation of certificates for all public domains, and minimal requirements for web applications. On the other hand, they raise new challenges: deployment and operational costs, more complex trust model and certificate revocation procedures, new privacy risks. Dacosta et al. [141], researchers surveyed both sides of the third-party solutions.

Clark et al. [11] carried out one of the most significant surveys of defence against SSL/TLS MITM attack. Authors reviewed the spectrum of issues concerning trust model between CA and browsers, and focused on the certificate pinning approaches. In Section IV-C2 we provide a brief overview of their work. In [12]–[14], researchers carried out small studies of detecting and defeating mechanisms of SSL/TLS MITM attack. In next sections, based on their work, we propose our own classification and add unmentioned schemes.

1) Detection of Forged Certificates: Solutions from this section are similar to previously discussed detection schemes in other protocols (see sections III-B1, III-D1). Certificate Transparency (CT) [142] creates a central audit log of HTTPS certificates, which is verifiably append-only and maintained by independent monitors. ICSI Certificate Notaries Service [143] passively collects certificates at multiple independent Internet sites, and then aggregates them into a central database almost in real-time. EFF SSL Observatory [144], or Crossbear [145] proposals actively scan the Internet either by querying the TLS-enabled servers or asking users to submit the certificates that they see. All these solutions provide additional data about certificates and help to validate their origin, but they use third-parties.

2) Certificate Pinning Solution: This approach deals with the MITM attack by associating hosts with their expected X.509 certificates or public keys. In such scheme, servers publish certificates and public keys (which will be used for future SSL/TLS handshakes), then users can detect if there were any changes to them. Clark at. al. [11] proposed the following categorisation of the certificate pinning approach:

- **Server-based** methods, where pins are stored on servers. In [12], [146], [147] papers researchers offered various implementation.
- **Client History based** methods, where browser remembers the last browser-acceptable public key encountered for a particular site, and warns the user if this information changes. In practice, based on this approach, Soghoian et al. [139] implemented a Firefox plugin.
- **Preloaded-based** methods, where browser vendors include a list of pins within the browser itself. For example, Google Chrome currently pins a number of certificates for its own domains, as well as others by request [148].
- **DNS-based** methods, where servers pin their public key in their DNSSEC record. In practice, the DNS-based Authentication of Named Entities (DANE) protocol was implemented [149].

In [12], authors pointed out two problems with certificate pinning technique: the demand of the secure associations database maintenance, and complexity of the trust between associations.

3) Multipath Probing Solutions: The SSL/TLS MITM attack is usually executed as a targeted attack, rather than as a global one. This means that the attacker needs to convince only one, or few victims in validity of certificate, while all other users are not effected. As a result, distributed voting approach is effective against SSL/TLS MITM attack. This methods are similar to ARP voting-based algorithms (see Section III-B3). Wendlandt et al. [150] proposed a system named Perspectives, which is distributed over the Internet, and acts as notaries. Each notary maintains a local database of known certificates. Depending on the voting mechanism of notaries certificates are rejected or accepted. Solution was presented as a Firefox plugin. Convergence [151] is a similar approach, but with more general architecture and crowdsourcing functionality. Another schemes are DoubleCheck [152] and The DetecTor Project [153]. Both of them work together with Tor anonymity network. The DetecTor Project makes every client to act as their own notary. It uses Tor network to check the authenticity of servers' certificates, due to the ability of making connections from different network locations. Multi-path probing can detect local certificate substitution attacks, but not attacks where all traffic to the host is modified. Also, notary approaches might produce false positives when servers switch between alternative certificates. Moreover, clients may experience slower SSL/TLS connection times due to querying multiple notaries during certificate validation [13].

4) Force SSL/TLS Connection Solutions: This category of solutions force communicating parties to use the SSL/TLS connection. ISAN-HTTPSEnforcer [154] uses the Javascript API to enforce redirection to HTTPS. When the web server responds to a request from the user's browser it also responds with messages and scripts to indicate redirection to HTTPS. The drawback of the method is that the first connection from the user is made over insecure HTTP. The attacker may strip the call to JavaScript API in the response packet, which leaves a chance to execute the SSL/TLS MITM attack. Another problem is that JavaScript may be disabled in the browser's configurations. Similar solution is HTTP Strict Transport Security (HSTS) [155], a proposal which allows websites to instruct browsers to make SSL/TLS connections mandatory on their sites. The web server attaches a special header in the response packet. This header gives the list of sub-domains and forces user's browser to make connection to them over HTTPS. The method has the same weakness as ISAN-HTTPSEnforcer - initial connection is over HTTP. Sugavanesh et al. introduced SHSHTTPS Enforcer (SHSE) [156], a local daemon, which enforces URL redirection before the request flows over the insecure protocol.

5) Friendly MITM: Conti et al. [157] introduced MITHYS (Mind The Hand You Shake) solution, which tackles the MITM vulnerability of mobile applications by taking on the security checks required to establish a proper secure connection. The main idea behind MITHYS is to act as a friendly MITM on the mobile device. Every time a *new* application (an application which has not been tested yet) requests a resource via the HTTP over the SSL protocol the MITHYS system tries to act as a MITM, forging a fake ad-hoc SSL certificate for the application. If the application is not vulnerable, it will immediately block the communication, otherwise, the communication will proceed normally. In both scenarios, MITHYS is able to protect the application from potentially malicious MITM attacks by performing additional checks on the SSL connection.

6) TLS Extensions: Firstly, Dietz et al. [158] proposed Origin-Bound Certificates approach, and later Balfanz and Hamilton modified it to Channel IDs extension [159]. It is a TLS extension, which strengthens client authentication, but as Karapanos et al. [14] showed it can not fully prevent MITM attacks. In brief, browser stores private/public key pairs, which were created during the TLS handshake with a TLS-enabled web server. Public keys are called Channel IDs, they are used for identification of browser across multiple TLS connections. The approach blocks most of the existing MITM attacks, since attackers can not impersonate the client (without stealing the self-signed private key from the legitimate browser). However, it does not prevent an impersonated server from supplying a cacheable malicious JavaScript file to the client, which later executes in the context of the victim website, and potentially exfiltrates data by reconnecting to the legitimate server [13]. Addressing this problem Karapanos et al. [14] proposed Server Invariance with Strong Client Authentication (SISCA). The main idea behind the SISCA is to ensure that the browser communicates only with one entity, either the legitimate server, or the attacker, but not with both simultaneously.

V. BGP MITM Attack

In this section, we briefly overview the BGP protocol (Section V-A) and the IP hijacking attack. Then, we describe in details the BGP MITM attack (Section V-B). Finally, we list prevention mechanisms against this specific attack (Section V-C).

A. BGP

The Border Gateway Protocol (BGP) is an inter-autonomous systems' routing protocol, whose primary function is to exchange network reachability information with other BGP systems. This information is sufficient to construct a graph of AS connectivity from which routing loops may be pruned, and some policy decisions at the AS level may be enforced [160]. BGP is the core routing protocol for the Internet, it helps to choose fastest paths through the ASs to deliver data. BGP usually selects a route that traverses the least number of ASs (more specific routes addresses over less specific one), named the shortest AS path.

BGP was designed before the Internet environment became perilous, thus the original proposal has little considerations for protection of the information it carries. According to [161], BGP does not provide protection against MITM attacks, because it does not perform peer entity authentication. This serious security vulnerability in BGP was brought to the light in 2008 by Pilosov and Capela [162] on the Defcon conference, where they demonstrated an Internet scale MITM attack. In their presentation, researchers succeed in redirecting legitimate network traffic to an unauthorised AS, recording it, and then sending it to the intended destination. All of this can be done without any approval or authorisation from service providers

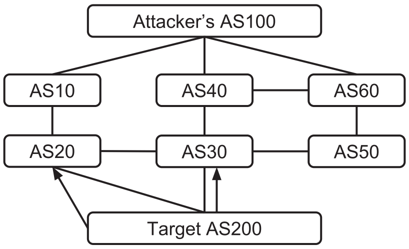


Fig. 8. Target AS200 announce its way to AS20 and AS30.

or end users. Such technique besides MITM attack may result into: generation of DoS traffic, sending spam, network unreachability, or service failure.

Later in 2012, the Federal Communications Commission (FCC) stated that the three top cyber threads are represented by botnets, domain name fraud, and Internet protocol hijacking[1] [163]. FCC called network operators to develop and adopt new technical standards that will secure Internet routing. FCC proposed to establish the secure BGP standards with certified registry that will enable ISPs to validate the authenticity of routing information.

B. BGP MITM Attack

The BGP MITM attack is based on the IP hijacking, also known as BGP hijacking, route hijacking, prefix hijacking. IP hijacking occurs when a compromised or misconfigured BGP speaker announces to its peers that its AS can reach more specific routes (long network prefixes) [164]. As a result, the traffic destined for the specific IP address space will never reach the destination.

Mizuguchi et al. [165] divided IP hijacking into prefix hijacking and AS path falsification. Prefix hijacking involves an attacker falsifying the NLRI field of a BGP Update message [166]. AS path falsification occurs when the attacker changes a path attribute in the Update message [166]. Hepner et al. [167] mentioned two possible variants of the hijacking attack: hijack unused (but maybe assigned) IP space and hijack currently used IP space. Both types of hijacking allow an attacker to attract all traffic bound for the hijacked space.

Based on [162], let us consider an example of a network consisting of 8 ASs, where AS100 is the attacker, and Target AS200 is the destination point of the traffic. Target AS200 originates 10.10.220.0/20 address, and announces its way to AS20 and AS30 (see Figure 8). To Attacker's AS100 this propagation goes through the map (see Figure 9). After converging, the view of forwarding information base (FIB) for 10.10.220.0/22 will look like on Figure 10.

To execute BGP MITM on this network, an attacker should firstly configure routers to advertise availability of network addresses owned by the target (Target AS200). Then it should advertise slightly more specific routes within the global Internet routing table, thus, become destination (prefix hijacking). In our example, Attacker's AS100 will announce longer prefix 10.10.220.0/24, and consequently, this AS will become recipient for any network traffic destined to the original hijacked prefix (10.10.220.0/20).

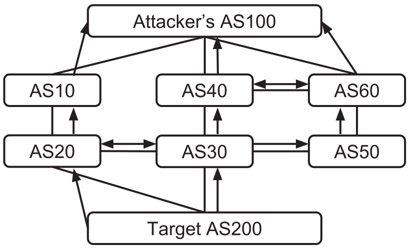


Fig. 9. Propagation map of the network.

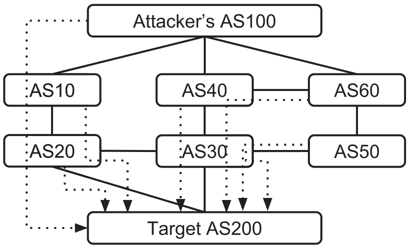


Fig. 10. FIB after converging.

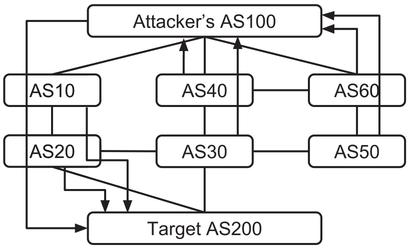


Fig. 11. BGP MITM attack.

Next, attacker needs to deliver network traffic to its appropriate destination. One of the options is to plan a *return path* through the Internet. To do this, the attacker should identify path between its AS and the destination network (from Attacker's AS100 to Target AS200 through AS10 and AS20). Also, this AS path must then be prepended to all BGP advertisements for the hijacked prefixes. As a result, we have the next BGP MITM attack scheme: AS100 has on input data from AS30, AS40, AS50, AS60, which can be manipulated and redirected to Target AS200 (see Figure 11). One more thing, which may be added to the attack is the TTL adjustment. Changing TTL attacker may remove tracks from the hijacked traffic, otherwise, packets will give Target AS200 the detour of the flow.

The drawback of the BGP MITM is a loss of some part of Internet (because of return path), but on the other hand, it still has quite significant size of hijacked traffic.

C. BGP MITM Defence Mechanisms

Subramanian et al. [168] and Huston et al. [169] carried out surveys of the BGP security, focusing on weaknesses and suggesting methods to make protocol more secure. More recently, in [170], [171], researchers performed surveys of IP hijacking and presented different categorisation of them. Shi et al.

[1] BGP MITM is based on IP hijacking.

[171] divided approaches on: cryptographic-based prevention, anomaly mitigation, and anomaly detection. Liux et al. [170] focused on detection methods and divided them based on timing: prevention before the attack, detection during the attack, and reaction after the attack. Based on these works, we decided that from the perspective of the MITM attack the best way to classify is to divide on: detection solutions and proactive mechanisms. In further sections, we survey both types.

1) Detection of BGP MITM: Detection solutions are similar to previously discussed detection methods (see sections III-B1, III-D1, and IV-C1). Detection approaches may be divided into [166]:

- Victim-centric: detects the hijacking of its own IPs.
- Infrastructure-based: relies on a centralised database, or a set of vantage locations distributed over the Internet.
- Peer-centric: detects the hijacking of communication peer IPs, based on the analysis of routing messages.

Shi et al. [171] proposed another categorisation - based on the type of used information:

- Control-plane: an AS announces BGP routes to network prefixes that it does not own.
- Data-plane: continuously probes target networks from multiple vantage points, looking for discrepancies that may arise.
- Combination of both: hybrid system that correlates the results from both perspectives to reduce the number of false positives.
- Correlation: correlates the control-plane route and data-plane reachability on a number of distributed diagnosis nodes in realtime.

Prefix Hijacking Alert System (PHAS) [172] is a proposal that alerts prefix owners when Multiple Origin AS (MOAS) conflicts detected. The schema collects updates about the network from RouteViews and RIPE repositories (centralised databases, that collects real-time information about the global routing system from several different backbones).

Topology [173] is an IP hijacking detection algorithm that finds invalid AS path contained in BGP Update messages that do not exist in AS topology of the Internet. However, attackers can fake valid routing information, and approach has low accuracy, because of the dependence on statistical data.

Zheng et al. [174] proposed approach based on path disagreement and hop count instability during IP hijacking. The solution monitors the network from multiple points, and calculate paths to ASs together with hop counts. Then mechanism analyses similarities before and after the routing update. In case number of inconsistencies cross curtain threshold alarms are raised. A bit later the same authors presented new victim-centric approach named ISPY [175]. The method uses prefix-owner-based probing to monitor network reachability from external transit networks. However, both solutions suffer from inaccuracy of the probing results.

Hu et al. [176] introduced real-time method, which includes both control-plane and data-plane approaches. When monitoring systems sense suspicious routing updates, they collect fingerprints from the prefix owner's ASs, thus, to avoid rogue ASs in the future routing. However, the authors have not produced appropriate fingerprinting method for detection, and have not described how to solve problem with firewalls, which may block data collection.

Hong et al. [166], [177] proposed network reachability-based IP hijacking detection together with fingerprinting methods. The scheme collects host and network fingerprints to develop network reachability tests. Then, through the fingerprint comparison, approach can conclusively detect the IP hijacking occurrence.

Shi et al. [171] pointed out the deep-rooted reason of the control-plane and data-plane methods inefficiency - a lack of close and pervasive correlation between them. Authors presented solution named Argus, which is an extended combination method. The main idea is to calculate the relations between the routes (control-plane) and the reach-abilities (data-plane) in different ASs, which may be regarded as a fingerprint of a route event. Based on this fingerprint, system decides whether a route change is actually caused by the IP hijacking. Besides, Argus system is easy to deploy in comparison to other solutions. Based on [171], we outline pros and cons of hijacking detection methods in Table V.

2) Proactive Prevention: Proactive prevention systems, in contrast, block attacks from propagation in the first place. Pretty Good BGP (PGBGP) [178] is non-cryptographic technique, which can mitigate BGPs most critical vulnerabilities. The main idea behind the protocol is to make process of adoption of new route information for routers more cautious, when origin of AS is unfamiliar. Such mechanism may help to stop spreading the attack, but more preventive techniques are cryptographic, which are listed next.

Kent et al. [179] performed one of the major studies of domain routing security, and proposed first cryptographic solution named secure BGP (sBGP). In this approach, BGP speakers verify the identities of other BGP speakers, AS administrators and address prefix owners by authorisation. To achieve this, sBGP uses digital signatures, X.509 certificates and PKIs (so-called basic security framework) [180]. Also, sBGP requires a PKI for address allocation, where every address assignment is reflected in an issued certificate (so-called verification framework) [180]. The drawback of the mechanism - is the performance capability of the routing systems, which appeared to be too slow.

Later, based on sBGP researchers presented secure origin BGP (soBGP) [181], where they tried to make protocol faster and less heavy for computation. In soBGP, authors reduced the amount of validated material in the BGP Update message, consequently, reduced the processing overhead for Update message validation [169]. However, protocol still has vulnerability to the DoS attack, and demands for changes in the existing BGP infrastructure. Both sBGP and soBGP are not deployed.

Pretty secure BGP (psBGP) [182] is another update to sBGP. Protocol introduces the notion of usage the reputation scheme in place of a hierarchical address PKI (Huston et al. [169] discussed infeasibility of hierarchical PKI for addresses). The solution uses complex inter-AS cross certification with prefix assertion lists, which cast doubts about reliability of the verification and treatment of confidence. Table VI shows comparison between BGP, soBGP, psBGP security design principles.

TABLE V
PROS AND CONS OF IP HIJACKING DETECTION METHODS

Methods	**Control-plane** Kruegel et al. [173], Lad et al. [172]	**Data-plane** Zheng et al. [174], [175]	**Combination** Hu et al. [176], Hong et al. [166]	**Correlation** Shi et al. [171]
Detection delay	Realtime	Minutes	Minutes	Realtime
Attacker Information	Y	N	Y	Y
Accuracy	Low	Medium	High	High
Sub-prefix hijacking	Y	N	Y	Y
Scalability	Good	Poor	Good	Good
Deployment	Easy	Easy	Hard	Easy

TABLE VI
COMPARISON OF sBGP, soBGP, psBGP SECURITY DESIGN PRINCIPLES [202]

BGP security design principles	**sBGP** [179]	**soBGP** [181]	**psBGP** [182]
BGP peer authentication	Y	N	N
BGP Update message integrity	Y	N	Y
IP prefix origin authentication	Y	Y	Y
Route advertisement authorisation	Y	Y	N
AS-Path Validation	Y	Y	Y
AS-Policy compliance check	Y	Y	N

Another cryptographic solution is Resource Publication Infrastructure (RPKI) [183], but the proposed method for BGP origin validation is not intended to secure the routing system against attackers, who are faking their origin AS. Lepinski et al. proposed a PKI-based mechanism named BGP Security (BGPSEC) [184], which is still under development. However, as recent paper [185] showed, the attacker can use *route leak* vulnerability and still succeed in BGP MITM execution.

VI. FBS-BASED MITM

In this section we give a brief description of the FBS attack and its connection to the MITM attack. Then we overview GSM and UMTS networks (Sections VI-A and VI-D), and we introduce the FBS spoofing-based MITM (sections VI-B and VI-E). Finally, we list prevention mechanisms against the FBS spoofing-based MITM (sections VI-C and VI-F).

False Base Station (FBS) attack (also known as Fake Base Station) is an attack, when malicious third party masquerades its Base Transceiver Station (BTS) as a real network's BTS [186]. A fake BTS system is an equipment that can act as a real BTS, it broadcasts BTS signal over the air and makes mobile phones in covered area to communicate with it [187]. FBS attack may include the real-time jamming system [188], which blocks all active carries in the area. Using jamming system, attacker can drown legal BTSs and force victim to connect to fake one.

Systems with one-way authentication, where connected nodes can not validate authenticity of the serving network show weakness to the FBS attack. For instance, GSM and GSM/UMTS combined networks are vulnerable to this attack.

FBS-based MITM is an attack where third party forces victim to connect to the fake BTS and then by using this station the attacker manipulates victims traffic. To execute such MITM, attacker may use few fake BTSs with different protocols.

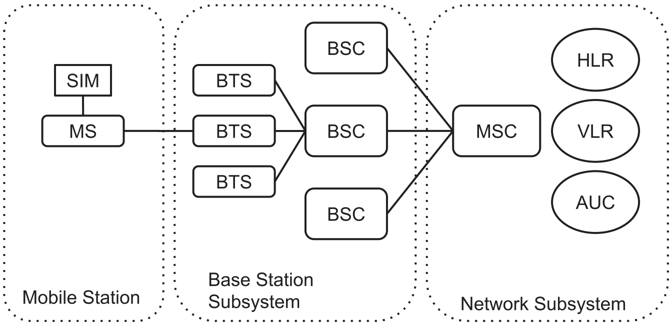


Fig. 12. GSM architecture.

A. GSM

In early 90s European Telecommunications Standards Institute introduced GSM as a second generation (2G) telecommunication standard. Today, according to the mobility report [10], GSM covers more than 90% of the world population. There are two basic types of services offered through GSM: telephony and data bearer. The GSM experienced gradual improvements that leaded to several versions such as GSM1800, HSCSD, EDGE, GPRS, and consequently, to the third generation of cellular networks (see Section VI-D).

The GSM architecture (see Figure 12) consists of Mobile Stations (MSs) and BTS, which communicate with each other through radio links. Each BTS connects to the Base Station Controller (BSC). BSC links to the Mobile Switching Center (MSC), which is responsible for routing signals to and from fixed networks [189]. Home Location Register (HLR) and the Visitor Location Register (VLR) are the two major databases for each mobile service provider in the GSM architecture. HLR is responsible for maintaining the subscriber information and their current location. VLR is responsible for keeping the visiting users information [190]. MSC acts as a bridge between wireless and wired networks. Each of GSM subscribers has the secret key, which is stored in Subscriber Identity Module (SIM) card of the MS. The Authentication Center (AUC) has other secret key, which is shared with the subscriber and AUC. AUC generates a set of security parameters for execution of encryption and authentication.

GSM was built with security considerations in mind, to provide the privacy and integrity of the cellular network [191]. In comparison to the first generation, it solved number of attacks, for instance, phone cloning. GSM security requirements include well known security services like anonymity, authentication, privacy, user data, and signalling protection. Many papers (for instance, [192]–[200]) have discussed key security issues in GSM. Among them, we can mention the following.

- **Confidentiality.** Communications between the MS and BTS must be encrypted by using long shared authentication key, thus, not be divulged to third parties. However, Paik et al. [201] showed that French carrier SFR sends SMS in cleartext, also, all Indian carriers send GSM voice and data traffic without any encryption at all.
- **Subscriber identity authentication.** Good identity authentication system can guarantee that no unauthorised user gets services from the GSM system. Although, GSM only authenticates the MS to the network, and not vice versa, which leads to the FBS, MITM, DoS attacks.
- **Weak cryptographic algorithms.** Barkan et al. [193] reported that some GSM encryption algorithms are easy to break. Researchers describe a cipher-text-only attack on A5/2 that requires a few dozen milliseconds of encrypted off-the-air cellular conversation and finds the correct key on a personal computer in less than a second. Also, authors discussed weak spots in other ciphers: A5/1 and A5/3. Later, Toorani et al. [198] showed that A3 and A8 algorithms, which are part of GSM standard, are also crackable.

GSM has a lack in authentication procedure, used mechanisms are unidirectional, and users can not verify the serving network. Also, user authentication information and cipher-suits may be reused. Consequently, GSM network is vulnerable to the FBS-based MITM attack, as we describe in details in the next section.

B. MITM Attack on GSM

The main idea behind the attack is to impersonate same mobile network code as the legitimate GSM network to false BTS (or IMSI cather [203]), and convince victim that this station is the valid one. Let us consider the next example (see Figure 13): network consists of the Legitimate MS, Legitimate BTS, False BTS, and False MS. Attacker's network is a combination of the False BTS and False MS. While in *stand-by* mode the MS connects to the best received BTS. Therefore, False BTS should be more powerful than the original one, or closer to the target. If the victim is already connected, then the attacker requires to drawn any present real stations.

The algorithm of the FBS-based MITM attack on GSM is the following (see Figure 14):

1) Attacker sets-up connection between False BTS and Legitimate MS.
2) False MS impersonates the victim's MS to the real network by resending the identity information, which was received from the step 1.
3) Victim's MS sends its authentication information and cipher-suites to the False BTS.
4) Attacker forwards message from step 3 to the Legitimate BTS, with changed authentication abilities of the MS to *do not support encryption* (A5/0 algorithm [204]), or to weak encryption algorithm (e.g., A5/2).

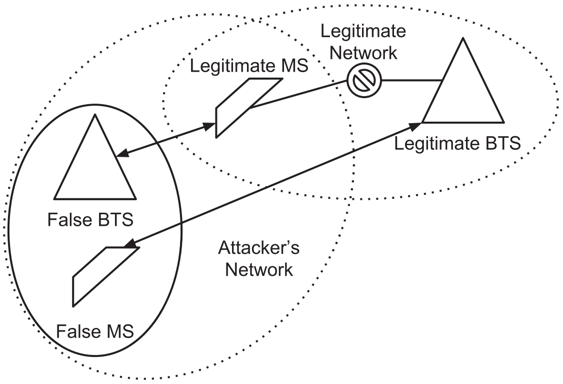


Fig. 13. FBS-based MITM attack on GSM: direct link between Legitimate MS and Legitimate BTS dropped, instead connection goes through the False BTS and False MS.

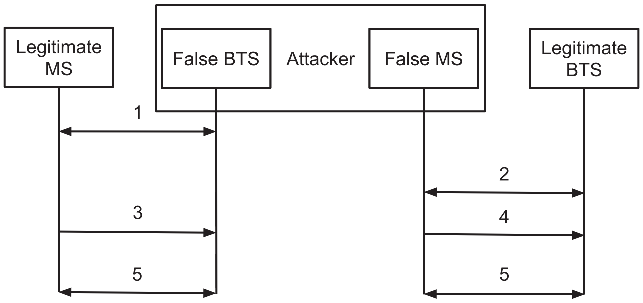


Fig. 14. Algorithm of the FBS-based MITM attack on GSM network.

5) Legitimate MS and Legitimate BTS exchange authentication challenge (RAND), and authentication response (SRES), attacker forwards them.

Finally, the authentication is finished. All following messages between victim and real network are going through attacker's entities, with encryption specified by attacker, or no encryption at all. This manipulation is possible, since GSM does not provide the data integrity [205], as a result, attacker can catch, modify, and resend messages.

At the designing phase of the GSM protocol, FBS seemed impractical due to costly required equipments, but currently this kind of attack is completely applicable since costs decreased [198]. Paik at. al. [201] besides describing GSM security concerns, pointed out that nowadays attackers are better equipped. Among the reasons we can identify open-source projects (e.g., OpenBTS [206]) and low-cost hardware (e.g., Ettus Research [207]). In particular, attacker can build its own false BTS for less than $1000.

C. GSM MITM Defence Mechanisms

To prevent MITM on GSM, researchers have been working in two main directions: better authentication protocols and better cipher algorithms. In the following, we will make overview of the first direction, while second is fully cryptographic.

Hwang et al. [208] proposed an approach which makes authentication of the VLR by MS using certificates generated by the HLR and a temporary secret key. However, mutual authentication takes place only for the first communication.

Chang et al. [195] improved such solution, in a way that mutual authentication is carried out for every communication. However, protocol is weak against the brute force [209] and MITM [210] attacks. Based on [208], Kumar et al. [190] proposed a method which makes MS and VLR authenticate each other, and establishes a session key. Unfortunately, the approach requires changes to the architecture and does not work with roaming users.

Later, Fanian et al. [211] proposed a solution based on the symmetric polynomials. In their approach, each MS and VLR stores a share of the symmetric polynomial, which is later used for authentication. Most recently, same authors proposed a novel mutual entity authentication using the TESLA protocol [210]. The proposed solution not only provides secure bilateral authentication, but also decreases the call setup time and the required connection bandwidth.

Lee et al. [212] described GSM mutual authentication protocol, which is MITM resistant. Researchers added identification of the legability to VLR, and authorisation for VLR to authenticate the MS without knowing the secret key (Ki) of the MS. Also, VLR keeps only a copy of temporary secret key (TKi) to identify the MS. To verify the legality of the visited VLR, the HLR sends to the VLR a certificate $CERT_{VLR}$ that will be authenticated by the MS. Also, the VLR can verify the MS by the received signed result (SRES). Both VLR and MS are protected, unless Ki or TKi is known. Therefore, no one can pretend to be the legal VLR to fool the MS and get the MSs credentials.

A Secure AKA (SAKA) protocol similarly to previous approaches, has Ki and VLRs's certificates (VLR_C), but researchers have added additional checks. Firstly, before issuing VLR_C to VLR, HLR verifies if MS is valid based on Ki. Secondly, VLR maintains a counter *count*, which helps MS to validate VLRs (counter does not require synchronisation). Using these measurements SAKA is able to prevent replay, redirection, impersonation, and MITM attacks.

D. UMTS

In early 2000s the 3rd Generation Partnership Project (3GPP) developed UMTS, which is a part of the third generation (3G) telecommunication standards. According to [213], in 2011 more than 45% of population is covered by 3G, and in 2017 this number will reach 85%. The process of increasing UMTS resulted in existence of one (GSM) or both (GSM and UMTS) base stations in some areas.

UMTS is a direct successor of the GSM standard, but it provides significant enhancements to overcome vulnerabilities. While GSM supports only subscriber authentication and encryption of the radio interface between the MS and BTS, UMTS provides additional features: authentication token and integrity protection of the signalling traffic [214]. Also, UMTS has stronger cryptographic primitives and bigger cipher key sizes (128 bits) [215]. Besides, UMTS networks have much higher speeds for communication.

From the beginning UMTS authentication and key agreement (AKA) procedure was designed to resist MITM attacks [216]. Meyer et al. [214] stated that the mutual authentication and the data integrity of signalling messages in radio interface are the keys for protection against MITM attack. The mutual authentication uses an authentication quintet, which helps to ensure that a bill is issued to the correct person.

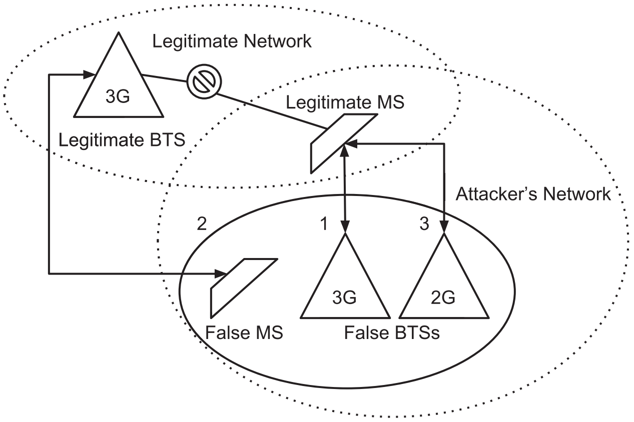


Fig. 15. MITM attack on combined GSM/UMTS networks: 1) IMSI catching, 2) obtaining valid AUTN, 3) GSM impersonation.

The support of GSM networks in UMTS leads to vulnerabilities. Integration of two standards allows them to interconnect, but this in fact, is roaming between two systems that are not equally well protected against malicious attacks. As a result, some of the security issues in GSM may be exploited in the UMTS networks to execute MITM attack. In the next section we will describe in details the FBS-based MITM attack on combined GSM/UMTS network.

E. MITM on Combined GSM/UTMS Network

MITM on combined GSM/UTMS network, as well as MITM on GSM, is based on the FBS attack. FBS-based MITM on combined GSM/UTMS network is directed on UMTS subscriber, and may be executed only while victim uses GSM [217]. In [215], [218], researchers introduced a similar method, which exploits same weaknesses of GSM legacy in UMTS network, but for real-time eavesdropping, without ability to transform data.

UMTS ensures origin and freshness of authentication due to means of AUTN. AUTN contains Message Authentication Code (MAC) and Sequence Number (SQN). A correct MAC indicates that the legitimate network originally generated the authentication token, and SQN shows how old is the AUTN. Both values are checked on MS side, but the mechanism of AUTN may be compromised if the subscriber is in GSM network.

To execute MITM on combined GSM/UTMS attacker has two difficulties: stealing AUTN and ensuring that no encryption is used after the authentication. The first problem may be solved by impersonating victim's MS to false MS. The second problem can be solved by sending false information about the encryption capabilities of victims's MS, which will lead to the *no encryption* mode (in further example we will suppose that victims's MS supports such mode).

Let us consider the following example: a network consists of the Legitimate 3G BTS, Legitimate MS, False 3G BTS, False

TABLE VII
FBS-BASED MITM PREVENTION APPROACHES VERSUS DIFFERENT ATTACKS

Prevent from	Cocktail-AKA [219]	S-AKA [221]	Hwang et al. [222]	NS-AKA [223]	Secure-AKA [15]
MITM attack	Y	Y	Y	Y	Y
Replay attack	Y	Y	Y	Y	Y
Active attack in corrupt network	Y	Y	Y	Y	Y
Redirection attack	Y	Y	Y	Y	Y
DoS attack	N	Partially	Y	N	Y

2G BTS, and False MS. Attacker's network is a combination of the False 3G BTS, False 2G BTS, and False MS. As well as in MITM attack on GSM (see Section VI-B) Legitimate MS should not be connected to local real station. The algorithm of MITM attack on combined GSM/UMTS is the following (see Figure 15):

1) Attacker launches 3G False BTS and catches International Mobile Subscriber Identity (IMSI) of the victim's MS.
2) Attacker uses victim's IMSI to impersonate victim's MS to False MS. Then it fetches AUTN from real network (from HLR using connection with Legitimate 3G BTS), and after receiving data it breaks the connection. At this point attacker has AUTN value, which will prove for the Legitimate MS that False 2G BTS is valid network.
3) Attacker starts 2G False BTS and forces victim to connect. After victim's MS successfully verifies authentication token, attacker sends message of preferred encryption, or no encryption (since GSM does not provide the data integrity [205]). In the end, to finish MITM, attacker should provide access to real network using connection between False MS and legitimate network.

Last step of the method has obvious drawback - it leaves traces. Using False MS for redirection of intercepted traffic will uncover Universal Subscriber Identity Module (USIM). Also, attacker should take into consideration that AUTN contains SQN, and it is crucial that victim has not made authentication between obtaining AUTN and GSM impersonation. Otherwise, token might be out of the range, and will be rejected by Legitimate MS.

F. GSM/UMTS MITM Defence Mechanisms

Saxena et al. [15] had collected proposals, which prevent MITM on GSM/UMTS networks. In the following we overview their work and add to that unmentioned approaches. In Table VII, we show FBS-based MITM prevention approaches versus different attacks. All these solutions as well as GSM proposals are cryptographic (see Section VI-F).

Cocktail-AKA [219] protocol is based on Authentication Vectors (AVs), in comparison to UMTS-AKA it decreases computational and communicational overheads. Cocktail-AKA uses two kinds of AVs: Medicated AV (MAV) and Prescription AV (PAV). MAV calculated in advance by service network (SN), PAV calculated by Home Environment (HE). Once the authentication stage is initiated, the SN distributes MAV, HE distributes PAV, and they produce an AV for mutual authentication with the MS. Saxena et. al [15] included Cocktail-AKA protocol to category *Prevent MITM attack*, but Wu et al. [220] stated that it is vulnerable to impersonation attacks, which lead to MITM.

S-AKA [221] is a new authentication and key agreement mechanism, which is protected against redirection, MITM and DoS attacks. In protocol, SN must continually generate random numbers to challenge MS to reply corresponding responses for every authentication. S-AKA reduces bandwidth consumption up to 38%, but increases the number of messages required in authenticating mobile subscribers.

Hwang et al. [222] proposed another AKA, which is MITM resistant. Researchers introduced additional keys (K or $K_{uv_{perm}}$), which are negotiated between the serving GPRS support node (SGSN) and MS (during the initial authentication between victim, VLR, and HLR). Thus, step 3 in MITM on GSM/UMTS algorithm (Section VI-E) will not work. With K or $K_{uv_{perm}}$, the SGSN has to encrypt the payload prior to transmission, even if *none encryption* command is specified by any GSM BTS. Hence, data confidentiality between MS and SGSN can be assured.

NS-AKA [223] prevents MITM due to usage of additional EK key between MS and VLR. Also, the proposed solution is free from redirection and impersonation attacks. Saxena et al. analysed the message exchanged ratio during the authentication process and discovered that NS-AKA reduces 60% of the messages exchanged ratio in comparison to UMTS-AKA protocol. Although, method does not provide resistance against DoS attack. Later, based on NS-AKA, authors proposed another protocol Secure-AKA [15], where they solved DoS problem.

VII. CONCLUSIONS

In this paper, we have analysed MITM attack and presented a comprehensive classification of such attack based on impersonation techniques. Also, we provided various MITM defence mechanisms along with their descriptions. In Table VIII, we bring together all MITM prevention mechanisms, according to used approaches and context (abstract layer) of applicability. To sum it up, we can collect the most effective methods in the following list. These methods have been discussed throughout the whole paper, so here we refer only to the section in which the method has been presented more in detail.

- Use strong mutual authentication to always fully authenticate endpoints of any communications channel (Section VI-C).
- Exchange public keys using a secure channel (Section IV-C).

TABLE VIII
MITM PREVENTION MECHANISMS

		Approaches				
		Detection	Cryptographic	Voting	Hardware	Other
OSI layers	Application	*BGP:* [172], [173], [174], [175], [176], [177], [166], [171]. *DNS:* [80], [75], [81], [82], [83], [21].	*BGP:* [179], [181], [182], [183], [184]. *DNS:* [90], [91], [93], [97], [83], [83]. *DHCP:* [105], [100], [106], [107], [108], [109], [110], [103], [47], [111], [100], [112].	*DNS:* [81], [82], [83], [21].	*DHCP:* [113].	*BGP*: [178]. *DNS:* [85], [86], [87], [88], [37].
	Presentation	*SSL/TLS:* [142], [143], [144], [145].	*SSL/TLS:* [149], [158], [159], [14].	*SSL/TLS:* [150], [151], [152], [153].		*SSL/TLS:* [11], [146], [147], [12], [139], [148], [154], [155], [156], [157].
	Transport		*IP:* [125], [126], [129].			*IP:* [118], [117], [122], [127], [128], [130], [123], [124].
	Network					
	Data Link	*ARP:* [19], [58], [60], [61], [62], [18], [19], [64], [36], [65], [66].	*ARP:* [45], [46], [20], [67], [24], [47], [49], [43].	*ARP:* [35], [39], [40].	*ARP:* [62], [56], [25], [57].	*ARP:* [68], [50], [51], [52], [23], [32], [53], [54].
Cellular networks	GSM		[208], [195], [190], [211], [210], [212], [224].			
	UMTS		[219], [221], [222], [223], [15].			

- Use few secure channels for verifying if data has not been compromised (Section IV).
- Sign public keys by a certified authority (Section IV-A).
- Use certificate pinning (Section IV-C).
- Encrypt your communication using cryptography (Section III).
- In general, examine the expected behaviour of interacting endpoints, according to the agreed interaction protocol.

The paper did not aim at providing an extensive analysis of future research directions. Indeed, this would require a detailed analysis of future directions in communication technologies as well as a technical analysis of new (but not yet mature) technologies (like Li-Fi, to mention only one). This was clearly out of the scope of the paper. However, on the other hand we can conclude with some thoughts that we think they could represent next steps in MITM research, at least accordingly to our analysis.

- First of all, we covered MITM attack not in all communication channels. During literature surveying we found a significant number of papers concerning MITM attacks in next communication channels: LTE, WiMax, WLAN, VoIP, RFID, Bluetooth, NFC, and combinations of them; the same for next protocols: SIP, HB-like. Research on these new technologies is still ongoing and the adoption of these new technologies still a question mark. Consequently, a full MITM taxonomy, covering also these technologies, would constitute the natural next step in MITM research, both to study the security of these technologies and to provide solutions, in order to help in spreading their adoption.
- A number of papers were covering MITM attacks against cryptography fields (e.g., quantum, elliptic curve), key exchange/distribution, and new authentication methods. There is a point for making comparison analyses of such solutions, and possibility of their implementation in protocols.
- All of previously discussed MITM have same traffic flow - traffic goes through middle man, but in some papers (e.g., [225]), authors discuss different models. Thus, another research direction could focus on these *novel MITM schemes*.

REFERENCES

[1] G. Nath Nayak and S. G. Samaddar, "Different flavours of man-in-the-middle attack, consequences and feasible solutions," in *Proc. 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT)*, 2010, vol. 5, pp. 491–495.

[2] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, 1992, pp. 72–84.

[3] R. Demillo and M. Merritt, "Protocols for data security," *Computer*, vol. 2, no. 16, pp. 39–51, 1983.

[4] W. Baker *et al.*, "Data breach investigations report," *Methodology*, vol. 36, pp. 1–63, 2011.

[5] CAPEC. (2014). *Capec-94: Man in the Middle Attack* [Online]. Available: http://capec.mitre.org/data/definitions/94.html

[6] S. Frankel, B. Eydt, L. Owens, and K. Scarfone, "Establishing wireless robust security networks: A guide to IEEE 802.11i," National Institute of Standards and Technology, Gaithersburg, Maryland, USA, Report no. NIST SP 800-97, 2007.

[7] R. Wagner, "Address resolution protocol spoofing and man-in-the-middle attacks," The SANS Institute, Bethesda, Maryland, USA, 2001. Available: https://www.ida.liu.se/~TDDC03/literature/dnscache.pdf

[8] K. M. Haataja and K. Hypponen, "Man-in-the-middle attacks on Bluetooth: A comparative analysis, a novel attack, and countermeasures," in *Proc. 3rd Int. Symp. Commun. Control Signal (ISCCSP)*, 2008, pp. 1096–1102.

[9] A. Ornaghi and M. Valleri, "Man in the middle attacks," in *Proc. Blackhat Conf. Eur.*, 2003 [Online]. Available: http://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-valleri.pdf

[10] I. Ericsson. (2014). *Ericsson Mobility Report* [Online]. Available: http://www.gsma.com/network2020/wp-content/uploads/2014/06/ericsson-mobility-report-june-2014.pdf

[11] J. Clark and P. C. van Oorschot, "SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2013, pp. 511–525.

[12] N. C. COE. (2014). *Detecting and Defeating Advanced Man-in-the-Middle Attacks Against TLS* [Online]. Available: https://ccdcoe.org/cycon/2014/proceedings/d2r2s2_delahoz.pdf

[13] L. S. Huang, A. Rice, E. Ellingsen, and C. Jackson, "Analyzing forged SSL certificates in the wild," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2014, pp. 83–97.

[14] N. Karapanos and S. Capkun, "On the effective prevention of TLS man-in-the-middle attacks in web applications," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 150, 2014.

[15] N. Saxena and N. S. Chaudhari, "Secure-AKA: An efficient AKA protocol for UMTS networks," *Wireless Pers. Commun.*, vol. 78, no. 2, pp. 1345–1373, 2014.

[16] S. Prowell, R. Kraus, and M. Borkin, *Seven Deadliest Network Attacks*. Amsterdam, The Netherlands: Elsevier, 2010.

[17] I. Green. (2005). *DNS Spoofing by the Man in the Middle* [Online]. Available: http://www.sans.org/rr/whitepapers/dns/1567.php

[18] V. Ramachandran and S. Nandi, "Detecting ARP spoofing: An active technique," in *Information Systems Security*. New York, NY, USA: Springer, 2005, pp. 239–250.

[19] M. Carnut and J. Gondim, "ARP spoofing detection on switched Ethernet networks: A feasibility study," in *Proc. 5th Symp. Seguranca Inf.*, 2003 [Online]. Available: ftp://www.linorg.cirp.usp.br/pub1/SSI/2003/A25.pdf

[20] V. Goyal and R. Tripathy, "An efficient solution to the ARP cache poisoning problem," in *Information Security and Privacy*. New York, NY, USA: Springer, 2005, pp. 40–51.

[21] L. Yuan, K. Kant, P. Mohapatra, and C.-N. Chuah, "DoX: A peer-to-peer antidote for DNS cache poisoning attacks," in *Proc. IEEE Int. Conf. Commun. (ICC'06)*, 2006, vol. 5, pp. 2345–2350.

[22] C. L. Abad and R. I. Bonilla, "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks," in *Proc. 27th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW'07)*, 2007, p. 60.

[23] Z. Trabelsi and W. El-Hajj, "Preventing ARP attacks using a fuzzy-based stateful ARP cache," in *Proc. IEEE Int. Conf. Commun. (ICC'07)*, 2007, pp. 1355–1360.

[24] W. Lootah, W. Enck, and P. McDaniel, "TARP: Ticket-based address resolution protocol," *Comput. Netw.*, vol. 51, no. 15, pp. 4322–4337, 2007.

[25] A. P. Ortega, X. E. Marcos, L. D. Chiang, and C. L. Abad, "Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt," in *Proc. Int. Conf. Latin Amer. Netw. Oper. Manage. Symp. (LANOMS)*, 2009, pp. 1–9.

[26] N. Hubballi *et al.*, "An active intrusion detection system for LAN specific attacks," in *Advances in Computer Science and Information Technology*. New York, NY, USA: Springer, 2010, pp. 129–142.

[27] W.-L. Chen and Q. Wu, "A proof of MITM vulnerability in public WLANs guarded by captive portal," *Asia-Pac. Adv. Net.*, vol. 30, pp. 66–70, 2010.

[28] M.-H. Chiu, K.-P. Yang, R. Meyer, and T. Kidder, "Analysis of a man-in-the-middle experiment with Wireshark," in *Proc. Int. Conf. Secur. Manage. (SAM'11)*, 2011, pp. 461–464.

[29] Y. Xi, C. Xiaochen, and X. Fangqin, "Recovering and protecting against DNS cache poisoning attacks," in *Proc. Int. Conf. Inf. Technol. Comput. Eng. Manage. Sci. (ICM)*, 2011, vol. 2, pp. 120–123.

[30] N. Thuc, N. Phu, T. Bao, and V. Hai. *A Software Solution for Defending Against Man-in-the-Middle Attacks on WLAN* [Online]. Available: http://goo.gl/FgGUbT, accessed on Sep. 15, 2015.

[31] S. Alshomrani and S. Qamar, "Investigation of DHCP packets using Wireshark," *Int. J. Comput. Appl.*, vol. 63, pp. 1–9, 2013.

[32] R. Philip, "Securing wireless networks from ARP cache poisoning," M.S. thesis, Comput. Sci., San Jose State Univ., San Jose, CA, USA, 2007.

[33] L. Wu, T. Yu, D. Wu, and J. Cheng, "The research and implementation of ARP monitoring and protection," in *Proc. Int. Conf. Internet Technol. Appl. (iTAP)*, 2011, pp. 1–4.

[34] G. Hao and G. Tao, "Principle of and protection of man-in-the-middle attack based on ARP spoofing," *J. Inf. Process. Syst.*, vol. 5, no. 3, pp. 131–134, 2009.

[35] S. Y. Nam, D. Kim, and J. Kim, "Enhanced ARP: Preventing ARP poisoning-based man-in-the-middle attacks," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 187–189, Feb. 2010.

[36] K. Kalajdzic and A. Patel, "Active detection and prevention of sophisticated ARP-poisoning man-in-the-middle attacks on switched Ethernet LANs," in *Proc. 6th Int. Workshop Digit. Forensics Incident Anal. (WDFIA'11)*, 2011, p. 81.

[37] X. Bai, L. Hu, Z. Song, F. Chen, and K. Zhao, "Defense against DNS man-in-the-middle spoofing," in *Web Information Systems and Mining*. New York, NY, USA: Springer, 2011, pp. 312–319.

[38] Y. Yang *et al.*, "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems," in *Proc. Int. Conf. Sustain. Power Gener. Supply*, 2012, vol. 7, pp. 1–8.

[39] S. Y. Nam, S. Jurayev, S.-S. Kim, K. Choi, and G. S. Choi, "Mitigating ARP poisoning-based man-in-the-middle attacks in wired or wireless LAN," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 1–17, 2012.

[40] S. Y. Nam, S. Djuraev, and M. Park, "Collaborative approach to mitigating ARP poisoning-based man-in-the-middle attacks," *Comput. Netw.*, vol. 57, no. 18, pp. 3866–3884, 2013.

[41] F. Fayyaz and H. Rasheed, "Using JPCAP to prevent man-in-the-middle attacks in a local area network environment," *IEEE Potentials*, vol. 31, no. 4, pp. 35–37, Jul./Aug. 2012.

[42] S. Shukla and I. Yadav, "An innovative method for detection and prevention against ARP spoofing in MANET," *Int. J. Comput. Sci. Inf. Technol. Secur.*, vol. 5, 2015 [Online]. Available: http://ijcsits.org/papers/vol5no12015/5vol5no1.pdf

[43] M. Oh, Y.-G. Kim, S. Hong, and S. Cha, "ASA: Agent-based secure ARP cache management," *IET Commun.*, vol. 6, no. 7, pp. 685–693, May 2012.

[44] R. Kumar, S. Verma, and G. S. Tomar, "Thwarting address resolution protocol poisoning using man in the middle attack in WLAN," *Int. J. Rel. Inf. Assur.*, vol. 1, no. 1, pp. 8–19, 2013.

[45] D. Bruschi, A. Ornaghi, and E. Rosti, "S-ARP: A secure address resolution protocol," in *Proc. 19th Annu. Comput. Secur. Appl. Conf. (ACSAC'03)*, Dec. 2003, pp. 66–74.

[46] M. G. Gouda and C.-T. Huang, "A secure address resolution protocol," *Comput. Netw.*, vol. 41, no. 1, pp. 57–71, 2003.

[47] Y. I. Jerschow, C. Lochert, B. Scheuermann, and M. Mauve, "CLL: A cryptographic link layer for local area networks," in *Security and Cryptography for Networks*. New York, NY, USA: Springer, 2008, pp. 21–38.

[48] K. F. Wahid, "Rethinking the link security approach to manage large scale Ethernet network," in *Proc. 17th IEEE Workshop Local Metropolitan Area Netw. (LANMAN)*, 2010, pp. 1–6.

[49] P. Limmaneewichid and W. Lilakiatsakun, "P-ARP: A novel enhanced authentication scheme for securing ARP," in *Proc. Int. Conf. Comput. Commun. Manage. (ICCCM'11)*, 2011, pp. 83–87.

[50] T. Demuth and A. Leitner, "ARP spoofing and poisoning: Traffic tricks," *Linux Mag.*, vol. 56, pp. 26–31, 2005.

[51] K. Kwon, S. Ahn, and J. W. Chung, "Network security management using ARP spoofing," in *Computational Science and Its Applications—ICCSA*. New York, NY, USA: Springer, 2004, pp. 142–149.

[52] D. Pansa and T. Chomsiri, "Architecture and protocols for secure LAN by using a software-level certificate and cancellation of ARP protocol," in *Proc. 3rd Int. Conf. Convergence Hybrid Inf. Technol. (ICCIT'08)*, 2008, vol. 2, pp. 21–26.

[53] H. R. V. Lab. (2002). *Anticap Naive ARP Poisoning Mitigation* [Online]. Available: https://antifork.org/git/anticap/

[54] W. Xing, Y. Zhao, and T. Li, "Research on the defense against ARP spoofing attacks based on Winpcap," in *Proc. 2nd Int. Workshop Educ. Technol. Comput. Sci. (ETCS)*, 2010, vol. 1, pp. 762–765.

[55] S. Bhirud and V. Katkar, "Light weight approach for IP-ARP spoofing detection and prevention," in *Proc. 2nd Asian Himalayas Int. Conf. Internet (AH-ICI)*, 2011, pp. 1–5.

[56] C. Inc. *Catalyst 6500 Release 12.2sx Software Configuration Guide* [Online]. Available: http://goo.gl/KJEjYv, accessed on Sep. 15, 2015.

[57] Y. Bhaiji. *Security Features on Switches* [Online]. Available: http://www.ciscopress.com/articles/article.asp?p=1181682

[58] I. I. S. GmbH. *ARP-Guard* [Online]. Available: https://www.arp-guard.com/en/info/product/arp-guard_en.html

[59] I. ARPDefender. (2010). *What is Arpdefender Designed For?* [Online]. Available: http://www.arpdefender.com/home

[60] LBNL's Network Research Group. *Arpwatch, the Ethernet Monitor Program; For Keeping Track of Ethernet/IP Address Pairings* [Online]. Available: http://ee.lbl.gov

[61] X. Hou, Z. Jiang, and X. Tian, "The detection and prevention for ARP spoofing based on snort," in *Proc. Int. Conf. Comput. Appl. Syst. Model. (ICCASM)*, 2010, vol. 5, pp. V5–137.

[62] J. Belenguer and C. T. Calafate, "A low-cost embedded IDS to monitor and prevent man-in-the-middle attacks on wired LAN environments," in *Proc. Int. Conf. SecureWare Emerging Secur. Inf. Syst. Technol.*, 2007, pp. 122–127.

[63] T. Sato and M.-A. Fukase, "Reconfigurable hardware implementation of host-based IDS," in *Proc. 9th Asia-Pac. Conf. Commun. (APCC)*, 2003, vol. 2, pp. 849–853.

[64] Z. Trabelsi and K. Shuaib, "NIS04-4: Man in the middle intrusion detection," in *Proc. Global Telecommun. Conf. (GLOBECOM'06)*, 2006, pp. 1–6.

[65] F. Barbhuiya *et al.*, "An active host-based detection mechanism for ARP-related attacks," in *Advances in Networks and Communications*. New York, NY, USA: Springer, 2011, pp. 432–443.

[66] M. S. Song, J. D. Lee, Y.-S. Jeong, H.-Y. Jeong, and J. H. Park, "DS-ARP: A new detection scheme for ARP spoofing attacks based on routing trace for ubiquitous environments," *Sci. World J.*, vol. 2014, 2014, Art. no. 264654, 7 pp. [Online]. Available: http://www.hindawi.com/journals/tswj/2014/264654/

[67] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[68] I. Teterin. *ARP Spoofing Defence* [Online]. Available: http://www.securityfocus.com/archive/1/299929, accessed on Sep. 15, 2015.

[69] M. Peradilla and J. W. Atwood, "Secure mobility management application capable of fast layer 3 handovers for MIPv6-non-aware mobile hosts," *IEICE Trans. Commun.*, vol. 97, no. 7, pp. 1375–1384, 2014.

[70] M. V. Tripunitara and P. Dutta, "A middleware approach to asynchronous and backward compatible detection and prevention of ARP cache poisoning," in *Proc. 15th Annu. Conf. Comput. Secur. Appl. Conf. (ACSAC'99)*, 1999, pp. 303–309.

[71] N. Su. (2005). *Streams Programming Guide* [Online]. Available: http://docs.oracle.com/cd/E19683-01/806-6546/

[72] N. Alexiou, S. Basagiannis, P. Katsaros, T. Dashpande, and S. A. Smolka, "Formal analysis of the Kaminsky DNS cache-poisoning attack using probabilistic model checking," in *Proc. IEEE 12th Int. Symp. High-Assur. Syst. Eng. (HASE)*, 2010, pp. 94–103.

[73] S. Son and V. Shmatikov, "The Hitchhiker's guide to DNS cache poisoning," in *Security and Privacy in Communication Networks*. New York, NY, USA: Springer, 2010, pp. 466–483.

[74] J. Stewart, "DNS cache poisoning—The next generation," LURHQ, Myrtle Beach, South Carolina, Tech. Rep., 2003 [Online]. Available: https://www.ida.liu.se/~TDDC03/literature/dnscache.pdf

[75] Y. W. Ju, K. H. Song, E. J. Lee, and Y. T. Shin, "Cache poisoning detection method for improving security of recursive DNS," in *Proc. 9th Int. Conf. Adv. Commun. Technol.*, 2007, vol. 3, pp. 1961–1965.

[76] A. Chopra. (2014). *Man in the Middle (MITM) DNS Spoofing Explained* [Online]. Available: https://rootserv.com/wp-content/uploads/2014/05/man-in-the-middle-mitm-dns-spoofing-explained.pdf

[77] T. Naqash *et al.*, "Protecting DNS from cache poisoning attack by using secure proxy," in *Proc. Int. Conf. Emerging Technol. (ICET)*, 2012, pp. 1–5.

[78] D. Kaminsky, "It's the end of the cache as we know it," in *presentation at Blackhat Briefings*, 2008.

[79] A. Herzberg and H. Shulman, "Antidotes for DNS poisoning by off-path adversaries," in *Proc. 7th Int. Conf. Availability Rel. Secur. (ARES)*, 2012, pp. 262–267.

[80] M. Antonakakis, D. Dagon, X. Luo, R. Perdisci, W. Lee, and J. Bellmor, "A centralized monitoring infrastructure for improving DNS security," in *Recent Advances in Intrusion Detection*. New York, NY, USA: Springer, 2010, pp. 18–37.

[81] K. Park, V. S. Pai, L. L. Peterson, and Z. Wang, "CoDNS: Improving DNS performance and reliability via cooperative lookups," in *Proc. Conf. Symp. Opear. Syst. Des. Implement. (OSDI)*, 2004, vol. 4, p. 14.

[82] L. Poole and V. S. Pai, "ConfiDNS: Leveraging scale and history to improve DNS security," in *Proc. 3rd Workshop Real Large Distrib. Syst. (WORLDS'06)*, 2006, pp. 3–3.

[83] H.-M. Sun, W.-H. Chang, S.-Y. Chang, and Y.-H. Lin, "DepenDNS: Dependable mechanism against DNS cache poisoning," in *Cryptology and Network Security*. New York, NY, USA: Springer, 2009, pp. 174–188.

[84] N. J. AlFardan and K. G. Paterson, "An analysis of depenDNS," in *Information Security*. New York, NY, USA: Springer, 2011, pp. 31–38.

[85] D. J. Bernstein. *DNS Forgery* [Online]. Available: http://cr.yp.to/djbdns/forgery.html, accessed on Sep. 15, 2015.

[86] D. Dagon, M. Antonakakis, P. Vixie, T. Jinmei, and W. Lee, "Increased DNS forgery resistance through 0x20-bit encoding: Security via LeET queries," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 211–222.

[87] R. Perdisci, M. Antonakakis, X. Luo, and W. Lee, "WSEC DNS: Protecting recursive DNS resolvers from poisoning attacks," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN'09)*, 2009, pp. 3–12.

[88] D. Eastlake and M. Andrews, "Domain name system (DNS) cookies," IEFT eastlake-dnsext-cookies-03, 2014.

[89] A. Herzberg and H. Shulman, "Unilateral antidotes to DNS poisoning," in *Security and Privacy in Communication Networks*. New York, NY, USA: Springer, 2012, pp. 319–336.

[90] P. Vixie, O. Gudmundsson, D. Eastlake, and B. Wellington, "Secret key transaction authentication for DNS (TSIG)," IETF, Fremont, California, USA, RFC 2845, Tech. Rep., May 2000.

[91] D. E. Eastlake *et al.*, "DNS request and transaction signatures (SIG (0) s)," RFC2931, 2000.

[92] M. Shrivastava and V. Singh, "DNS server cryptography using symmetric key cryptography," *Int. J. Innov. Adv. Comput. Sci.*, vol. 3, pp. 21–25, 2014.

[93] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS security introduction and requirements," RFC 4033, Tech. Rep., Mar. 2005.

[94] H. Shulman and M. Waidner, "Towards forensic analysis of attacks with DNSSEC," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, 2014, pp. 69–76.

[95] S. Ariyapperuma and C. J. Mitchell, "Security vulnerabilities in DNS and DNSSEC," in *Proc. 2nd Int. Conf. Availability Rel. Secur. (ARES)*, 2007, pp. 335–342.

[96] S. Krishnaswamy, W. Hardaker, and R. Mundy, "DNSSEC in practice: Using DNSSEC-tools to deploy DNSSEC," in *Proc. Conf. Homeland Secur. Cybersecur. Appl. Technol. (CATCH'09)*, 2009, pp. 3–15.

[97] D. J. Bernstein. (2009). *DNSCurve: Usable security for DNS* [Online]. Available: http://dnscurve.org

[98] M. Anagnostopoulos, G. Kambourakis, E. Konstantinou, and S. Gritzalis, "DNSSEC vs. DNSCurve: A side-by-side comparison," in *Situational Awareness in Computer Network Defense: Principles, Methods and Applications: Principles, Methods and Applications*. Hershey, PA, USA: IGI Global, 2012, p. 201.

[99] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*. New York, NY, USA: Springer, 2001, pp. 213–229.

[100] S. Duangphasuk, S. Kungpisdan, and S. Hankla, "Design and implementation of improved security protocols for DHCP using digital certificates," in *Proc. 17th IEEE Int. Conf. Netw. (ICON)*, 2011, pp. 287–292.

[101] H. Mukhtar, K. Salah, and Y. Iraqi, "Mitigation of DHCP starvation attack," *Comput. Elect. Eng.*, vol. 38, no. 5, pp. 1115–1128, 2012.

[102] Y. Bhaiji. (2007). *Understanding, Preventing, and Defending Against Layer 2 Attacks*, Cisco [Online]. Available: http://www.nanog.org/meetings/nanog42/presentations/Bhaiji_Layer_2_Attacks.pdf

[103] D. D. Dinu and M. Togan, "DHCP server authentication using digital certificates," in *Proc. 10th Int. Conf. Commun. (COMM)*, 2014, pp. 1–6.

[104] R. Droms *et al.*, "Authentication for DHCP messages; RFC3118. txt," IETF Standard, Internet Engineering Task Force, IETF, CH, 2001.

[105] T. Komori and T. Saito, "The secure DHCP system with user authentication," in *Proc. 27th Annu. IEEE Conf. Local Comput. Netw. (LCN)*, 2002, pp. 123–131.

[106] K. De Graaf, J. Liddy, P. Raison, J. C. Scano, and S. Wadhwa, "Dynamic host configuration protocol (DHCP) authentication using challenge handshake authentication protocol (CHAP) challenge," U.S. Patent 8 555 347, Oct. 8, 2013.

[107] H. Ju and J. Han, "DHCP message authentication with an effective key management," *World Acad. Sci. Eng. Technol.*, vol. 8, pp. 570–574, 2007.

[108] K. Hornstein, T. Lemon, B. Aboba, and J. Trostle, "DHCP authentication via Kerberos V," IEFT draft-hornstein-dhc-kerbauth-02. txt, 2000.

[109] M. Wong, Y. Xu, and S. Manning, "An authentication method based on certificate for DHCP," IEFT draft-xu-dhc-cadhcp-01, 2011.

[110] G. Glazer, C. Hussey, and R. Shea, "Certificate-based authentication for DHCP," 2003 [Online]. Available: https://www.researchgate.net/publication/265190277_Certificate-Based_Authentication_for_DHCP

[111] T. Aura, M. Roe, and S. J. Murdoch, "Securing network location awareness with authenticated DHCP," in *Proc. 3rd Int. Conf. Secur. Privacy Commun. Netw. Workshops (SecureComm)*, 2007, pp. 391–402.

[112] C. Shue *et al.*, "A unified approach to intra-domain security," in *Proc. Int. Conf. Comput. Sci. Eng. (CSE'09)*, 2009, vol. 3, pp. 219–224.

[113] C. Catalyst, "6500 series switches," Catalyst Switched Port Analyzer (SPAN) Configuration Example [Online]. Disponível em: http://www.cisco. com/image/gif/paws/10570/41. pdf, Acesso em vol. 20, 2012.

[114] M. Oche, M. K. Nasir, A. B. Tambawal, and R. M. Noor, "Securing VOiP network: An overview of applied approaches and analysis," in *Proc. Pan Afr. Int. Conf. Inf. Sci. Comput. Telecommun. (PACT)*, 2013, pp. 104–109.

[115] M. Tanase. (1674). *IP Spoofing: An Introduction* [Online]. Available: http://www.securityfocus.com/infocus/

[116] B. Liu, J. Bi, and A. V. Vasilakos, "Toward incentivizing anti-spoofing deployment," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 3, pp. 436–450, Mar. 2014.

[117] P. Ferguson, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," RFC2827, 2000.

[118] F. Baker and P. Savola, "Ingress filtering for multihomed networks," BCP 84, RFC 3704, Tech. Rep., Mar. 2004.

[119] G. Yao, J. Bi, and P. Xiao, "Vase: Filtering IP spoofing traffic with agility," *Comput. Netw.*, vol. 57, no. 1, pp. 243–257, 2013.

[120] S. Patel, "Various anti IP spoofing techniques," *J. Eng. Comput. Appl. Sci.*, vol. 4, no. 1, pp. 7–31, 2015.

[121] T. Ehrenkranz and J. Li, "On the state of IP spoofing defense," *ACM Trans. Internet Technol.*, vol. 9, no. 2, p. 6, 2009.

[122] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 15–26, 2001.

[123] Z. Duan, X. Yuan, and J. Chandrashekar, "Constructing inter-domain packet filters to control IP spoofing based on BGP updates," in *Proc. INFOCOM*, 2006, pp. 1–12.

[124] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, "SAVE: Source address validity enforcement protocol," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2002, vol. 3, pp. 1557–1566.

[125] H. Lee, M. Kwon, G. Hasker, and A. Perrig, "BASE: An incrementally deployable mechanism for viable IP spoofing prevention," in *Proc. 2nd ACM Symp. Inf. Comput. Commun. Secur.*, 2007, pp. 20–31.

[126] X. Liu, A. Li, X. Yang, and D. Wetherall, "Passport: Secure and adoptable source authentication," in *Proc. Netw. Syst. Des. Implement. (NSDI)*, 2008, vol. 8, pp. 365–378.

[127] A. Bremler-Barr and H. Levy, "Spoofing prevention method," in *Proc. 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2005, vol. 1, pp. 536–547.

[128] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 40–53, Feb. 2007.

[129] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet protocol (AIP)," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 339–350, 2008.

[130] A. Yaar, A. Perrig, and D. Song, "PI: A path identification mechanism to defend against DDoS attacks," in *Proc. Symp. Secur. Privacy*, 2003, pp. 93–107.

[131] J. Kwon, D. Seo, M. Kwon, H. Lee, A. Perrig, and H. Kim, "An incrementally deployable anti-spoofing mechanism for software-defined networks," *Comput. Commun.*, vol. 64, pp. 1–20, Jun. 15, 2015.

[132] T. Dierks and C. Allen, "RFC 2246: The TLS protocol," IETF, Jan. 1999.

[133] R. Oppliger, R. Hauser, and D. Basin, "SSL/TLS session-aware user authentication—Or how to effectively thwart the man-in-the-middle," *Comput. Commun.*, vol. 29, no. 12, pp. 2238–2246, 2006.

[134] T. S. Prentow and M. V. Krarup. *MITM Attacks on SSL/TLS Related to Renegotiation* [Online]. Available: http://goo.gl/FHdzKJ, accessed on Sep. 15, 2015.

[135] K. Benton, J. Jo, and Y. Kim, "SignatureCheck: A protocol to detect man-in-the-middle attack in SSL," in *Proc. 7th Annu. Workshop Cyber Secur. Inf. Intell. Res.*, 2011, p. 60.

[136] J. Du, X. Li, and H. Huang, "A study of man-in-the-middle attack based on SSL certificate interaction," in *Proc. 1st Int. Conf. Instrum. Meas. Comput. Commun. Control*, 2011, pp. 445–448.

[137] R.-G. Holz, "Empirical analysis of public key infrastructures and investigation of improvements," Ph.D. dissertation, Comput. Sci., Universitätsbibliothek der TU München, München, Germany, 2014.

[138] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: Validating SSL certificates in non-browser software," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 38–49.

[139] C. Soghoian and S. Stamm, "Certified lies: Detecting and defeating government interception attacks against SSL (short paper)," in *Financial Cryptography and Data Security*. New York, NY, USA: Springer, 2012, pp. 250–259.

[140] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor, "Crying wolf: An empirical study of SSL warning effectiveness," in *Proc. USENIX Secur. Symp.*, 2009, pp. 399–416.

[141] I. Dacosta, M. Ahamad, and P. Traynor, "Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties," in *Computer Security—ESORICS*. New York, NY, USA: Springer, 2012, pp. 199–216.

[142] A. Langley. *Certificate Transparency. ImperialViolet* [Online]. Available: http://www.imperialviolet.org/2012/11/06/certtrans.html, accessed on Sep. 15, 2015.

[143] B. International Computer Science Institute (ICSI). *The ICSI Certificate Notary* [Online]. Available: http://notary.icsi.berkeley.edu, accessed on Sep. 15, 2015.

[144] J. B. a. i. P. EFF. Eff. *The EFF SSL Observatory* [Online]. Available: https://www.eff.org/observatory, accessed on Sep. 15, 2015.

[145] R. Holz, T. Riedmaier, N. Kammenhuber, and G. Carle, "X. 509 forensics: Detecting and localising the SSL/TLS men-in-the-middle," in *Computer Security—ESORICS*. New York, NY, USA: Springer, 2012, pp. 217–234.

[146] C. Evans and C. Palmer, "Public key pinning extension for HTTP," RFC 7469, 2011.

[147] M. Marlinspike, "Trust assertions for certificate keys," IEFT draft-ietf-perrin-tls-tack, 2013.

[148] A. Langley, "Public key pinning," ImperialViolet (blog), 2011.

[149] P. Hoffman and J. Schlyter, "The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: TLSA," RFC 6698, Tech. Rep., Aug. 2012.

[150] D. Wendlandt, D. G. Andersen, and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing," in *Proc. USENIX Annu. Tech. Conf.*, 2008, pp. 321–334.

[151] M. Marlinspike, "SSL and the future of authenticity," Black Hat USA, 2011.

[152] M. Alicherry and A. D. Keromytis, "DoubleCheck: Multi-path verification against man-in-the-middle attacks," in *Proc. IEEE Symp. Comput. Commun.*, 2009, pp. 557–563.

[153] K. Engert. *Detector.io* [Online]. Available: http://detector.io, accessed on Sep. 15, 2015.

[154] S. Puangpronpitag and N. Sriwiboon, "Simple and lightweight HTTPS enforcement to protect against SSL striping attack," in *Proc. 4th Int. Conf. Comput. Intell. Commun. Syst. Netw. (CICSyN)*, 2012, pp. 229–234.

[155] J. Hodges, C. Jackson, and A. Barth. (2012). *HTTP Strict Transport Security (HSTS)* [Online]. Available: http://tools.ietf.org/html/draft-ietf-websec-strict-transport-sec-04

[156] B. Sugavanesh, H. Prasath, and S. Selvakumar, "SHS-HTTPS enforcer: Enforcing HTTPS and preventing MITM attacks," *ACM SIGSOFT Softw. Eng. Notes*, vol. 38, no. 6, pp. 1–4, 2013.

[157] M. Conti, N. Dragoni, and S. Gottardo, "MITHYS: Mind the hand you shake-protecting mobile devices from SSL usage vulnerabilities," in *Security and Trust Management*. New York, NY, USA: Springer, 2013, pp. 65–81.

[158] M. Dietz, A. Czeskis, D. Balfanz, and D. S. Wallach, "Origin-bound certificates: A fresh approach to strong client authentication for the web," in *Proc. USENIX Secur. Symp.*, 2012, pp. 317–331.

[159] D. Balfanz and R. Hamilton, "Transport layer security (TLS) channel IDs," IETF Draft, 2013.

[160] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC4271, 1995.

[161] S. Murphy, "BGP security vulnerabilities analysis," RFC 4272, 2006.

[162] A. Pilosov and T. Kapela, "Stealing the Internet: An Internet-scale man in the middle attack," NANOG-44, Los Angeles, CA, USA, Oct. 2008, pp. 12–15.

[163] J. Genachowski. *FCC CSRIC Makes Recommendations Regarding ISP Cyber Security* [Online]. Available: http://www.techlawjournal.com/topstories/2012/20120322.asp, accessed on Sep. 15, 2015.

[164] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the Internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 265–276, 2007.

[165] T. Mizuguchi and T. Yoshida, "Inter-domain routing security BGP route hijacking," in *Proc. Asia Pac. Reg. Internet Conf. Oper. Technol. (APRICOT'07)*, 2007 [Online]. Available: https://www.apricot.net/apricot2007/presentation/conference/security_stream/BGP-Route-Hijacking-0301.pdf

[166] S.-C. Hong, H. Ju, and J. W.-K. Hong, "Network reachability-based IP prefix hijacking detection," *Int. J. Netw. Manage.*, vol. 23, no. 1, pp. 1–15, 2013.

[167] C. Hepner and E. Zmijewski, "Defending against BGP man-in-the-middle attacks," in *Proc. Black Hat DC Conf.*, 2009.

[168] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and whisper: Security mechanisms for BGP," in *Proc. 1st Symp. Netw. Syst. Des. Implement. (NSDI)*, 2004, p. 11.

[169] G. Huston, M. Rossi, and G. Armitage, "Securing BGP—A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 199–222, May 2011.

[170] Y. Liux, J. Su, and R. K. Chang, "LDC: Detecting BGP prefix hijacking by load distribution change," in *Proc. IEEE 26th Int. Conf. Parallel Distrib. Process. Symp. Workshops PhD Forum (IPDPSW)*, 2012, pp. 1197–1203.

[171] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the Internet with argus," in *Proc. ACM Conf. Internet Meas. Conf.*, 2012, pp. 15–28.

[172] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," in *Proc. Usenix Secur. Symp.*, 2006, pp. 153–166.

[173] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based detection of anomalous BGP messages," in *Recent Advances in Intrusion Detection*. New York, NY, USA: Springer, 2003, pp. 17–35.

[174] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting IP prefix hijacks in real-time," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 277–288, 2007.

[175] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "ISPY: Detecting IP prefix hijacking on my own," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 327–338, 2008.

[176] X. Hu and Z. M. Mao, "Accurate real-time identification of IP prefix hijacking," in *Proc. IEEE Symp. Secur. Privacy (SP'07)*, 2007, pp. 3–17.

[177] S.-C. Hong, J. W.-K. Hong, and H. Ju, "IP prefix hijacking detection using the collection of as characteristics," in *Proc. 13th Asia-Pac. Conf. Netw. Oper. Manage. Symp. (APNOMS)*, 2011, pp. 1–7.

[178] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes," in *Proc. 14th IEEE Int. Conf. Netw. Protocols (ICNP'06)*, 2006, pp. 290–299.

[179] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, Apr. 2000.

[180] K. Seo, C. Lynn, and S. Kent, "Public-key infrastructure for the secure border gateway protocol (S-BGP)," in *Proc. DARPA Inf. Survivability Conf. Expo. II (DISCEX'01)*, 2001, vol. 1, pp. 239–253.

[181] R. White, "Securing BGP through secure origin BGP (SOBGP)," *Bus. Commun. Rev.*, vol. 33, no. 5, pp. 47–53, 2003.

[182] P. C. van Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure BGP (psBGP)," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, p. 11, 2007.

[183] M. Wahlisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP route hijacking using the RPKI," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 103–104, 2012.

[184] M. Lepinski, "BGPsec protocol specification," draft-ietf-sidr-bgpsecprotocol, 2014.

[185] D. McPherson, E. Osterweil, S. Amante, and D. Mitchell, "Route-leaks & MITM attacks against BGPSEC," draft-ietf-grow-simple-leak-attack-bgpsec-no-help-01, 2014.

[186] C. Mitchell, "The security of the GSM air interface protocol," Univ. of London, Royal Holloway, Egham, Surrey, RHUL-MA-2001-3, 2001.

[187] S. Yubo, H. Xili, and L. Zhiling, "The GSM/UMTS phone number catcher," in *Proc. 3d Int. Conf. Multimedia Inf. Netw. Secur. (MINES)*, 2011, pp. 520–523.

[188] J. Pousada-Carballo, F. Gonzblez-Castaiio, and F. I. de Vicente, "Jamming system for mobile communications," *Electron. Lett.*, vol. 34, no. 22, pp. 2166–2167, 1998.

[189] J. G. Sempere. *An Overview of the GSM System* [Online]. Available: http://www.comms.eee.strath.ac.uk/ gozalvez/gsm/gsm.html, accessed on Sep. 15, 2015.

[190] K. P. Kumar, G. Shailaja, A. Kavitha, and A. Saxena, "Mutual authentication and key agreement for GSM," in *Proc. Int. Conf. Mobile Bus. (ICMB'06)*, 2006, p. 25.

[191] K. Vedder, "Security aspects of mobile communications," in *Computer Security and Industrial Cryptography*. New York, NY, USA: Springer, 1993, pp. 193–210.

[192] J. Vales-Alonso *et al.*, "Real-time interception systems for the GSM protocol," *IEEE Trans. Veh. Technol.*, vol. 51, no. 5, pp. 904–914, Sep. 2002.

[193] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of GSM encrypted communication," in *Advances in Cryptology–CRYPTO*. New York, NY, USA: Springer, 2003, pp. 600–616.

[194] A. Peinado, "Privacy and authentication protocol providing anonymous channels in GSM," *Comput. Commun.*, vol. 27, no. 17, pp. 1709–1715, 2004.

[195] C.-C. Chang, J.-S. Lee, and Y.-F. Chang, "Efficient authentication protocols of GSM," *Comput. Commun.*, vol. 28, no. 8, pp. 921–928, 2005.

[196] S. M. Siddique and M. Amir, "GSM security issues and challenges," in *Proc. 7th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel Distrib. Comput.*, 2006, pp. 413–418.

[197] V. Bocan and V. Cretu, "Mitigating denial of service threats in GSM networks," in *Proc. 1st Int. Conf. Availability Rel. Secur. (ARES)*, 2006, p. 6.

[198] M. Toorani and A. Beheshti, "Solutions to the GSM security weaknesses," in *Proc. 2nd Int. Conf. Next Gener. Mobile Appl. Serv. Technol. (NGMAST'08)*, 2008, pp. 576–581.

[199] C. Xenakis, "Security measures and weaknesses of the GPRS security architecture," *IJ Netw. Secur.*, vol. 6, no. 2, pp. 158–169, 2008.

[200] P. Chandra, *Bulletproof Wireless Security: GSM, UMTS, 802.11, and Ad Hoc Security*. Amsterdam, The Netherlands: Elsevier, 2011.

[201] M. Paik, "Stragglers of the herd get eaten: Security concerns for GSM mobile banking applications," in *Proc. 11th Workshop Mobile Comput. Syst. Appl.*, 2010, pp. 54–59.

[202] M. Siddiqui, D. Montero, R. Serral-Gracià, X. Masip-Bruin, and M. Yannuzzi, "A survey on the recent efforts of the Internet standardization body for securing inter-domain routing," *Comput. Netw.*, vol. 80, pp. 1–26, 2015.

[203] D. Strobel, "IMSI catcher," Chair for Communication Security, Ruhr-Universitat Bochum, Bochum, Germany, 2007, p. 14.

[204] B. Luthi. *Improving Security of Mobile Devices* [Online]. Available: https://goo.gl/Vn70yr, accessed on Sep. 15, 2015.

[205] Z. Chen, S. Guo, K. Zheng, and Y. Yang, "Modeling of man-in-the-middle attack in the wireless networks," in *Proc. Wireless Commun. Netw. Mobile Comput.*, 2007, pp. 2255–2258.

[206] OpenBTS.org. *OpenBTS—Open Cource Cellular Infrastructure* [Online]. Available: http://openbts.org, accessed on Sep. 15, 2015.

[207] A. N. I. C. Ettus Research. *Ettus Research—The Leader in Software Defined Radio (SDR)* [Online]. Available: http://www.ettus.com/

[208] M.-S. Hwang, Y.-L. Tang, and C.-C. Lee, "An efficient authentication protocol for GSM networks," in *Proc. IEEE Inf. Syst. Enhanced Public Safety Secur. (EUROCOMM)*, 2000, pp. 326–329.

[209] V. Bocan and B. Cretu, "Threats and countermeasures in GSM networks," *J. Netw.*, vol. 1, no. 6, pp. 18–27, 2006.

[210] A. Fanian, M. Berenjkoub, and T. A. Gulliver, "A TESLA-based mutual authentication protocol for GSM networks," *ISC Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 3–15, 2015.

[211] A. Fanian, M. Berenjkoub, and T. A. Gulliver, "A new mutual authentication protocol for GSM networks," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE'09)*, 2009, pp. 798–803.

[212] C.-C. Lee, I.-E. Liao, and M.-S. Hwang, "An efficient authentication protocol for mobile communications," *Telecommun. Syst.*, vol. 46, no. 1, pp. 31–41, 2011.

[213] I. Ericsson. *Traffic and Market Report* [Online]. Available: http://goo.gl/gKDr9z, accessed on Sep. 15, 2015.

[214] U. Meyer and S. Wetzel, "A man-in-the-middle attack on UMTS," in *Proc. 3rd ACM Workshop Wireless Secur.*, 2004, pp. 90–97.

[215] Z. Ahmadian, S. Salimi, and A. Salahi, "New attacks on UMTS network access," in *Proc. IEEE Wireless Telecommun. Symp. (WTS)*, 2009, pp. 1–6.

[216] CHRISTOFFERSSON. *3G Security; Security Threats and Requirements* [Online]. Available: http://www.3gpp.org/DynaReport/21133.htm, accessed on Sep. 15, 2015.

[217] U. Meyer and S. Wetzel, "On the impact of GSM encryption and man-in-the-middle attacks on the security of interoperating GSM/UMTS networks," in *Proc. 15th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2004, vol. 4, pp. 2876–2883.

[218] Z. Ahmadian, S. Salimi, and A. Salahi, "Security enhancements against UMTS-GSM interworking attacks," *Comput. Netw.*, vol. 54, no. 13, pp. 2256–2270, 2010.

[219] H.-H. Ou, M.-S. Hwang, and J.-K. Jan, "A cocktail protocol with the authentication and key agreement on the UMTS," *J. Syst. Softw.*, vol. 83, no. 2, pp. 316–325, 2010.

[220] S. Wu, Y. Zhu, and Q. Pu, "Security analysis of a cocktail protocol with the authentication and key agreement on the UMTS," *IEEE Commun. Lett.*, vol. 14, no. 4, pp. 366–368, Apr. 2010.

[221] Y.-L. Huang, C.-Y. Shen, and S. W. Shieh, "S-AKA: A provable and secure authentication key agreement protocol for UMTS networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4509–4519, Nov. 2011.

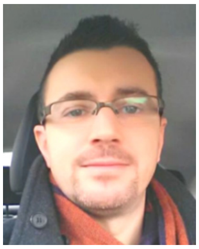
[222] T. Hwang and P. Gope, "Provably secure mutual authentication and key exchange scheme for expeditious mobile communication through synchronously one-time secrets," *Wireless Pers. Commun.*, vol. 77, no. 1, pp. 197–224, 2014.

[223] N. Saxena and N. S. Chaudhari, "NS-AKA: An improved and efficient AKA protocol for 3G (UMTS) networks," in *Proc. Int. conf. Adv. Comput. Sci. Electron. Eng. (CSEE'14)*, Kuala Lampur, Malaysia, 2014, pp. 220–224.

[224] N. Saxena and N. S. Chaudhari, "SAKA: A secure authentication and key agreement protocol for GSM networks," *CSI Trans. ICT*, vol. 1, no. 4, pp. 331–341, 2013.

[225] N. R. Samineni, F. A. Barbhuiya, and S. Nandi, "Stealth and semi-stealth MITM attacks, detection and defense in IPv4 networks," in *Proc. IEEE Int. Conf. Parallel Distrib. Grid Comput. (PDGC)*, 2012, pp. 364–367.

Mauro Conti (S'07–M'08–SM'14) received the Ph.D. degree from Sapienza University of Rome, Rome, Italy, in 2009. After receiving Ph.D. degree, he was a Post-Doc Researcher with Vrije Universiteit Amsterdam, Amsterdam, The Netherlands. In 2011, he joined the University of Padua, Padua, Italy, where he is now an Associate Professor. He has been a Visiting Researcher at GMU (2008), UCLA (2010), UCI (2012, 2013, and 2014), and TU Darmstadt (2013). His research interests include security and privacy. In this area, he published more than 100 papers in topmost international peer-reviewed journals and conference. He is an Associate Editor for several journals, including the IEEE COMMUNICATIONS SURVEYS & TUTORIALS. He was the Program Chair for TRUST 2015, and the General Chair for SecureComm 2012 and ACM SACMAT 2013. He received a Marie Curie Fellowship (2012) from the European Commission, and a Fellowship from the German DAAD (2013).



Nicola Dragoni received the M.Sc. and Ph.D. degrees in computer science from the University of Bologna, Bologna, Italy, in 2002 and 2006, respectively. He visited the Knowledge Media Institute, Open University, Milton Keynes, U.K., in 2004 and the MIT Center for Collective Intelligence, Cambridge, MA, USA in 2006. In 2007 and 2008, he was Postdoctoral Research Fellow with the University of Trento, Trento, Italy, working on security for mobile systems. In 2009, he joined Technical University of Denmark (DTU), Kgs. Lyngby, Denmark, as an Assistant Professor of security and distributed systems and was promoted to Associate Professor in 2011. Since 2014, he has also been a Professor of computer engineering with Orëbro University, Orëbro, Sweden.

Viktor Lesyk received the M.Sc. degree in security and mobile computing from the Danish Technical University (DTU), Kgs. Lyngby, Denmark, and Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2015. Also, he was an ERASMUS student at the University of Padua, Padua, Italy. His research interests include software security, privacy, cyber warfare, and IoT.