Chapter – 4
Proposed System Architecture and Methodology

4.1 Introduction

With the literature review and our various experiments, we are now well aware that spoofing is still easily possible. We also know how the attackers spoof an email to bypass the existing anti-spoofing protocols like SPF, DKIM and DMARC. With this knowledge, we are now in a position to try to find techniques to prevent the spoofing of email. For the possible ways to prevent spoofing of email we can think of the following two basic approaches:

- **Domain Level Approach**

  All the existing anti-spoofing protocols currently in use like SPF, DKIM and DMARC work at individual domain level. This means that for these anti-spoofing protocols to work, the domain owner needs to publish its respective records in its own DNS. The existing protocols will prevent or detect spoofing of emails only for those domains whose owners have published SPF, DKIM and DMARC records. And those domains which are not explicitly using these protocols become easy targets for the attackers for spoofing of email addresses from their domains. Also, the current protocols working on domain level, don't protect an individual domain from getting spoofed emails just because they have published SPF, DKIM and DMARC records in their DNS. This means that for example, if the domain owner of abc.com is using all the three anti-spoofing protocols currently available, still any user of abc.com can receive a spoofed email from a domain which is not using these protocols. We can say, in a way, the current anti-spoofing protocols will prevent spoofing of emails only if all the domain owners of all the existing domains start using these protocols which is seemingly quite impossible.

- **Email Server Level Approach**

  For our research we try to find solutions to the problem of spoofing of email using email server level approach so that of if all the email servers adopt our solution the whole email system will get benefit from it as compared to the domain specific approach wherein something is needed to be done by all the domain owners. We wish to find a solution to the problem of spoofing of emails which can be easily adopted and doesn't require involvement of individual domain owners. None of the researchers have given successful measures to prevent email spoofing using this type of approach until now.

4.2 Proposed Framework

To overcome the issue of spoofing of emails we propose a new Adaptive Authentication Framework for Email (AAFE) which will work in addition to the existing protocols for authentication of email. Our proposed framework will take advantage of the benefits of the existing anti-spoofing protocols currently in use like SPF, DKIM and DMARC and try to give a complete solution for prevention of email spoofing. Although we use existing protocols in our proposed framework, they are not necessary for the working of our proposed framework. This means that even if a domain is not using existing anti-spoofing protocols, the domain users can still get benefit of prevention and detection of spoofing of email if their email service provider uses our proposed framework.

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).

Basic terminologies used in our proposed framework is described in *Table 2* and the flow chart of the process of email is shown in *Figure 64*.

Terminology	Meaning
Sending Mail Server (SMS)	This represents the email server which is used for sending an email.
Receiving Mail Server (RMS)	This represents the email server which will be used for receiving an email.
Sending Mail User (SMU)	This represents a specific user who is going to send an email to the recipient user.
Receiving Mail User (RMU)	This represents a specific user who is going to receive an email from the Sending Mail User (SMU)

Table 2: Basic terminologies used in proposed system

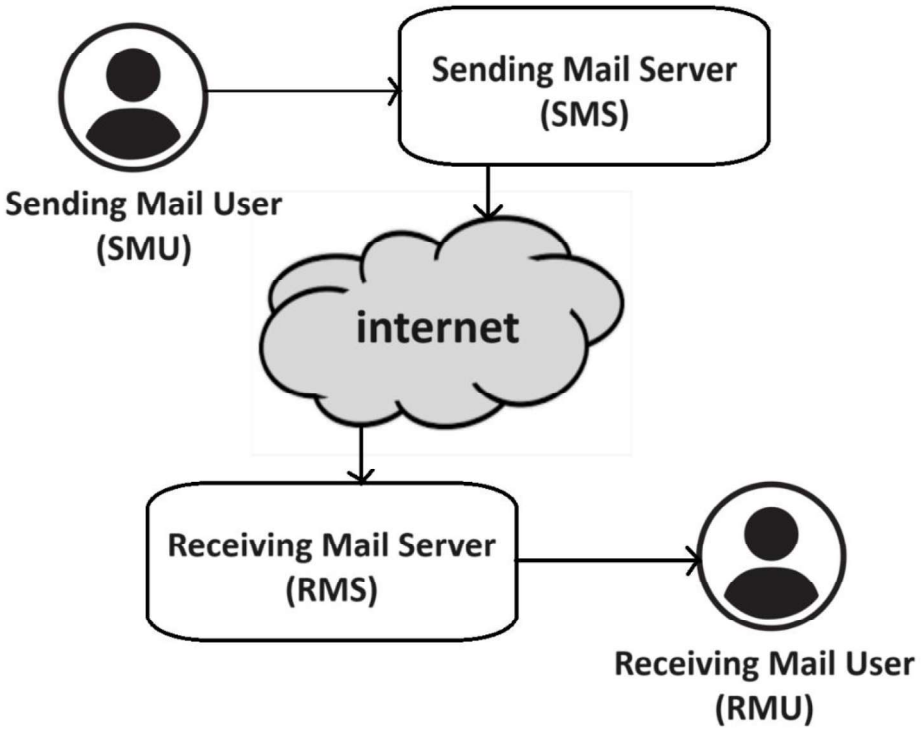

Figure 64: Basic flow chart of various components involved in email process

4.2.1 Email Authentication Identity Code (EAIC)

The Email Authentication Identity Code (EAIC) consists of following two components:

1. Initial Random Number (IRN) – 16 digits
2. Random Increment Number (RIN) – 4 digits

The process of calculation of Email Authentication Identity Code (EAIC) using Random Increment Number (RIN) and Initial Random Number (IRN) can be seen in *Figure 65*.

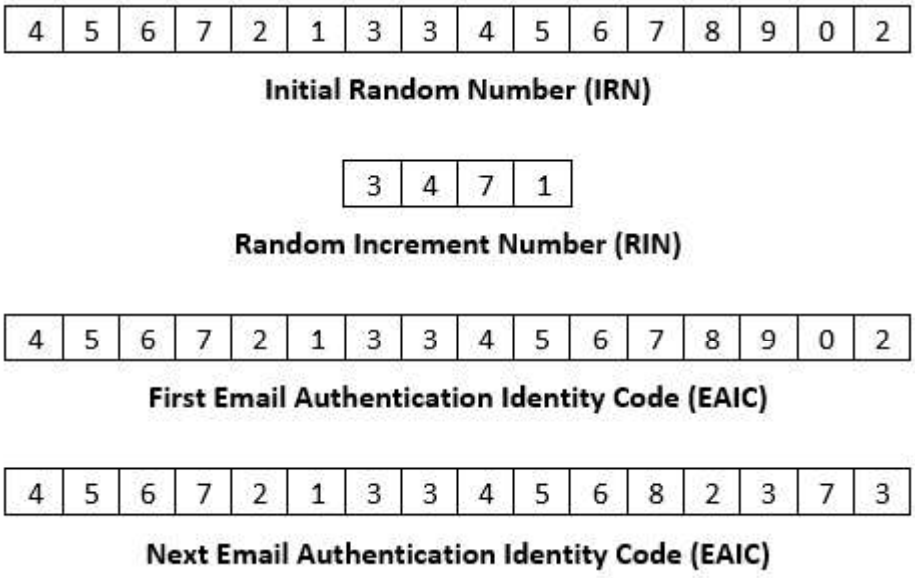

Figure 65: Process of calculation of EAIC

As shown in the figure, for the first time, the IRN is directly used as the EAIC. For consequent requests, the new EAIC is calculated by adding the Random Increment Number (RIN) to the last EAIC used. In this way the calculation of new EAIC takes place until the EAIC gets more than 16 digits. The process of calculation of EAIC and the new RIN can be seen in *Figure 66*.

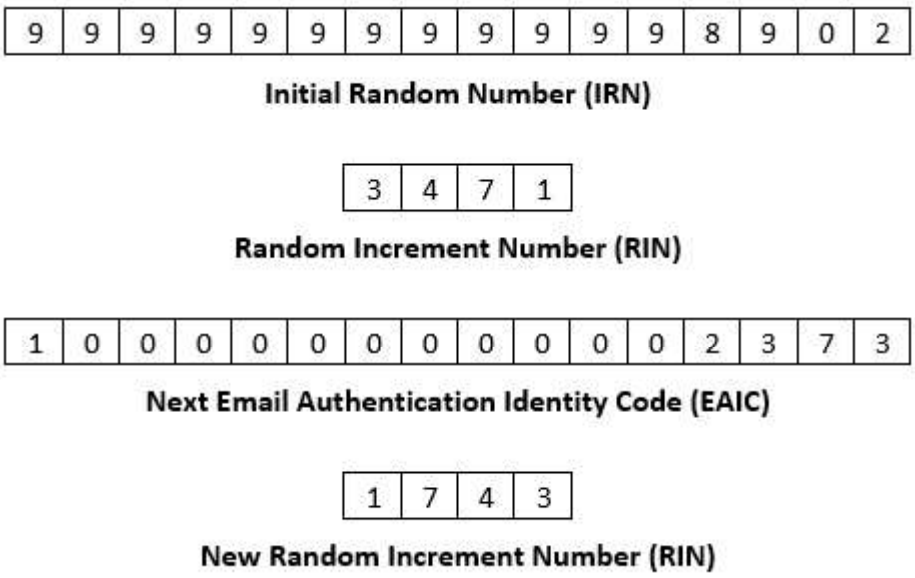

Figure 66: Process of calculation of EAIC in case EAIC gets more than 16 digits

As seen in the figure whenever by adding the RIN to last EAIC, if the new EAIC gets larger than 16 digits, one of the '0' from the EAIC code is truncated resulting in getting the EAIC code again of 16 digits. Along with this process the current value of RIN is also reversed to generate the new RIN.

4.2.2 Email Authentication Identity Score (EAIS)

A new term Email Authentication Identity Score (EAIS) is introduced to find the score of possibility of spoofing of an email. The benefit of existing anti-spoofing protocols like SPF, DKIM and DMARC is also considered in this score so as to add extra confidence to the Receiving Mail Server (RMS) in deciding the authenticity of any email.

Though our proposed framework considers the existing protocols it is not necessary to use them as our framework will work perfectly well even if no one is using currently available anti-spoofing protocols like SPF, DKIM and DMARC.

The process of calculation of EAIS is as following:

1. The EAIS score is initialized to 0 when a new email is received.
2. Whenever a new email comes from the same Sending Mail User (SMU), 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 already available with the Receiving Mail Server (RMS). 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 and EAIS score remains 0 as initialized.
3. 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.
4. Then, SPF is checked and if it is passed, 10 points are added to the EAIS.
5. Next, DKIM is checked and if it is passed, 10 points are added to the EAIS.
6. 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 very critical for checking spoofed email.

The flowchart of steps used in calculation of EAIS score at the Receiving Mail Server (RMS) side is shown in *Figure 67*.

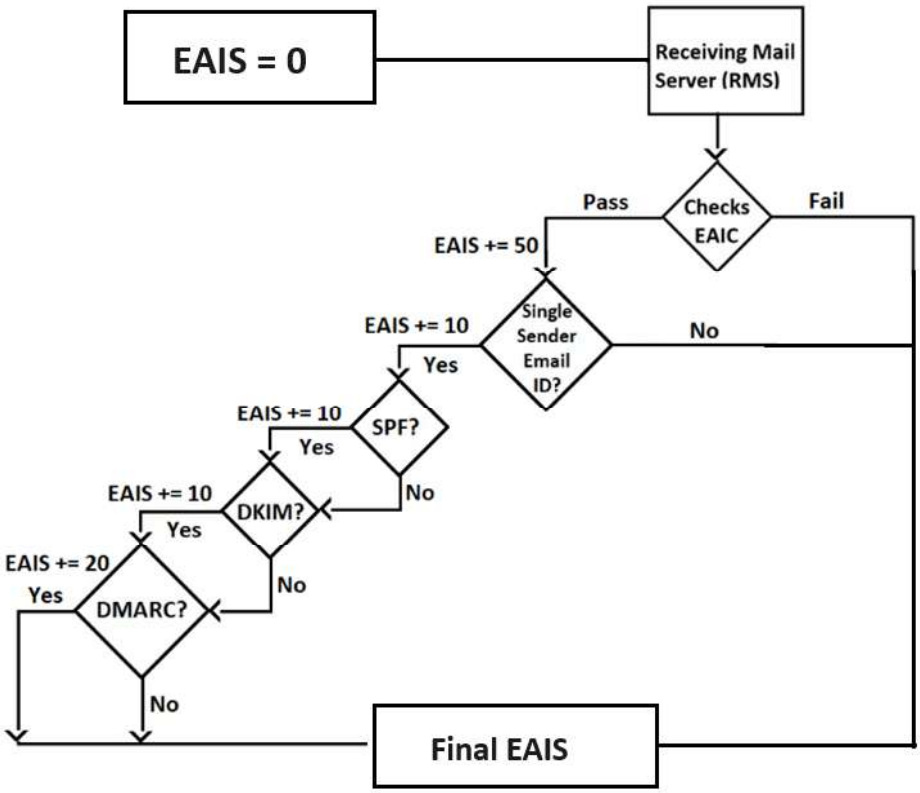

Figure 67: Flowchart for calculation of EAIS score

It is to be noted that the highest points are given to the EAIC check that is the core of our proposed framework. Apart from that, in case the received email is found to have multiple email addresses in the "From" field of email header, the other tests are not performed which check SPF, DKIM and DMARC. In this case, it is left in the hands of the receiving email server whether to allow such emails or not. A proper threshold value for EAIS can be decided by the RMS as per their policy. According to us, a minimum threshold value for EAIS must be kept as 50. This will mean that any email having EAIS score of 50 or more can be considered as authentic.

Different email servers have different requirements and different sets of security levels to be adopted, therefore what threshold value to be used can be decided by the respective email server only. Those email servers intending to have a high rate of precision may keep a higher value of EAIS and those wishing to keep a moderate level of security may configure a lower value of EAIS for their server.

4.2.3 Adaptive Authentication Framework for Email (AAFE)

A unique EAIC code is generated per pair of Sending Mail User (SMU) and the Receiving Mail Server (RMS). 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 (IRN) is generated as 8400580227532596 and a Random Increment Number (RIN) 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 i.e. the same IRN generated for the given pair. 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 (IRN) and Random Increment Number (RIN) will be generated which will be used for generating EAIC code for all communications between these 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 (SMS) and Receiving Mail Server (RMS). In both cases a new EAIC code is generated and sent along with the email to the Receiving Mail Server (RMS) which will deliver the email to the Receiving Mail User (RMU).

A sample table having brief illustration of EAIC codes to be used by Sending Mail Server (SMS) for given pair of Sending Mail User (SMU) and Receiving Mail Server (RMS) is shown in *Table 3*. Also note that 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.

Sr. No.	SMU	RMS	IRN	RIN	EAIC
1.	prashant@gmail.com	msubaroda.ac.in	10000	23	10000
2.	prashant@gmail.com	msubaroda.ac.in	10000	23	10023
3.	prashant@gmail.com	yahoo.in	20000	12	20000
4.	sandeep@gmail.com	msubaroda.ac.in	30000	15	30000
5	prashant@msubaroda.ac.in	gmail.com	12345	325	12345

Table 3: Illustration of EAIC codes calculated for various pairs of SMU and RMS

As seen from the table, for the first email from "prashant@gmail.com" any email id of domain "msubaroda.ac.in", IRN is 10000, RIN is 23 and EAIC is same as IRN i.e. 10000. Then for the next email from same SMU to RMS, the new EAIC is calculated by addition of last sent EAIC and RIN i.e. New EAIC = 10000 + 23 = 10023. Also, if the same user "prashant@gmail.com" sends another main to an email address of some other domain "yahoo.in", a separate pair of IRN and RIN is generated and stored at the server. These values will be used to calculate the EAIC code to be sent along with the email for this pair of SMU-RMS.

In this way a value of IRN, RIN and Last EAIC code is stored for each pair of Sending Mail User (SMU) and Receiving Mail Server (RMS) at the Sending Mail Server (SMS) and this value of EAIC is sent as email header along with each email to the Receiving Mail Server (RMS) as seen in *Figure 68*. Now the final EAIS score is tested against the predefined threshold value by the Receiving Mail Server (RMS). If it is above the threshold, it is concluded that the received email is authentic, otherwise it is assumed to be 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 68: Email header details showing EAIC code received by RMS

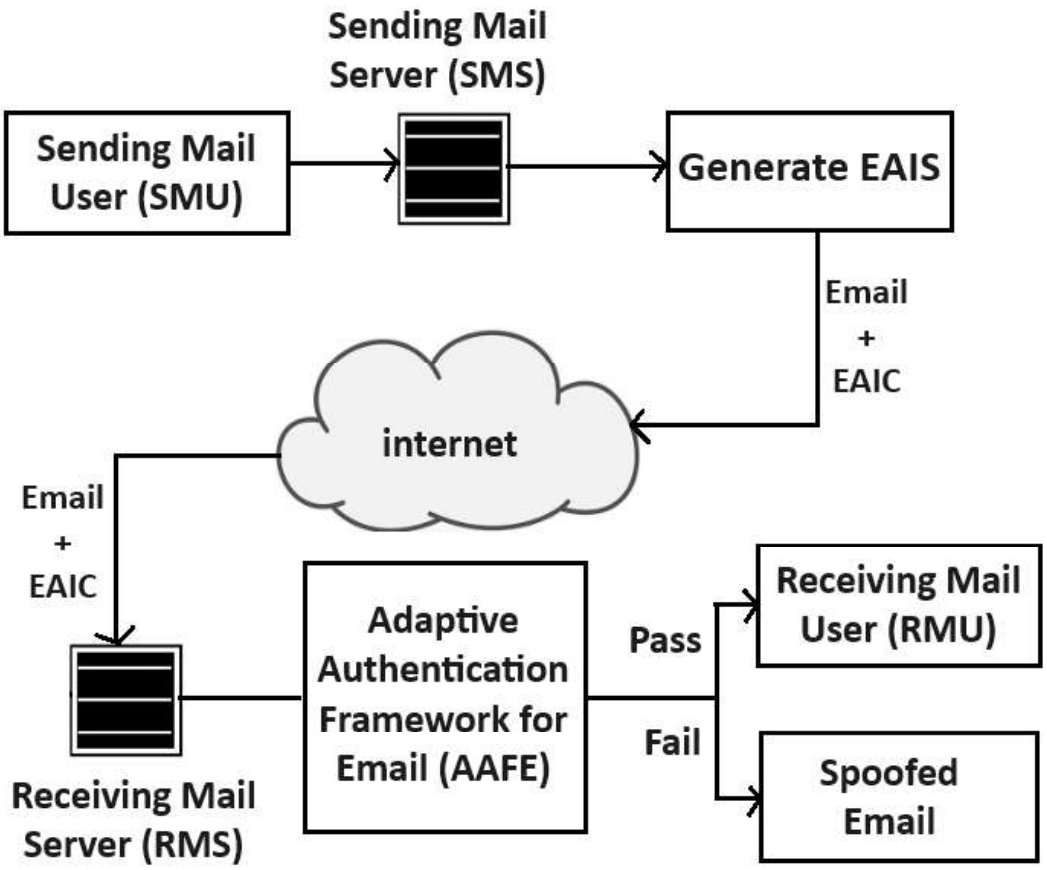

Figure 69: Flowchart for Adaptive Authentication Framework for Email (AAFE)

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.

It is to be noted that for better security purposes we have selected 16 digits for the IRN. And we stored the IRN and RIN only for the pair of SMU – RMS and not SMU – RMU. This means that for example, if we have stored IRN and RIN for a pair abc@msubaroda.ac.in – gmail.com, then these same numbers will be used for any email sent from abc@msubaroda.ac.in to any user of gmail.com domain. This is done to decrease the number of pairs for storage of necessary information. The other option would be to store these details for each pair of Sending Mail User (SMU) & Receiving Mail User (RMU), but this would unnecessarily increase storage burden on the email server using our proposed system.

For testing purpose, we created three separate applications representing three separate email servers namely Server – A representing domain “yahoo.com”, Server – B representing domain “gmail.com” and Server – C representing domain “msubaroda.ac.in” as seen in *Figure 70, Figure 71* and *Figure 72*. At the backend all the three applications use our own email server setup for the domain “mail-server.in”.

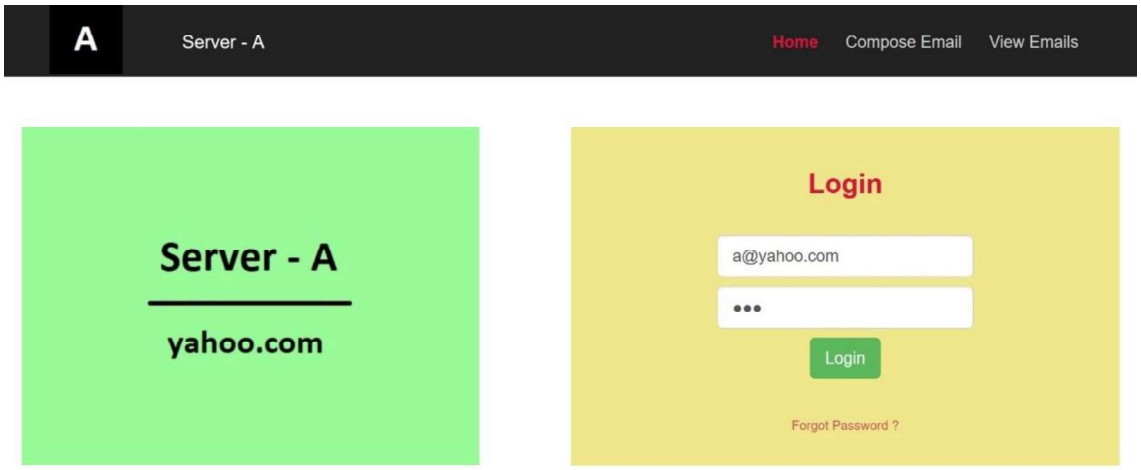

Figure 70: Application server representing domain "yahoo.com"

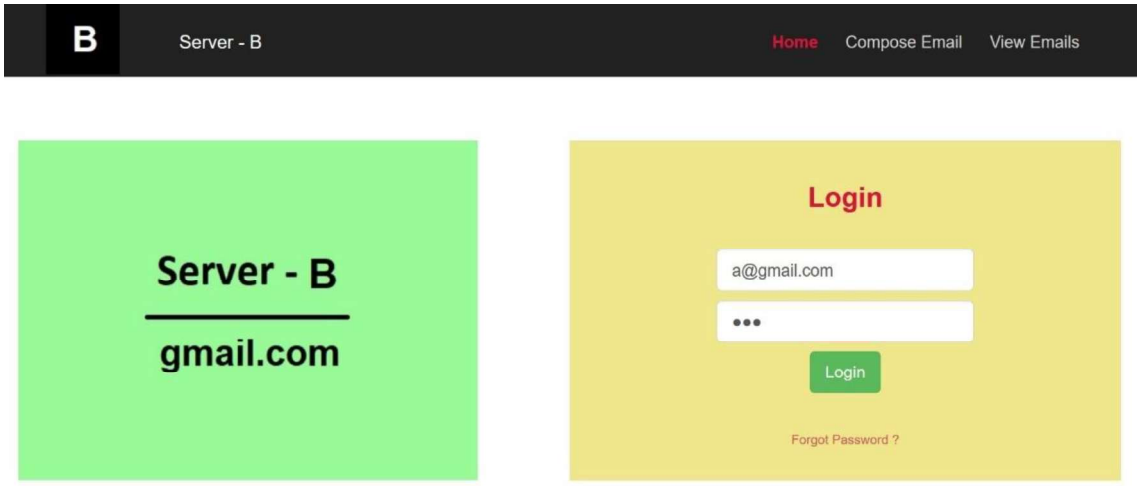

Figure 71: Application server representing domain "gmail.com"

We performed the testing of our proposed Adaptive Authentication Framework for Email (AAFE) using these three application servers. As Sending Mail Server (SMS), our application server sends an email from specific Sending Mail User (SMU) to Receiving Mail Server (RMS). The SMS generates the EAIC code using the specified algorithm and stores it in its own database for further processing and use.

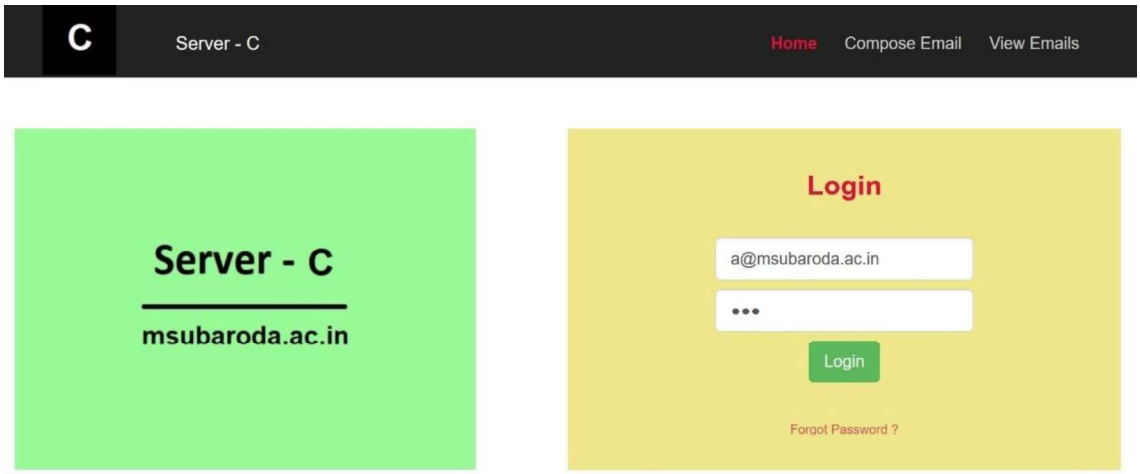

Figure 72: Application server representing domain "msubaroda.ac.in"

After generating the EAIC code for the specific pair of SMU-RMS, email is sent to RMS along with the EAIC code. When the Receiving Mail Server (RMS) receives an email, it extracts the EAIC code from the email header and passes it to our proposed Adaptive Authentication Framework for Email (AAFE). It performs step-by-step tests and calculates the EAIS score. At the end if the EAIS score is found to be above the prefixed threshold value setup by the Receiving Mail Server (RMS), the email is successfully delivered with confidence of authenticity of the sender to the Received Mail User (RMU) who is the intended recipient of the email. And in case the EAIS score fails to pass the threshold value, the email server may reject the email as it fails to pass the standards set for authentication of email.

4.3 Illustration of AAFE at Sending Mail Server Side

The process of sending email from Sending Mail User (SMU) to Receiving Mail User (RMU) is as follows:

1. If this is the first time the Sending Mail User (SMU) is sending an email to the Receiving Mail Server (RMS), a new value for 16-digit Initial Random Number (IRN) is generated for the pair (SMU - RMS) and stored at the Sending Mail Server (SMS).
2. Also, a Random Increment Number (RIN) is generated and stored at the Sending Mail Server (SMS) for the same pair.

3. If this is the first time the Sending Mail User (SMU) is sending an email to the Receiving Mail Server (RMS), the Initial Random Number (IRN) is considered as the EAIC code,
4. If the sending mail user has already sent an email to the receiving mail server, an EAIC code will already exist with the Sending Mail Server (SMS) which was used in the last email sent for the same pair of SMU - RMS. In this case, the new EAIC code will be an addition of the previous EAIC code, and the Random Increment Number (RIN) stored for the same pair with the Sending Main Server (SMS).
5. Once an EAIC code is generated for the given pair, email is sent to the Receiving Mail Server (RMS) from the Sending Mail User (SMU) along with the EAIC code attached as header to the email body.
6. Also, the current EAIC code sent along with the email is stored with the Sending Mail Server (SMS) for the current pair of SMU - RMS

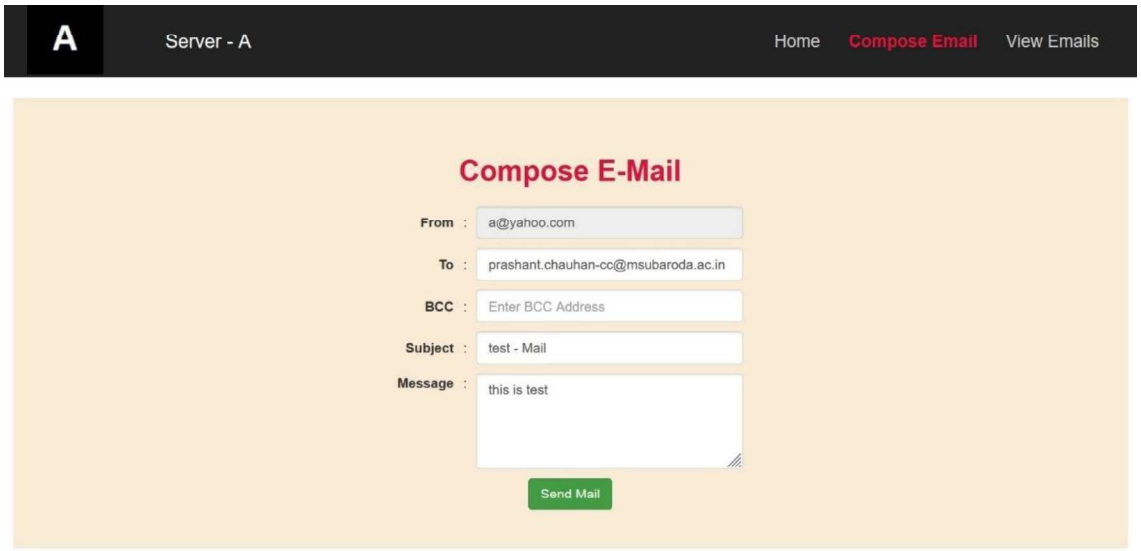

Figure 73: Sending email using our application server

As seen in *Figure 73*, we used one of our application servers for sending an email from "a@yahoo.com" to "prashant.chauhan-cc@msubaroda.ac.in" using compose e-mail page of our application. At the time of sending of the email a new EAIC code is generated and stored in the database along with IRN and RIN. Then, actual email is sent along with the EAIC code generated for the given pair to the Receiving Mail Server (RMS).

4.4 Illustration of AAFE at Receiving Mail Server Side

The process of receiving of email and calculation of EAIS score by the Receiving Mail Server (RMS) for the proposed Adaptive Authentication Framework for Email (AAFE) is as follows:

1. The EAIS score is initialized to 0 when a new email is received.
2. When the Receiving Mail Server (RMS) receives the first email from any Sending Mail Server (SMS) along with EAIC code, it simply stores the following values in its database for that specific SMU:
    - SendingUsername = SMU's email address
    - LastEAIC = EAIC code received in current email
    - RandomIncrementNumber = Initialized to 0
    - Count = 1 (being the first email from this specific user)
3. When the Receiving Mail Server (RMS) 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 in this way finds out the Random Increment Number (RIN) which should be the same as the one stored at the Sending Mail Server (SMS) for the given pair of SMU-RMS.
4. The Receiving Mail Server (RMS) stores entries of latest EAIC code received along with the Random Increment Number found for each Sending Mail User (SMU).
5. Now, whenever a new email comes from the same Sending Mail User (SMU), 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.
6. 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.
7. Then, SPF is checked and if it is passed, 10 points are added to the EAIS.
8. Next, DKIM is checked and if it is passed, 10 points are added to the EAIS.
9. Next, DMARC is checked and if it is passed, 20 points are added to the EAIS.

10. Final EAIS score is tested against the predefined threshold value. If it passes the threshold, the email is authentic and delivered to the respective RMU otherwise it may be rejected. But the decision of what to do in case of EAIS score less than the threshold value set by the email server remains with the RMS itself.

As seen in *Figure 74*, the email sent from "a@yahoo.com" (SMU) using our application server Server-A (SMS) along with the system generated EAIC code to "prashant.chauhan@msubaroda.ac.in" (RMU) hosted on our application server Server-C (RMS) successfully passes through our proposed Adaptive Authentication Framework for Email (AAFE) and may be delivered to the inbox of the recipient user with confidence of authenticity.

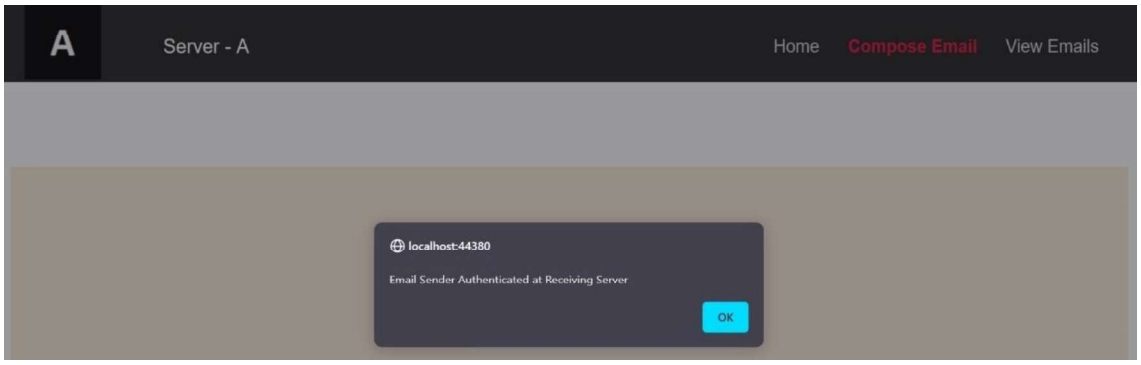

Figure 74: Email successfully authenticated at RMS using our proposed AAFE

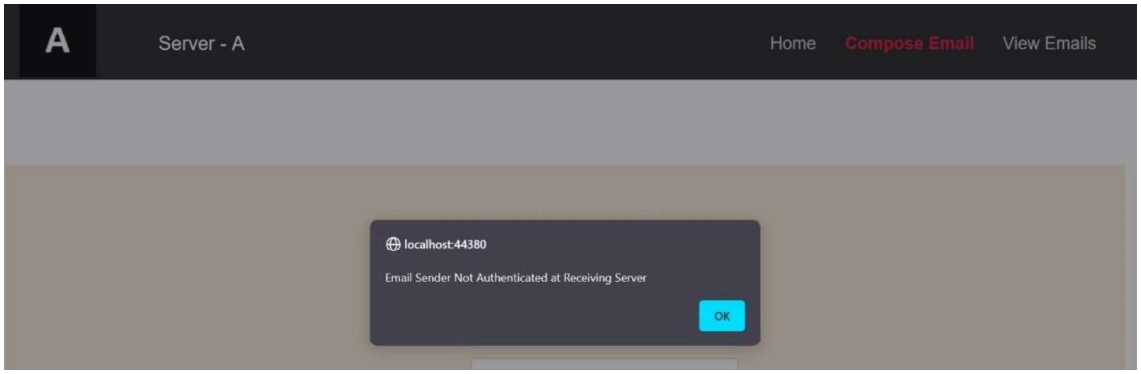

Figure 75: Email not authenticated at RMS using our proposed AAFE

In similar manner when we tried to send email from a@yahoo.com" having a fake EAIC code to "prashant.chauhan@msubaroda.ac.in", the EAIS score calculated based on the received EAIC code, was not able to pass the threshold value (60 in this case) and ultimately rejected by the RMS as seen in *Figure 75*.

4.5 Adaptive Authentication Framework for Email (AAFE) Security

Our proposed framework is expected to work more perfectly and give exceptional results as compared to the existing anti-spoofing protocols. Few of the advantages of our proposed framework from security perspective are as follows:

1. Random numbers of 16 digits are used so as to increase security and at the same time to make it difficult to guess it by any attacker.
2. The proposed framework adds extra security provisions for checking authentication of sender user along with the existing mechanisms. Therefore, the benefit of existing mechanism is not at all lost. Our proposed framework will always perform better than the existing protocols.
3. Man-in-the-middle attack, wherein an attacker somehow convinces the person whose email address is to be spoofed to send an email to his email server, modifies the received email to get advantage and forwards it to the victim user as a spoofed email from the original sender but with the modified content of the email, will not be possible using our proposed framework as the EAIC code is always separate for separate pair of SMU-RMS and the attacker will not be able to find out the EAIC code in any way.
4. On the Receiving Mail Server (RMS) side a count is kept which is increased by 1 for each email received from a specific SMU. For even better security and for making it more difficult for the attacker to break our proposed framework, we may devise a mechanism which changes the Random Increment Number (RIN) stored at both the the locations SMS and RMS. For example, after reaching count to 100, the SMS will add the RIN to itself and round off to 4 digits only resulting in a new value for the RIN at SMS. The RMS also follows the same process to calculate the new RIN stored with the RMS for the given pair of SMU-RMS.
5. To further add randomness to the EAIC code and RIN, the proposed framework truncates a “0” from EAIC code generated at SMS, whenever the value of EAIC code gets more than 16 digits and the RIN is reversed. This same process is followed at the RMS side also to update the RIN and the expected EAIC code.

4.6 Conclusion and Summary

In this chapter we explained our proposed framework i.e. Adaptive Authentication Framework for Email (AAFE), various terminologies used in this framework and the basic working of the proposed framework. We also set up three different application servers to demonstrate and test the working of our proposed framework. With the experiment we were able to successfully authenticate an email based on the EAIS score calculated using the EAIC code passed along with each email for a given pair of Sending Mail User (SMU) and Receiving Mail Server (RMS). We briefly explained the steps to be followed by the Sending Main Server (SMS) and the Receiving Mail Server (RMS) to authenticate the sender of an email using our proposed framework. We also discussed about the security perspectives of our proposed framework and can conclude that our proposed framework is expected to check the authenticity of an email effectively and give better performance as compared to the existing protocols using SPF, DKIM and DMARC.