ADAPTIVE AUTHENTICATION SYSTEM TO COUNTERMEASURE THE PROBLEM OF EMAIL SECURITY

A SYNOPSIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE & ENGINEERING

By
PRASHANT KUMAR CHAUHAN
Under Guidance of
PROF. (DR.) APURVA M. SHAH

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA-390002 (INDIA)

SEPTEMBER 2023

ABSTRACT

The continuous and exponentially increasing use of email as the main means of communication within two parties is also increasing risk to the involved parties due to the easily possible attacks on the email system. The core of the whole email system depends on SMTP protocol which came to actual existence only after 1980s. At the time of development of SMTP, present day highly computational security attacks couldn't have been considered which results in the security loophole in the present email system.

In today's world, Email (or electronic mail) has become one of the most widely used means of communication whether it be one to one communication or broadcast communication including personal and professional work. Email has been a cost-efficient tool for information sharing and communication. With increasing use of technology and increasing usage of internet among users, number of emails sent per day is exponentially increasing day by day. It is estimated that more than 300 billion e-mails are expected to be communicated each day in the year 2020. Also, the number of emails send per day is expected to be more than 370 billion till the year 2025.

Email spoofing is an attack in which an attacker sends an email that has been modified to seem as if it has been originated from a different trusted source other than the original sender. Email spoofing is a popular attack used in spam campaigns and phishing attacks as a user gets confidence that the email comes from a genuine sender. The main purpose of email spoofing is to trick recipients into opening or responding to the email sent from a spoofed sender. The aim of email spoofing is to trick users into believing that the email has come from some person that they actually know or someone whom they can trust. Once the recipient gets convinced about the sender of the email, the attacker asks about some kind of information that he can misuse in some way.

In my Ph.D. research, we study the working and effectiveness of various anti-spoofing protocols in present day email system like SPF, DKIM and DMARC and work to propose a much effective way to countermeasure the problem of spoofing of email.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Email	Electronic Mail
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
DKIM	DomainKeys Identified Mail
DMARC	Domain-based Message Authentication, Reporting and Conformance
ARPANET	Advanced Research Projects Agency
IEFT	Internet Engineering Task Force
MUA	Mail User Agent
MSA	Mail Submission Agent
MTA	Mail Transfer Agent
MDA	Mail Delivery Agent
AAF	Adaptive Authentication Framework
DNS	Domain Name Server
EAIC	Email Authentication Identity Code
EAIS	Email Authentication Identity Score
IRN	Initial Random Number
RIN	Random Increment Number
SMS	Sending Mail Server
RMS	Receiving Mail Server
RMD	Receiving Mail Domain
SMU	Sending Mail User
RMU	Receiving Mail User

1 INTRODUCTION

With over billions of active users currently in operation, email has become one of the most important and widely used means of communication for information interchange over the internet [1]. For its over 50-years history, email has been an open medium. If we know someone's email address, we can easily send email to them.

Despite its ubiquity, email was created without security in mind and remains largely insecure even as of today. Though many different protocols have been developed with time to overcome the continual problem of email security, none have been able to completely solve all the problems persisting today. Malicious emails pose substantial threats to businesses as well as individuals [2]. Whether it is a malware attachment or a URL leading to malware, exploitation or phishing, attackers have been employing emails as an effective way to gain a foothold inside victim's computer [3] [4]. E-mail sender spoofing is a malicious activity in which the source is being modified and presented as if the E-mail is coming from intended sender whereas the original sender is an attacker. To block the attackers' entrance, it is important for organizations and service providers to strengthen their email security and identify email threats effectively and proactively [5].

In 1971 the first ARPANET email was sent. As of 2017, there were 6.3 billion email accounts, owned by 3.7 billion users, sending over 269 billion email messages per day [6] [7]. Email spoofing is a critical step in phishing attacks where the attacker impersonates someone that the victim knows or trusts. By spoofing the email address of a reputable organization or a close friend, the attacker has a better chance to deceive the victim.

To prevent spoofing, there has been an active effort since early 2000 to develop, promote, and deploy anti-spoofing protocols. Protocols such as SPF, DKIM, and DMARC have become Internet standards, allowing email receivers to verify the sender's identity. Global implementation of SPF is reported at 53.8%, DKIM at 38.8%, and DMARC at 46.8% [7].

1.1 Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP) is the Internet standard for all email transmission. SMTP is an application-layer protocol that enables the transmission and delivery of email over the Internet [8] [9]. SMTP is created and maintained by the Internet Engineering Task Force (IETF). It is generally integrated within an email client application and is composed of four key components:

- Local user or client-end utility, known as the Mail User Agent (MUA)
- Server, known as Mail Submission Agent (MSA)
- Mail Transfer Agent (MTA)
- Mail Delivery Agent (MDA)

As shown in *Figure* 1, SMTP works by initiating a session between the user and server, whereas MTA and MDA provide domain searching and local delivery services. A key limitation of SMTP is that it has no built-in security features to prevent people (attackers) from impersonating or spoofing an arbitrary sender address [10].

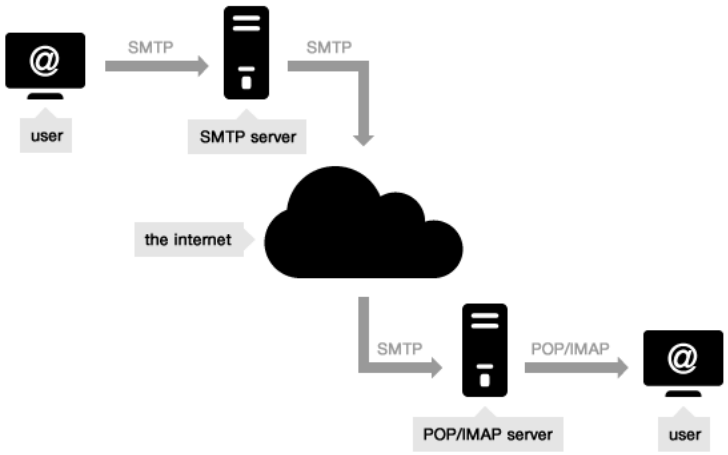

Figure 1: Email Delivery Process

The SMTP permits any computer to send an email claiming to be from any source address. This is exploited by spammers who often use forged email addresses, making it more difficult to trace a message back to its source, and easy for spammers to hide their identity in order to avoid responsibility [11]. It is also used in phishing techniques, where users can be duped into disclosing private information in response to an email purportedly sent by an organization such as a bank.

1.2 Email Authentication

Malware often search for email addresses within the computer they have infected, and use those addresses both as targets for email, but also to create credible forged From fields in the emails that they send, so that these emails are more likely to be opened [12]. Email spoofing is the creation of email messages with a forged sender address. Because the core email protocols do not have any mechanism for authentication, it is common for spam and phishing emails to use such spoofing to mislead the recipient about the origin of the message [13].

To perform a spoofing attack, attackers can manipulate two key fields to send emails. First, after establishing an SMTP connection to the target mail server, the attacker can use the "MAIL FROM" command and set the sender address to anyone that they want to impersonate [14]. After that, the "MAIL FROM" address is inserted into the header as the "Return-Path". In addition, attackers can modify another field called "From" in the email header. This "From" field specifies the address that will be displayed on the email interface. When a user receives the email, the user will see the "From" address. If the user replies to the email, the reply message will go to the "Return-Path" set by "MAIL FROM". Note that the two addresses are not necessarily the same [15].

1.3 Problem Statement, Objectives, Research Contributions

1.3.1 Problem Statement

Email spoofing is continuously increasing with the increase in use of email. Attackers try to take advantage of the low security in authentication of email in SMTP and try to spoof emails to attack the victims in some or other way to take advantage. The purpose of this research is to check whether email spoofing is still possible or not and to find possible solutions to countermeasure this problem of email spoofing.

1.3.2 Objectives of Research

1. Systematic and detailed analysis of the complete email delivery process
2. Study and comparison of algorithms currently in use for the authentication of email
3. Exploration of new ideas to be applicable in the email delivery system to enhance the security in present system
4. Development of an adaptive algorithm to countermeasure the everlasting problem of authentication of email
5. Modeling and implementation of the proposed algorithm and analysis of the results achieved with the use of proposed algorithm
6. Finding the pros and cons of the proposed system and defining the relative future work

1.3.3 Research Contributions

1. Setup of own email server is done along with implementation of existing anti-spoofing protocols SPF, DKIM and DMARK, to test the spoofing of emails.

2. Testing of possibility of email spoofing on top 10 most used email servers considering all possible configurations of the anti-spoofing protocols. With our experiment, it was concluded that even after usage of anti-spoofing protocols by various email servers it is still easily possible to spoof an email.

3. Proposed Adaptive Authentication Framework (AAF) for Prevention and Detection of Spoofing of Email. The proposed framework works in addition to the existing anti-spoofing protocols SPF, DKIM and DMARC.

4. If the existing email servers like Gmail, Outlook etc. adopt our proposed framework, the number of spoofed emails is expected to become negligible. There is no proper measure of the amount lost by individual persons and organizations due to spoofed emails which is expected to be in billions. With the use of our framework, the billions lost because of a spoofed email can be saved along with other non-monetary losses.

2 LITERATURE STUDY

2.1 Email Spoofing

2.1.1 What is Email Spoofing

Email spoofing is an attack in which an attacker sends an email that has been modified to seem as if it has been originated from a different trusted source other than the original sender. Email spoofing is a popular attack used in spam campaigns and phishing attacks as a user gets confidence that the email comes from a genuine sender [16].

The main purpose of email spoofing is to trick recipients into opening or responding to the email sent from a spoofed sender [17]. The aim of email spoofing is to trick users into believing that the email has come from some person that they actually know or someone whom they can trust. Once the recipient gets convinced about the sender of the email, the attacker asks about some kind of information that he can misuse in some way [18].

2.1.2 Why Email Spoofing Happens?

An attacker uses the spoofed email for one or more of the following reasons:

1. To hide the real sender's identity from the recipient.
2. To gain important and sensitive information from individuals or even from organizations by performing man in the middle attack.
3. To pretend to be a genuine and trustworthy organization to get access to private information of any kind like personal details or credit card details.
4. To pretend to be a genuine and trustable sender like someone who the recipient already knows to gather some personal information.
5. To bypass various blocklists and spam filters.
6. To spoil the original sender's reputation.
7. To commit identity theft attack by impersonating a genuine sender.
8. To launch and install various kind of malwares hidden in files attached in emails spoofed to be sent from fake sender.

2.1.3 How does Email Spoofing Happen?

To spoof an email, main requirement is of an SMTP (Simple Mail Transfer Protocol) server and an application for sending an email. The attacker modifies the FROM, REPLY-TO, and RETURN-PATH addresses in the message header while the email is being composed using this kind of application [19]. Now, regardless of its actual source, any person who receives this updated email message will presume it has originated from the falsified address [20].

Email spoofing is only possible because SMTP was not initially intended to validate email sender addresses. Although certain methods have been created over time to avoid and detect email spoofing, their acceptance is quite low [21].

The “Reply-To” field can also be modified or even completely changed by the sender of an email and thus can result in spoofing of email phishing attack. Via the “Reply-To” field the receiving email server gets to know where to send a reply to, and this can be spoofed to be different email id instead of the original sending user’s email address.

The receiving email server or even the SMTP protocol cannot identify or take any action to identify the email as being a faked one in such circumstances, where the reply to address and the sender address are different [21]. The receiving server now has full responsibility for comparing the two addresses and deciding whether it believes the email to be from a reliable sender or not. A user risks falling prey to the attacker, if they fail to recognize an email with a spoofed sender. Before concluding that an email is authentic, users must analyze its source code. The IP address of the user who sent the email can be found in the email's source code, which is accessible to the recipient user. From the source code, the user can verify if the email message has successfully passed the SPF, DKIM and DMARC testes or not [22].

An example of source code of a forged email can be seen in *Figure 2*. As seen in the figure, the “Return-Path” field has the spoofed email address “prashant@gmail.com” which is shown to the user in inbox, while “Received-From” field has domain “mail-server.in” which is the original sending domain.

```
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@mail-server.in header.s=modoboa header.b="MM2L8A5/";
       spf=softfail (google.com: domain of transitioning prashant@gmail.com does
       dmarc=fail (p=NONE sp=QUARANTINE dis=NONE) header.from=gmail.com
Return-Path: <prashant@gmail.com>
Received: from mail-server.in (mail-server.in. [117.240.215.158])
        by mx.google.com with ESMTPS id j70-20020a638049000000b0040d40aa9825si133
        for <prashant.chauhan-cc@msubaroda.ac.in>
        (version=TLS1_3 cipher=TLS_AES_256_GCM_SHA384 bits=256/256);
        Mon, 04 Jul 2022 22:47:45 -0700 (PDT)
Received-SPF: softfail (google.com: domain of transitioning prashant@gmail.com do
Authentication-Results: mx.google.com;
       dkim=pass header.i=@mail-server.in header.s=modoboa header.b="MM2L8A5/";
       spf=softfail (google.com: domain of transitioning prashant@gmail.com does
       dmarc=fail (p=NONE sp=QUARANTINE dis=NONE) header.from=gmail.com
```

Figure 2: Example of a spoofed email

A number of the fields in the email message header are modified by the attacker to spoof emails [23]. A person who wants to spoof an email address can change the address in the "From" field of email header to any desired email address. Then, as the "Return-Path" field, this fake "From" field is added to the email message header.

The email address that is displayed to the recipient user in their inbox is the one in the "From" field of the email header. Now, when a person receives a spoof email, he or she will see the fake "From" email address rather than the real one. Numerous SMTP protocols have been developed in order to stop and detect email spoofing [24]. The three most efficient ones are Sender Policy Framework (SPF), DomainKeys identified Mail (DKIM) and Domain-based Message Authentication, Routing and Conformance (DMARC).

2.2 Anti-Spoofing Protocols

To detect and prevent email spoofing, SMTP extension protocols are proposed including SPF, DKIM, and DMARC. All three protocols have been published or standardized by the Internet Engineering Task Force (IETF).

2.2.1 Sender Policy Framework (SPF)

Sender Policy Framework (SPF) was proposed in early 2000 and standardized in 2014. SPF allows the owner of an Internet domain to specify which computers are authorized to send mail from addresses in that domain, using Domain Name Server (DNS) records. The list of authorized sending hosts and IP addresses for a domain is published in the DNS records for that domain in the form of a specially formatted TXT record [24] [25].

For instance, the domain abc.com can publish its SPF record in the DNS. When the receiving server receives the MAIL FROM command claiming to be xyz@abc.com, the receiving server can check if the sender IP is listed in the SPF record of abc.com or not.

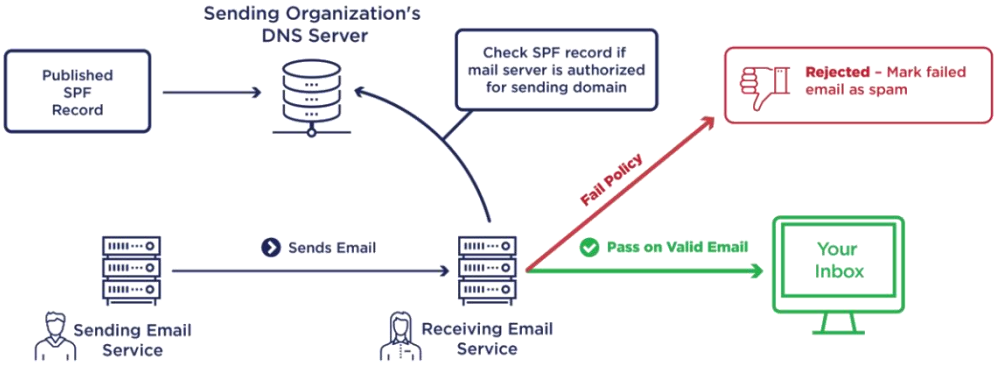

Figure 3: Sender Policy Framework (SPF) Process Flow

Receivers verifying the SPF information in TXT records may reject messages from unauthorized sources before receiving the body of the message. Essentially what SPF checks for is that the person in the 'Received' field is authorized to send email from the server that the email was sent from.

Therefore, an SPF-protected domain is less attractive to spammers and phishers. Because an SPF-protected domain is less attractive as a spoofed address, it is less likely to be blacklisted by spam filters and so ultimately the legitimate email from the domain is more likely to get through.

2.2.2 DomainKeys Identified Mail (DKIM)

DomainKeys Identified Mail (DKIM) is an email authentication method designed to detect email spoofing. It was first drafted in 2004 and standardized in 2011. It allows the receiver to check that an email claimed to have come from a specific domain was indeed authorized by the owner of that domain. It is intended to prevent forged sender addresses in emails, a technique often used in phishing and email spam [4].

DKIM uses a public-key based approach to authenticate the email sender and check the email integrity. By verifying the DKIM signature, the receiver can detect if the signed message has been modified, to ensure integrity and authenticity [25]. This tool attempts to ensure that no changes are made to an email while travelling from sender to receiver.

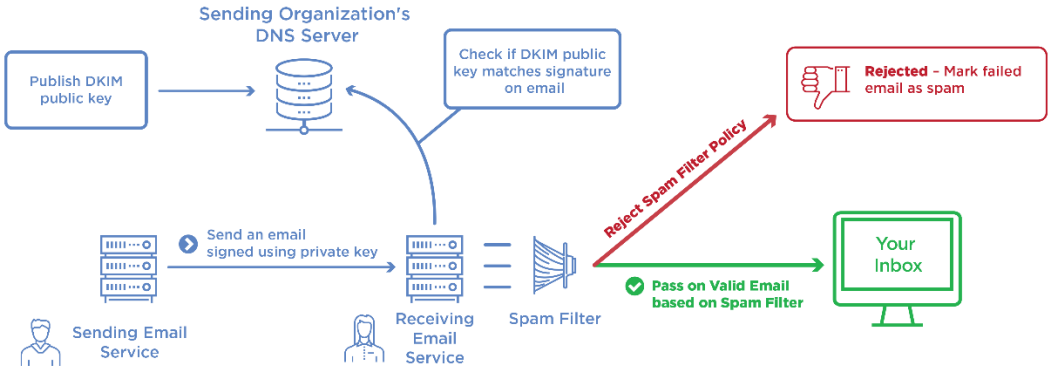

Figure 4: DomainKeys Identified Mail (DKIM) Process Flow

On the sender side the server signs the email message with a private key that is generated from a public key. The public key is stored on the DNS. The receiver side server queries the DNS for the public key and generates a private key based on the same algorithm. It compares the public key, the new private key, and the decrypted private key to determine whether the email was delivered unaltered from the source. Receivers who successfully validate a signature can use information about the signer as part of a program to limit spam, spoofing, phishing, or other undesirable behaviour, although the DKIM specification itself does not prescribe any specific actions by the recipient. A spoofer cannot forge a DKIM signature. The server will flag a spoofed email as failing a DKIM check in the header [27].

The primary advantage of this system for e-mail recipients is in allowing the signing domain to reliably identify a stream of legitimate email, thereby allowing domain-based blacklists and whitelists to be more effective.

2.2.3 Domain-based Message Authentication, Reporting and Conformance (DMARC)

Domain-based Message Authentication, Reporting and Conformance (DMARC) is an email-validation system designed to detect and prevent email spoofing. It was drafted in 2011 and published in 2015. It is intended to combat certain techniques often used in phishing and email spam, such as emails with forged sender addresses that appear to originate from legitimate organizations. It counters the illegitimate usage of the exact domain name in the "From" field of email message headers [5].

DMARC is built on top of two existing mechanisms, SPF and DKIM. It allows the administrative owner of a domain to publish a policy on which mechanism (DKIM, SPF or both) is employed when sending email from that domain and how the receiver should deal with failures [26]. Additionally, it provides a reporting mechanism of actions performed under those policies.

DMARC coordinates the results of DKIM and SPF and specifies under which circumstances the "From" header field, which is often visible to end users, should be considered legitimate. DMARC policies are published in the public Domain Name System (DNS) as text (TXT) resource records (RR) and announce what an email receiver should do with non-aligned mail it receives. It allows the domain owner to publish a "failing policy" which specifies what actions the receiver should take when the incoming email fails the DMARC checks [27].

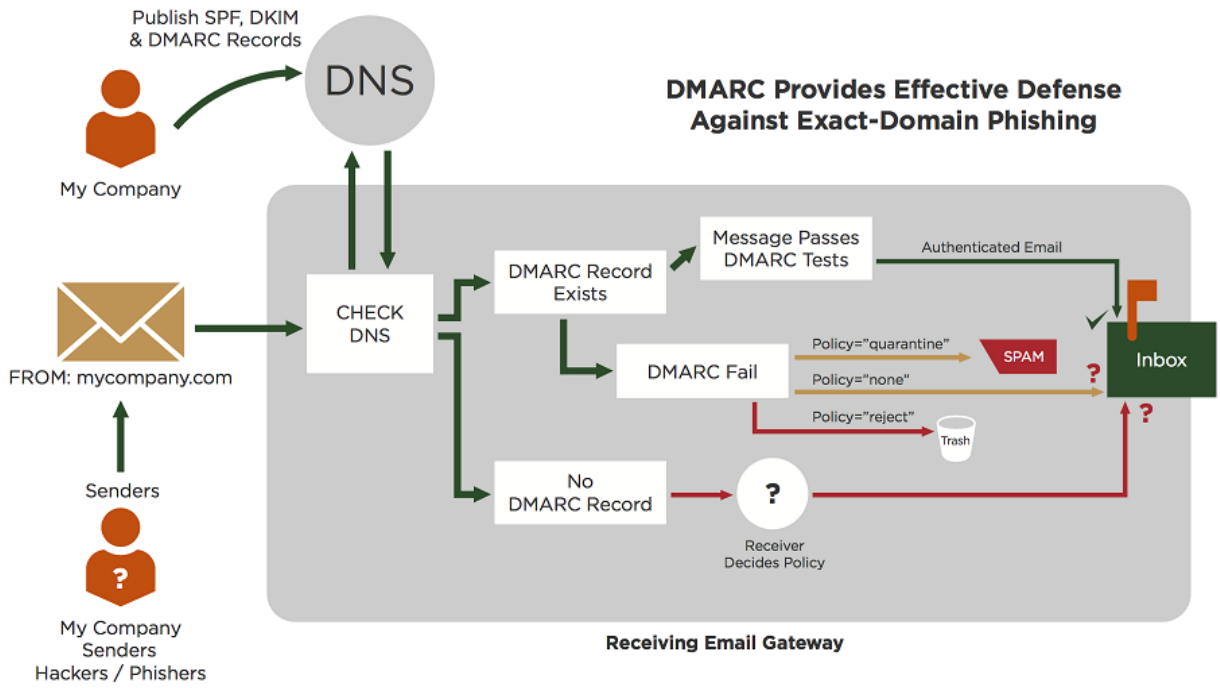

Figure 5: DMARC Process Flow

In addition, DMARC requires identifier alignment from SPF or DKIM. DMARC pulls SPF and DKIM results and uses them to implement policies set up by the email server administrator. A DMARC policy allows a sender's domain to indicate that their emails are protected by SPF and/or DKIM and tells a receiver what to do if neither of those authentication methods passes such as junk or reject the message. It removes guesswork from the receiver's handling of these failed messages, limiting or eliminating the user's exposure to potentially fraudulent & harmful messages [28] [29].

3 GAPS IN EXISTING SYSTEM

Despite these efforts, however, sending spoofing emails is still surprisingly easy today. In 2017, the global implementation of SPF is reported at 53.8%, DKIM at 38.8%, and DMARC at 46.8% [29]. It is found that most email administrators are aware of the technical weaknesses of SPF, DKIM and DMARC which is one of the several reasons for low adoption rate of these protocols. The general perception is that these protocols are "helpful", but "cannot solve the spoofing problem completely". Some of the key limitations found in the existing system are as follows:

1. **Problem of Identifier Alignment**

   SPF and DKIM both have the problem of "identifier alignment". It means that the sender email address that user sees can be different from the address that is actually used to do perform authentication.

   For SPF, the authentication focuses on the "Return-Path" and examines whether the sender's IP is listed in the "Return-Path" domain's SPF record. An attacker can set the "Return-Path" domain to his own domain and set his SPF record to pass the authentication. This is possible because what the receiving user sees on the email interface is set by the "From" field and not by the "Return-Path" field.

   DKIM has a similar problem given that the domain to sign the email with the DKIM key can be different from the domain on the "Return-Path". DMARC helps to resolve the problem by enforcing the alignment of the identifiers.

2. **Mail Forwarding is a Problem for SPF**

   Mail forwarding means one email service automatically forwards emails to another email service. A common scenario is that corporate personnel often configure their company email service to forward all their emails to Outlook or Gmail. During mail forwarding, the email metadata (e.g., "Return-Path") remains unchanged. SPF will fail after mail forwarding because the forwarder's IP will not match the original sender's SPF record.

3. **Mailing List is a Major Problem for both SPF and DKIM**

   When a message is sent to a mailing list, the mailing list will "broadcast" the message to all the subscribers. This is a similar process as mail forwarding. During this process, the mailing lists' IP will become the sender IP, which is different from the original sender's IP [30]. This will lead to SPF failure.

   Mailing lists will also cause trouble for DKIM because most mailing lists modify the email content before broadcasting it to the subscribers. The common modification is to add a "footer" with the mailing list's name and a link for un-subscription. Tempering the email content will cause DKIM failure. DMARC helps to solve some of the problems, but not the mailing list problem.

For mailing lists, DMARC+SPF will be sure to fail — if the "Return-Path" is modified, DMARC will fail due to the misalignment of identifiers; if the "Return-Path" is unmodified, SPF will fail due to the IP mismatch. For DMARC+DKIM, it will fail if the mailing list still has to modify the email content.

4. **Lack of Critical Mass**

   Due to the technique's weaknesses, the general perception is that SFP, DKIM, and DMARC are "helpful" but "cannot solve the spoofing problem completely" which is the reason for low adaption rate of these protocols. If everyone doesn't use these protocols the overall effect is diminished.

5. **No Penalty for not Publishing SPF, DKIM, and DMARC**

   There is no penalty to domains for not publishing an SPF/DKIM/DMARC record. Their emails are typically not discriminated unless other malicious signals are detected. Thus, all people are not encouraged or motivated for active usage of these protocols.

6. **Benefits not Significantly Overweight Costs**

   The protocol adopter does not directly benefit from publishing their SPF, DKIM or DMARC records in the DNS. Instead, these DNS records mainly help other email services to verify incoming emails and protect the customers (users) of other email services. Domains that publish the DNS records receive the benefit of a better reputation, which is a relatively vague benefit. For non-email domains (e.g., office.com), the benefit of publishing the SPF/DMARC record is to prevent attackers from impersonating the non-email domain and helps the non-email domain to maintain a good reputation [31].

   The domain administrators publish the SPF/DMARC records to be a good Internet "citizen" and help other email services to detect spoofing emails. However, these benefits are considered indirect and thus relatively weaker.

7. **Lack of Control on the DNS or Mail Servers**

   Certain services do not have a control over their DNS record. Publishing SPF/DKIM/DMARC record will incur additional overhead to coordinate with their DNS providers. In addition, many companies and organizations even don't maintain their own mail servers but rely on cloud-based email services. The organization needs to rely on the cloud email service to deploy the anti-spoofing protocols [32].

8. **Perception of Difficulty**

   There is a general perception that deploying anti-spoofing protocols is difficult. Regardless of the actual level of the difficulty, the perceived difficulty makes email administrators hesitated to even try.

9. **Risks of Breaking the Existing System**

   Email providers need to go through careful testing to make sure that the protocol does not block legitimate incoming emails, and their own emails are not blocked by others [33]. This is the reason why most protocol adopters (as the sender domain) configure a relaxed SPF/DMARC policy. Even if sender domain specified a strict protocol, the receiver may not enforce it anyway [34].

10. **Lack of External 3rd Party Verification of SPF Records**

    Anyone can sign the SPF records of a domain. Verisign, for example, goes out and verifies websites to make sure that they are secure when they sign up for SSL. If you aren't a good website, no "Verified by Verisign" for you [35]. However, there is no equivalent "Signed by SPF" authority that makes sure that whoever signs up for it truly deserves to get it. By means of this loophole, a scammer could get past the SPF portion of email security by simply creating a domain on rented servers (perhaps in another country) and creating their own SPF record for that domain. Thus, when the target server sends a DNS query the domain checks out and the email passes unmarked through the SPF system.

11. **Typo-Squatted Domain**

    A spoofer can set up an email server with a typo-squatted domain and set up his own DKIM public/private key system. This will bypass DKIM's protections, so it is best used as part of a layered defence system.

4 RESEARCH WORK

4.1 Setup of Own Email Server

The technical specifications of our Email Server are as follows:
Domain Name: https://www.mail.mail-server.in
Public IP Address: 117.240.215.158
Email Server: Modoboa version 1.17.0
Operating System: Ubuntu 20.04.3 LTS
Database Server: PostgreSQL
Web Server: Nginx
Other Important Packages Installed: Dovecot, Postfix, OpenDKIM

For successful set of any email server also required the reverse DNS of IP address of the server set to the specified domain name [25]. The reverse DNS entry can only be done by the ISP providing the public IP address which is BSNL in our case. So, with the help of BSNL we setup reverse DNS entry for IP address 117.240.215.158 to point to our mail server i.e. mail-server.in

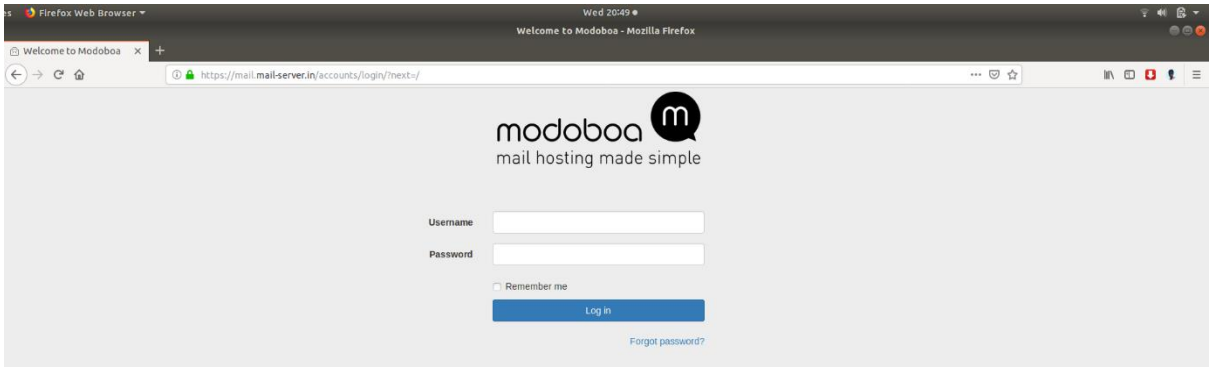

Figure 6: mail-server.in Email Server

4.2 Implementation of Anti-Spoofing Protocols

After the successful setup of our own email server, we set up all the three anti-spoofing algorithms in our email server. For this we published respective entries of SPF, DKIM and DMARC in DNS of our domain mail-server.in as shown in *Figure 7,* Figure 8 *and* Figure 9 respectively.

mail-server.in.	3600	IN	TXT	v=spf1 ip4:117.240.215.158

Figure 7: DNS entry for SPF

default._domainkey.mail-server.in.	14400	IN	TXT	v=DKIM1; k=rsa; p=MIICIjANBgk qhkiG9w0BAQEFAAOCAg8AMIICCg KCAgEAl2yboh/jZ7GiAlUdhK1Lukr6 RX2sRwxSmxOYK55sJfE2lMSKD29 45AHOWodb5OTWhZ+hj0Ognpb1e /quRyAEVIQLKdTS+Dotn93ngtapQ 6tEiODjf41bCfquIBOJtRaNe0KjweiZ YN3ogMLKIfT46fmgw6ww+OH2F4 pgHXGFXLZW9FXLcn21H9lhLopSz mv9npQYWxGrCABgjDO9f9isweGj WmNegCD88rcugIk5kzxcgW0XAR2 EoU7IiqUVOHx4TlHC0VMiNagbpeo 0K+dENeC3DU/h6Z1RmRZQPgtRi4 vFZ4jzUhr534GBgWX3UQC4cTsnG yRq00isBKkjF1WW1P1yT0EBHkP7d UP3IFYJTl7s9y0mjUXjtTAp610Iu4Q scMwsKZLE4UxE4PDeMP2gfMYQw mipbnb9e9EwrXeasZ0seTZPPFDrm c4VdlczVOqwiq2OSGt7KHMjxNKj8 ZmIQSvvolS/XrtGtZ21iAU+Tl6iFpB F/WgW7X0HTsRQVTT3MNCCGRkoP NryFwPczOqWH9bw2f7GL3CGLBtjX 43PE36MQ1c+ppmUaWbzIfOzDP6L Gbt2XqGW0Xte4+9iTsEeye4ak6cD BS7+uv4EC4OcMPCCGlxTEUzRkTV gFt02UOv79YO6Kbvgq4q8dy2H7lL x1dpdwxq2w+JGOVLy0SB+8XkCA wEAAQ==

Figure 8: DNS entry for DKIM

_dmarc.mail-server.in.	14400	IN	TXT	v=DMARC1;p=none;sp=none;adkim=r;aspf=r;

Figure 9: DNS entry for DMARC

4.3 Testing of Delivery of Email

After the successful implementation of all the three anti-spoofing protocols, SPF, DKIM and DMARC to our email server we tested the delivery of emails to various other email servers sent from our own email server as shown in *Figure 10*. On our email server we added various other domain names also for testing email spoofing and we were successfully able to send spoofed email to the inbox of recipients as shown in *Figure 11*.

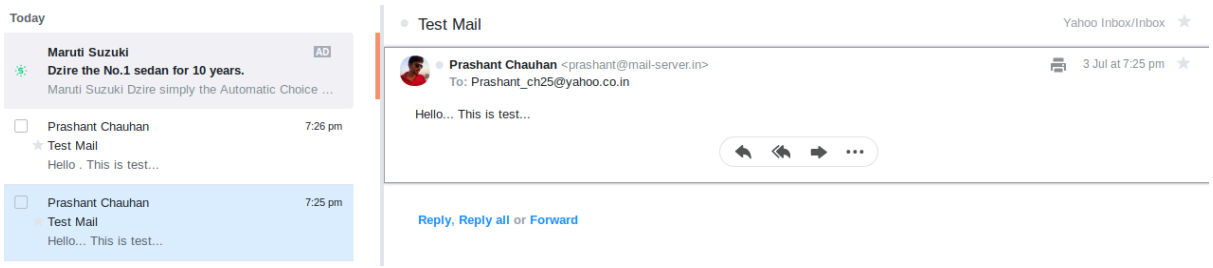

Figure 10: Normal genuine email sent from mail-server.in

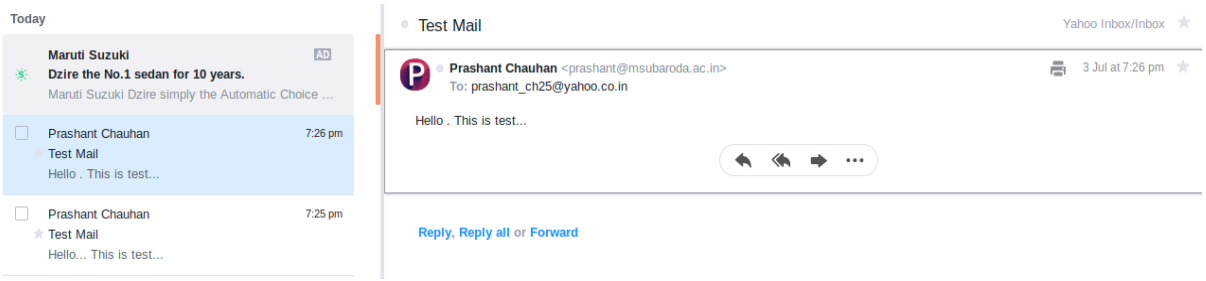

Figure 11: Spoofed email sent from mail-server.in

4.4 Attacks on Anti-Spoofing Protocols

Since the inception of these anti-spoofing protocols, attackers are continuously trying to find ways to bypass one or more of these protocols and they have been successful to bypass all of these anti-spoofing protocols i.e. Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM) and Domain-based Message Authentication, Routing and Conformance (DMARC) by adopting different mechanisms.

4.4.1 How to Bypass SPF?

To bypass the SPF, we took advantage of the fact that SMTP uses two separate domains for displaying “mail-from” to users and as actual “mail-from”. The “mail-from” written in the email body is used to check the SPF test as well as to display as sender to the recipient user. The “mail-from” sent as header is used to get the actual domain the email came from.

As shown in *Figure 12*, we sent “prashant@mail-server” as the actual sender email address in the spoofed email and “prashant@gmail.com” as the spoofed sender email address.

The spoofed email address was displayed to the recipient user as “sender”, while the domain “mail-server.in” was used for SPF check which successfully passed as it was already configured for our own IP address.

```
sender = 'prashant@mail-server.in'
receivers = ['prashantchauhan25@gmail.com']

message = """From: Prashant Chauhan <prashant@gmail.com>
Subject: Meeting Reminder 12

Sir, reminder again..
Please find below the updated schedule for the meeting:
Date: 10 Nov 2020
Time: 10 am onwards
```

Figure 12: Part of python script used to send spoofed email

4.4.2 How to Bypass DKIM?

To bypass the DKIM, we added a signature encrypted by our own private key for the domain "mail-server.in", along with the email as shown in *Figure 13*. Once, the receiving server receives the email it tried to find the domain which sent the email, and it gets our original email server as that domain name.

Then, it checks our DNS entry for finding the public key of our domain and tries to decrypt the signature with the public key. As the signature gets decrypted successfully, it confirms that this email came from an authentic user and the DKIM test is passed successfully.

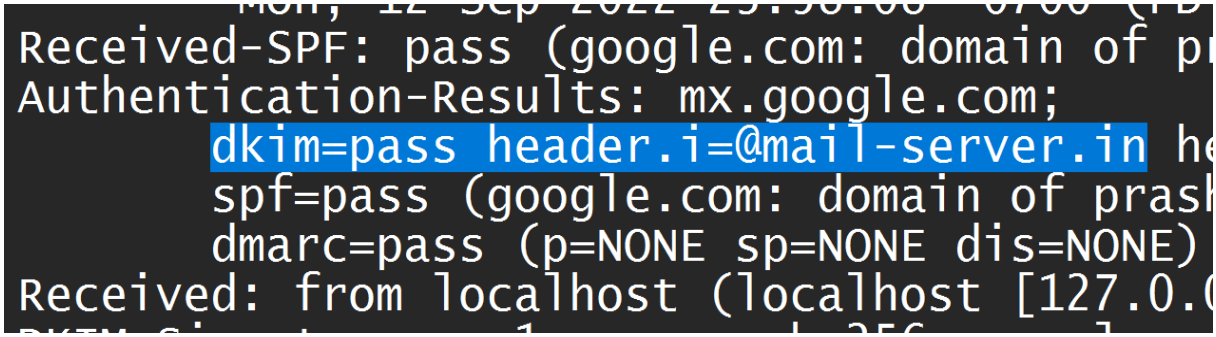

Figure 13: Raw Source Code Message of a spoofed email

4.4.3 How to Bypass DMARC?

To bypass the DMARC, we tried to insert multiple email addresses with the "mail-from" field in the email one of them being the spoofed email address and the other one email address of our own email server as shown in *Figure 14*. As one of the email addresses in the "mail-from" field is the actual originating email address, the receiving email server performs DMARC test on our original email address resulting in passing of DMARC test. Since one of the email addresses in the "mail-from" field is same as the one used for checking SPF and DKIM, identifier alignment is passed even in case of spoofed email.

```
sender = 'prashant@mail-server.in'
receivers = ['prashantchauhan25@gmail.com']

message = """From: Prashant Chauhan <prashant@gmail.com>
<prashant@mail-server.in>
To: <prashantchauhan25@gmail.com>
Subject: Meeting Reminder 12
```

Figure 14: Part of python script used to send spoofed email with multiple email ids

To further try to bypass the authentication protocols, we tried to send the spoofed email with multiple sender email ids, and we found that DMARC is now passed in many cases using multiple sender email ids which was not passed when spoofing using single sender email id.

4.5 Adaptive Authentication Framework (AAF)

To overcome the issue of spoofing of email we propose a new adaptive authentication framework which will work in addition to the existing protocols for authentication of email.

All the existing authentication protocols need each domain owner to publish the records in their own DNS to countermeasure the spoofing of email. On the other hand, our proposed framework doesn't require any effort from domain owners, and only requires the email servers to make the necessary changes. We define a new term Email Authentication Identity Code (EAIC) which works at the core of our proposed authentication framework. The EAIC consists of an Initial Random Number (IRN) of 16 digits along with a 4-digit Random Increment Number (RIN).

The process of generation of EAIC code used when sending mail user wants to send an email to receiving mail user, is as follows:

1. If this is the first time the sending mail user is sending an email to the receiving mail server, a new value for 16-digit Initial Random Number is generated for the pair (sending mail user and receiving mail server) and stored at the sending email server.
2. Also, a random increment number is generated and stored at the sending email server for the same pair.
3. If this is the first time the sending mail user is sending an email to the receiving mail server, the Initial Random Number is considered as the EAIC code.
4. If the sending mail user has already sent an email to the receiving mail server, EAIC code will be already existing. The new EAIC code will be the addition of the previous EAIC code and the Random increment Number.
5. Once an EAIC code is generated for the given pair, email is sent to the receiving email server from the sending mail user along with the EAIC code attached as header to the email body.

A unique EAIC code is generated per pair of Sending email User and the receiving email server. For instance, if the sending main user "prashant@gmail.com" wants to send an email to any user belonging to the domain "msubaroda.ac.in", an Initial Random number is generated as 8400580227532596 and a Random Increment Number is generated as 2389. Now for the first email sent by "prashant@gmail.com" to any user belonging to the domain "msubaroda.ac.in", the EAIC code sent will be 8400580227532596. For the second mail the EAIC code will become 8400580227534985 i.e., the addition of previously sent EAIC code and the Random Increment Number for the given pair.

In similar way for all subsequent communications from "prashant@gmail.com" to any user belonging to the domain "msubaroda.ac.in", new EAIC code will be calculated based on the previously sent EAIC and the Random Increment Number.

Now suppose the same user, "prashant@gmail.com" wants to send an email to any user belonging to the domain "yahoo.in", a new pair of Initial Random Number and Random

Increment Number will be generated which will be used for generating EAIC code for all communications between this pair.

Now whenever a new email is to be sent, first it is checked whether EAIC code exists for the given pair of sending mail user and receiving mail server. In both cases a new EAIC code is generated and sent along with the email to the receiving mail server. Also, a new term Email Authentication Identity Score (EAIS) is used to find the score of possibility of spoofing of an email. The process of authentication of email using EAIC code is as follows and can also be seen in *Figure 15.*

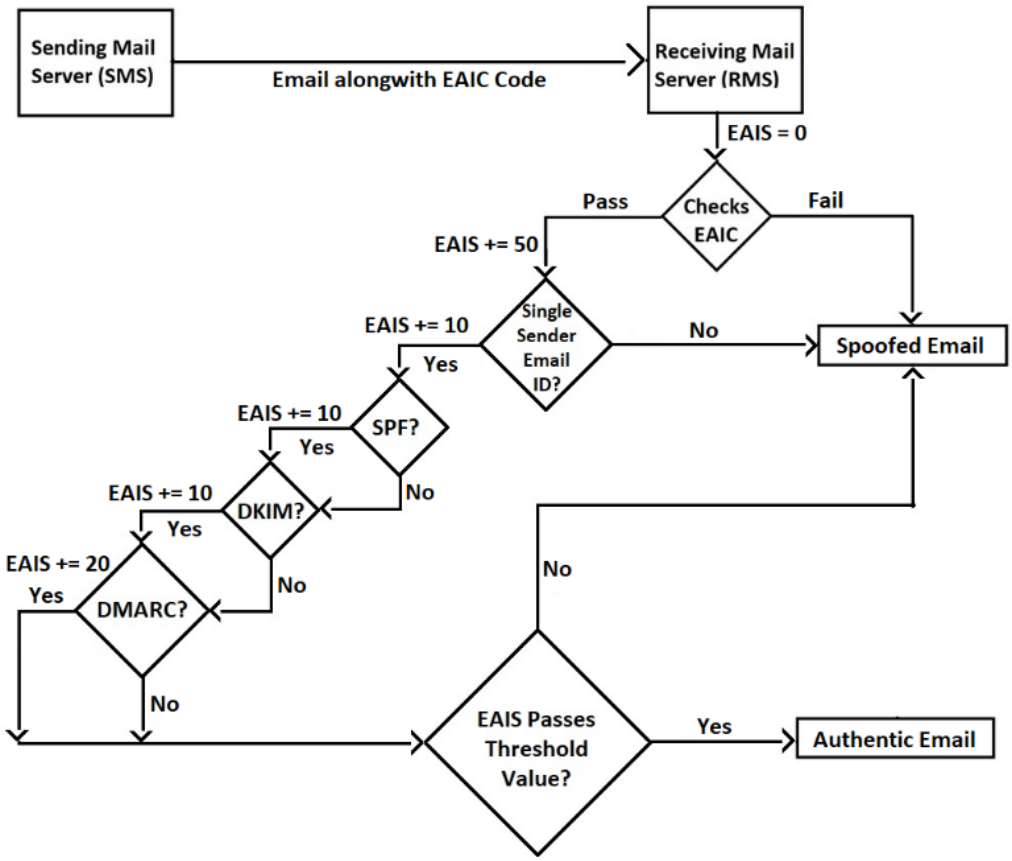

Figure 15: Flowchart for Adaptive Authentication Framework (AAF)

If it is the first time the receiving mail server is receiving an email from the sending mail user, it stores the EMAIC code in its records for future processing. If it is not the first mail from the sending mail user, this means that the receiving mail server already has a value for EAIC code received in the previous email sent from the same sending mail user.

A brief illustration of EAIC codes to be used by Sending Mail Server (SMS) for given pair of Sending Mail User (SMU) and Receiving Mail Domain (RMD) is shown in *Table 1*. For simplicity, the Initial Random number is shown as 5 digits instead of 16 digits and the Random Increment Number is shown as 2 digits instead of 4 digits making an EAIC code of 5 digits instead of 16 digits in actual.

SMS	RMD	IRN	RIN	EAIC
prashant@gmail.com	msubaroda.ac.in	10000	23	10000
prashant@gmail.com	msubaroda.ac.in	10000	23	10023
prashant@gmail.com	yahoo.in	20000	12	20000
sandeep@gmail.com	msubaroda.ac.in	30000	15	30000
prashant@msubaroda.ac.in	gmail.com	12345	325	12345

Table 1: Illustration of EAIC codes calculated for various pairs of SMU and RMD

The process of calculation of EAIS score of the proposed Adaptive Authentication Framework (AAF) is as follows:

1. The EAIS score is initialized to 0 when a new email is received.
2. When the email server receives the second email having enclosed EAIC code, it finds out the difference between the current and the previous EAIC code for the given pair and finds out the random increment number which should be the same as the one stored at the sending mail server.
3. The Receiving Mail Server stores entries of latest EAIC code received along with the Random Increment Number found for each sending email user.
4. Now whenever a new email comes from the same sending mail user, the EAIC code received in the email is checked against the expected EAIC code calculated using the previous EAIC code received and the Random Increment Number. If both the EAIC codes are found to be equal, it is concluded that the received email is authentic, and 50 points are added to the EAIS, otherwise it is assumed to be spoofed.
5. After EAIC check, it is checked if there are multiple sender email ids present in the received email or not. If the received email came from only a single sender email id, 10 more points are added to the EAIS.
6. Then, SPF is checked and if it is passed, 10 points are added to the EAIS.
7. Next, DKIM is checked and if it is passed, 10 points are added to the EAIS.
8. Next, DMARC is checked and if it is passed, 20 points are added to the EAIS. More points are given to DMARC test than the SPF and DKIM as it checks the Identifier alignment which is one of the most critical tests for checking spoofed email.

Now the final EAIS score is tested against the predefined threshold value by the email server. If it is above the threshold, the email is concluded that the received email is authentic, otherwise it is assumed to be spoofed.

5 TESTING, RESULT AND PERFORMANCE ANALYSIS

5.1 Testing

For testing purposes, we selected the top 10, mostly used email servers around the world for our study to find out whether spoofing of email is still possible or not on these email servers. We also selected some domain names for the experiment based on their DNS entries of SPF, DKIM and DMARC. We then performed our experiment wherein we tried to think as an attacker and tried various techniques to bypass some or all of the anti-spoofing protocols.

We divided our testing experiment into various test cases as follows:

1. **Scenario-I:** Sending normal email from our own domain (mail-server.in) which has all three protocols SPF, DKIM & DMARC
2. **Scenario-II:** Spoofing a domain (gmail.com) which has all three protocols SPF, DKIM and DMARC
3. **Scenario-III:** Spoofing a domain (outlook.com) which has all three protocols SPF, DKIM and DMARC
4. **Scenario-IV:** Spoofing a domain (msubaroda.ac.in) having only DKIM record
5. **Scenario-V:** Spoofing a domain (orthocarehospital.in) having only SPF record
6. **Scenario-VI:** Spoofing a domain (orthocarehospital.in) having both SPF and DKIM records
7. **Scenario-VII:** Spoofing a domain (gujarattourism.com) having none of the authentication records

5.2 Result

From the experiment, we can conclude that our email server can successfully deliver email to the inbox of all major email service providers and the sent email passed all three authentication protocols successfully as shown in *Figure 16.*

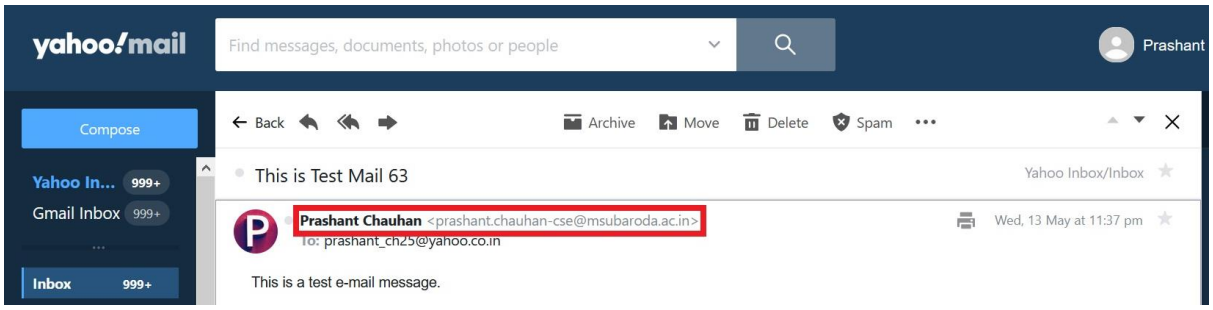

Figure 16: Spoofed email successfully delivered in inbox of recipient

Only in the case of outlook.com, our email was sent to spam instead of inbox despite passing all three protocols. The raw message showing result of SPF, DKIM and DMARC in case of spoofing of email with domain msubaroda.ac.in is shown in *Figure 17* and the result of spoofing of email with domain gmail.com is show in *Figure 18*.

```
 by atlas203.free.mail.sg3.yahoo.com with SMTPs; Mon, 29 Jun 2020 05:07:47 +000
X-Originating-Ip: [209.85.208.53]
Received-SPF: none (domain of msubaroda.ac.in does not designate permitted send
Authentication-Results: atlas203.free.mail.sg3.yahoo.com;
 dkim=pass header.i=@msubaroda-ac-in.20150623.gappssmtp.com header.s=20150623;
 spf=none smtp.mailfrom=msubaroda.ac.in;
 dmarc=unknown
X-Apparently-To: prashant_ch25@yahoo.co.in; Mon, 29 Jun 2020 05:07:48 +0000
X-YMailISG: f4hEagoWLDuOVSAnXuGuF4Eu0eUpyPebtCzTywRGWNHkU4O_
 uXDhZqvLPZlYsA2kGi_bQpBTaYcUDmd6Kr674KvyMkVK59lS08NJkgP_mks0
 tCvkT46gdr9BoFXq24Mk_JPLgKT.d_GqWVArNSYMbbmhDBX21Qa8VM70y15i
```

Figure 17: Raw email message for spoofed email for domain msubaroda.ac.in

```
Received: from 117.240.215.158 (EHLO mail.mail-server.in)
 by atlas116.free.mail.sg3.yahoo.com with SMTPs; Fri, 26 Jun
X-Originating-Ip: [117.240.215.158]
Received-SPF: pass (domain of mail-server.in designates 117.2
Authentication-Results: atlas116.free.mail.sg3.yahoo.com;
 dkim=pass header.i=@mail-server.in header.s=modoboa;
 spf=pass smtp.mailfrom=mail-server.in;
 dmarc=success(p=NONE,sp=QUARANTINE) header.from=gmail.com;
X-Apparently-To: prashant_ch25@yahoo.co.in; Fri, 26 Jun 2020
```

Figure 18: Raw email message for spoofed email for domain gmail.com

We also found that DMARC failed in all the cases due to identifier alignment issue, while both SPF and DKIM were passed by all the email servers used in the experiment for the spoofed email. Gmail and Zoho Mail shows “via” message along with the sender email id. Proton Mail and Rediffmail displays a warning message to the user along with the email and in most cases, the email is successfully delivered to inbox without any kind of warning to the user.

To further try to bypass the authentication protocols, we tried to send the spoofed email with multiple senders, the results of which are shown in *Figure 19*. As seen in the figure all the three protocols SPF, DKIM and DMARC passed for our spoofed email when testing was done on the domain of google.com.

Original Message

Message ID	<5fb76ad3.1c69fb81.cada0.355f28122020132115@mail-server.in>
Created at:	Mon, Dec 28, 2020 at 1:21 PM (Delivered after 7 seconds)
From:	"Prashant Chauhan M.S.U. <prashantchauhan25@gmail.com> The M.S. University of Baroda" <prashant@mail-server.in>
To:	prashant.chauhan-cse@msubaroda.ac.in
Subject:	A test message using python 57
SPF:	PASS with IP 117.240.215.158 Learn more
DKIM:	'PASS' with domain mail-server.in Learn more
DMARC:	'PASS' Learn more

Figure 19: Test results for spoofed email for domain gmail.com

The raw message displaying the headers of the received email including the EAIC code as one of the headers can be seen in *Figure 20*. The received EAIC code will then be used by our proposed Adaptive Authentication Framework (AAF). And depending on the value of EAIC code the proposed framework will detect whether the received email is authentic or spoofed.

```
Message-ID: <60e5f224.1c69fb81.f711b.5f15@mx.google.com>
Date: Wed, 07 Jul 2021 11:27:48 -0700 (PDT)
X-Google-Original-Date: 7 Jul 2021 23:57:48 +0530
EAIC: 4788149943119843
MIME-Version: 1.0
From: ams-cc@msubaroda.ac.in
To: prashantchauhan25@gmail.com
Subject: Test Mail - 15
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: quoted-printable

This is Test Mail.
```

Figure 20: Email message headers displaying EAIC code

The result of our experiment under the above-mentioned scenarios is displayed in *Table 2*. From the table we found that DMARC failed in all the cases except the scenario-I, which is our own email server, due to identifier alignment issue, while both SPF and DKIM were passed by all the email servers used in the experiment for the spoofed email.

Receiving Server Name	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI	Scenario VII
Gmail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Outlook	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Proton Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
AOL Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Yahoo! Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Zoho Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
iCloud Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Rediffmail	SPF, DKIM, DMARC	SPF	SPF	SPF	SPF, DKIM	SPF, DKIM	SPF
Mail.com	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM
Yandex Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM

Table 2: Result for Spoofing under various scenarios

The location of delivery of email in case of spoofing of email with single sender id is displayed *in Table 3*.

Receiving Server Name	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI	Scenario VII
Gmail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Outlook	Spam	Spam	Spam	Spam	Inbox	Inbox	Spam
Proton Mail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
AOL Mail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Yahoo! Mail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Zoho Mail	Inbox	Spam	Spam	Inbox	Inbox	Inbox	Inbox
iCloud Mail	Inbox	Inbox	Inbox	Spam	Inbox	Inbox	Inbox
Rediffmail .com	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Mail.com	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Yandex Mail	Inbox	Spam	Spam	Spam	Spam	Spam	Spam

Table 3: Location of email delivery under various scenarios

Receiving Server Name	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI	Scenario VII
Gmail	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC
Outlook	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Proton Mail	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM, DMARC	SPF, DKIM, DMARC
AOL Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Yahoo! Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM, DMARC	SPF, DKIM, DMARC
Zoho Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
iCloud Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM, DMARC	SPF, DKIM, DMARC	SPF, DKIM, DMARC
Rediffmail	SPF, DKIM, DMARC	SPF	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM
Mail.com	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered
Yandex Mail	SPF, DKIM, DMARC	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM	SPF, DKIM

Table 4: Result for Spoofing under various scenarios using multiple sender email ids

The result of the experiment for spoofed email with multiple sender email ids is displayed in *Table 4*. From this table, we can see that DMARC is now passed in many cases using multiple sender email ids which was not passed when spoofing using single sender email id. The location of delivery of email in case of spoofing of email with multiple sender ids is displayed in *Table 5*.

Receiving Server Name	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V	Scenario VI	Scenario VII
Gmail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Outlook	Spam	Spam	Spam	Spam	Inbox	Inbox	Inbox
Proton Mail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
AOL Mail	Inbox	Inbox	Inbox	Inbox	Spam	Spam	Spam
Yahoo! Mail	Inbox	Inbox	Inbox	Inbox	Spam	Inbox	Inbox
Zoho Mail	Inbox	Spam	Spam	Inbox	Inbox	Inbox	Inbox
iCloud Mail	Inbox	Inbox	Inbox	Spam	Spam	Inbox	Inbox
Rediffmail	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox	Inbox
Mail.com	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered	Not Delivered
Yandex Mail	Inbox	Spam	Spam	Inbox	Spam	Spam	Spam

Table 5: Location of email delivery under various scenarios using multiple email ids

5.3 Performance Analysis

From our testing, we found that our proposed Framework can detect an email to be spoofed in all of the types of attacks found and tested earlier, which was not possible using only the SPF, DKIM and DMARC protocols. Comparison of performance analysis of the proposed system with the existing system is shown in *Table 6*.

Sr. No.	Existing Method (SPF, DKIM, DMARC)	Adaptive Authentication Framework (AAF)
1.	Solution is at individual domain level	Solution is at email server level
2.	Successful only if all the domain users publish their SPF, DKIM and DMARC records	Can be successful by implementation of proposed model only by major email servers
3.	Even after successful implementation, spoofing is still possible till date	With proposed model rate of spoofing is expected to become negligible
4.	SPF, DKIM and DMARC can be bypassed by specific type of attacks	Proposed model is resistant towards such attacks

Table 6: Comparison of proposed framework with existing methods

6 CONCLUSION, LIMITATIONS AND FUTURE WORK

6.1 Conclusion

From our experiment, we found that our proposed protocol can detected an email to be spoofed in all the types of attacks found and tested earlier, which was not possible using only the SPF, DKIM and DMARC protocols. Comparison of our proposed framework with the existing methods used for prevention and detection of spoofing of email is shown is Table 1.

It can be said that identifier alignment used in DMARC is quite helpful in marking an email as DMARC fail. It checks if the email id in “From” field of the email body is same as the one in email header. If both these email ids are not similar, DMARC test is failed, and decision is made by the recipient email server based on the DMARC policy established by the sending email domain owner. It is found that even when DMARC test is failed, none of the email servers rejected our spoofed email because of their selected DMARC policy. The liberal DMARC policy of allowing email even in case of failure of DMARC, leads the email to the inbox of the user which might be dangerous for the recipient user.

Also, with our experiment we can say that using multiple email ids in the “From” field of email body, identifier alignment test can be passed, if one of the email ids in the “From” field of email body is kept same as the one in email header. There must be some techniques to ensure identifier alignment is checked and passed before any email is received at the receiving email server which will ensure prevention of email spoofing.

Also, it is found that some of the email service providers like Gmail and Zoho Mail show a “via” message along with the sender email id which is quite helpful in detecting the source of the email and warn the user for possibility of spoofing. And, in few cases of spoofing of email, only Proton Mail and Rediffmail displayed a warning message to the user to warn them of possibility of spoofing of email, while rest others displayed no warning to the user regarding possibility of spoofing. A normal user is not expected to go to email details and see the raw email message to check the status of SPF, DKIM and DMARC tests.

In such cases, where users are not very tech-savvy, they will only see whatever is displayed on their main screen of the inbox and won’t in detail to check the authenticity of the received email and may get in the trap of the attacker. In some cases, and by few of the email servers, this kind of warning message is displayed to the user but not by majority of them.

We suggest the email receivers to provide such warning message to the user in all cases where any of the tests are failed or in case spoofing of email is presumed by the receiving email server. The ideal threshold value used to check the authenticity of an email in Adaptive Authentication Framework (AAF) is kept to be 60 so an email can be assumed to be authentic even if it passes the EAIC test and if it came from a single sender email id, irrespective of the other checks.

Thus, our framework will work even if the authentication protocols like SPF, DKIM and DMARC are not used by any domain owner. Also, the threshold value can be adjusted by the email server as per their needs.

In present scenario the only options for preventing spoofing of email are authentication protocols namely SPF, DKIM and DMARC which are not sufficient to prevent email spoofing completely. We also found that there are multiple ways by which we may bypass all the three authentication protocols and make way for the spoofed email to the inbox of the recipient user instead of marking the email as spam or rejecting the email all together.

As a conclusion of this experimental study, we propose our Adaptive Protocol to be used by all the email servers to ensure prevention and detection of email spoofing in addition to the existing protocols like SPF, DKIM and DMARC.

6.2 Limitations

One major drawback of the proposed Framework may be that the system starts working only after receiving at least 2 emails from the same sending mail user. It may be possible that an attacker sends the first few emails and sets the EAIC code as per its need. In that case the emails will be bypassed from our proposed framework. But this won't be a major concern considering the fact that attackers normally spoof only the famous or well-known email ids and not a new email id. Thus, for any well-known email id the genuine EAIC code will exist at the receiving email server side which will not be possible for any attacker to bypass.

6.3 Future Work

The proposed system is well equipped to countermeasure the problem of authentication of email. To further increase the security of the system and to prevent attackers from finding a way to bypass the proposed system following enhancements could be made:

1. Provision of change of Initial Random Number and Random Increment Number used for generation of EAIC code after every 100^{th} or any other number of emails sent by a specific user to a specific email server domain.

2. At present, the EAIC code is sent as a header along with the email body. To further increase security, the receiving email server could remove the EAIC code after its purpose of authentication is served. So, no user can see the EAIC code used for the authentication of emails received by him or her.

7 REFERENCES

[1] Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten, Elie Bursztein, Nicolas Lidzborski, Kurt Thomas, Vijay Eranti, Michael Bailey and J. Alex Halderman, "Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security," in *Proceedings of the 2015 Internet Measurement Conference*, 2015.

[2] Zakhar Yung, "SPF Record Explained," [Online]. Available: https://mailtrap.io/blog/spf-records-explained.

[3] "DMARC Explained," [Online]. Available: https://mailtrap.io/blog/dmarc-explained/.

[4] Yuanyuan Grace Zeng, "Identifying email threats using predictive analysis," in *International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, 2017.

[5] Wissem Soussi, Maciej Korczynski, Sourena Maroofi and Andrzej Duda, "Feasibility of Large-Scale Vulnerability Notifications after GDPR," in *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2020.

[6] statista.com, "Number of sent and received e-mails per day worldwide from 2017 to 2026 (in billions)," [Online]. Available: https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/.

[7] Sourena Maroofi, Maciej Korczyński, Arnold Hölzel and Andrzej Duda, "Adoption of Email Anti-Spoofing Schemes: A Large Scale Analysis," in *IEEE Transactions on Network and Service Management*, 2021.

[8] Shuji Sakuraba, Minami Yoda, Yuichi Sei, Yasuyuki Tahara and Akihiko Ohsuga, "Sender Reputation Construction method using Sender Authentication," in *IEEE International Conference on Data Science and Computer Application (ICDSCA)*, 2021.

[9] Shuji Sakuraba, Minami Yoda, Yuichi Sei, Yasuyuki Tahara and Akihiko Ohsuga, "Improvement of Legitimate Mail Server Detection Method using Sender Authentication," in *IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2021.

[10] Sarah Scheffler, Sean Smith, Yossi Gilad and Sharon Goldberg , "The Unintended Consequences of Email Spam Prevention," in *International Conference on Passive and Active Network Measurement*, 2018.

[11] L. B. N. K. a. G. C. R. Holz, "The SSL landscape: A thorough analysis of the X.509 PKI using active and passive measurements," in *11th ACM Internet Measurement Conference*, 2011.

[12] Piotr Malek, "Everything You Need to Know About SMTP Security," [Online]. Available: https://mailtrap.io/blog/smtp-security/.

[13] Piotr Malek, "DKIM Explained," [Online]. Available: https://mailtrap.io/blog/dkim/.

[14] M. H. Jalalzai, W. B. Shahid and M. M. W. Iqbal, "DNS security challenges and best practices to deploy secure DNS with digital signatures," in *12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2015.

[15] S. A. D. B. a. C. J. L.-S. Huang, "An experimental study of TLS forward secrecy deployments," in *Web 2.0 Security and Privacy (W2SP)*, 2014.

[16] J. Klensin., "Simple mail transfer protocol," 2008. [Online]. Available: http://tools.ietf.org/html/rfc5321.

[17] S. Kitterman., "Sender policy framework (SPF) for authorizing use of domains in email," 2014. [Online]. Available: http://tools.ietf.org/html/rfc7208.

[18] Kenya Dan, Naoya Kitagawa, Shuji Sakuraba and Nariyoshi Yamai, "Spam Domain Detection Method Using Active DNS Data and E-Mail Reception Log," in *43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2019.

[19] Kaiwen Shen, Chuhan Wang, Minglei Guo, Xiaofeng Zheng, Chaoyi Lu, Baojun Liu, Yuxuan Zhao, Shuang Hao, Haixin Duan and Qingfeng Pan, "Weak Links in Authentication Chains: A Large-scale," in *30th USENIX Security Symposium*, 2021.

[20] Jeremy Clark, P.C. van Oorschot, Scott Ruoti, Kent Seamons and Daniel Zappala, "SoK: Securing Email -- A Stakeholder-Based Analysis (Extended Version)," arXiv, 2018.

[21] Ian D. Foster, Jon Larson, Max Masich, Alex C. Snoeren, Stefan Savage and Kirill Levchenko, "Security by Any Other Name: On the Effectiveness of Provider Based Email Security," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[22] I. Dolnák, "Secure mutual exchange of messages between network nodes inspired by security technologies for electronic mail exchange," in *19th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2021.

[23] Hongming Che, Qinyun Liu, Lin Zou, Hongji Yang, Dongdai Zhou and Feng Yu, "A Content-Based Phishing Email Detection Method," in *IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2017.

[24] J. Hoffman-Andrews, "Isps removing their customers' email encryption," [Online]. Available: https://www.eff.org/deeplinks/2014/11/starttls-downgrade-attacks.

[25] P. Hoffman., "SMTP service extension for secure SMTP over transport layer security," Feb. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3207.txt.

[26] Hang Hu, Peng Peng and Gang Wang, "Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems," in *IEEE Cybersecurity Development (SecDev)*, 2018.

[27] Hang Hu and Gang Wang, "End-to-End Measurements of Email Spoofing Attacks," in *27th USENIX Security Symposium*, 2018.

[28] Geethapriya Liyanage and Shantha Fernando, "A comprehensive secure email transfer model," in *IEEE International Conference on Industrial and Information Systems (ICIIS)*

[29] Frank Li, Zakir Durumeric, Jakub Czyz, Mohammad Karami and Michael Bailey, "You've Got Vulnerability: Exploring Effective Vulnerability Notifications," in *25th USENIX Security Symposium*, 2016.

[30] J. H. A. a. P. Eckersley, "STARTTLS everywhere," June 2014. [Online]. Available: https://github.com/EFForg/starttls-everywhere.

[31] Bridget Opazo, Don Whitteker and Chen-Chi Shing, "Email Trouble: Secrets of Spoofing, the Dangers of Social Engineering, and How We Can Help," in *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2017.

[32] Asif Karim, Sami Azam, Bharanidharan Shanmugam, Krishnan Kannoorpatti and Mamoun Alazab, "A Comprehensive Survey for Intelligent Spam Email Detection," in *IEEE Access*, 2019.

[33] "Spoofing - Who Did That Email Really Come From?," [Online]. Available: https://www.consumerfraudreporting.org/spoofing.php.

[34] "Simple Mail Transfer Protocol," [Online]. Available: http://www.ietf.org/rfc/rfc2821.txt.

[35] "Modoboa: Open Source email server," [Online]. Available: https://modoboa.org/en/.

[36] "Email Spoofing," [Online]. Available: http://en.wikipedia.org/wiki/E-mail_spoofing.

[37] "Domain-based message authentication, reporting, and conformance (DMARC)," 2015. [Online]. Available: https://tools.ietf.org/html/rfc7489.

[38] "DMARC Overview," [Online]. Available: https://dmarc.org/overview/.

[39] "DKIM FAQs," [Online]. Available: https://dkim.org/info/dkim-faq.html.

[40] "25 Common Mistakes - Email Security," [Online]. Available: http://www.windowsecurity.com/whitepapers/25-Common-Mistakes-Email-Security.html.

8 RESEARCH PUBLICATIONS

The research publications done during the period of research for Ph.D. work is as follows:

1. "Effectiveness of Anti-Spoofing Protocols for Email Authentication", IEEE 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT'23), Manipal University Jaipur, January 19-20, 2023
2. "Email Spoofing: In Today's Era", Journal of Harbin Engineering University, Vol 44 No. 7, July 2023, ISSN Number: 1006-7043 [Scopus Indexed]
3. "Adaptive Protocol for Detection and Prevention of Email Spoofing", IEEE International Conference on Multidisciplinary Research in Technology and Management (MRTM), New Horizon College of Engineering, Bengaluru, Karnataka, India, 22-23, September 2023
4. "Adaptive Authentication Framework for Prevention of Email Spoofing", International Journal of Information Security, Springer Nature, ISSN Number: 16155270, 16155262 [Submission Under Peer Review Since: 07 July 2023]