**Documentation**

**Phase 1: Investigate E-mail Spoofing**

**Header for email 1 (Shefali Sharma shefali92@gmail.com To sharma92@buffalo.edu)**

Delivered-To: sharma92@g-mail.buffalo.edu

Received: by 10.74.189.150 with SMTP id k22csp133846oop;

Wed, 29 Nov 2017 19:12:33 -0800 (PST)

X-Google-Smtp-Source: AGs4zMZVxxLxOLzBUMyCCaTAGiKvJFJ81Nr7F4u0dhiACm/hMEKyfXoHe9Ho/u7oY9slkNp9sZiW

X-Received: by 10.55.170.130 with SMTP id t124mr858354qke.84.1512011553798;

Wed, 29 Nov 2017 19:12:33 -0800 (PST)

ARC-Seal: i=1; a=rsa-sha256; t=1512011553; cv=none;

d=google.com; s=arc-20160816;

b=YfNiowCkU3QImR165DxTWfhxNkaDcfNoMCb6eASQJXWnQI0aafnN2ZZIyfKu9YMwHm

9LcGlplQYNi2hH2klhgzs8iRct344k1vn/omFKwIVPlIDZ7pPRq2DBt8ijrQ73541uQN

71ljxXaN6NQzQZUJSXPZGsEmKxXi4z5e8bansizfbsxcznvhUjIf27lvait/ld3AIgjh

J35zLEWroK9LzTfLOC7Y3I5v8d6ftUZ1pMCufrkoU+XGl+W66AaXG9BmM3M8zUC8tBQS

deK5IpzvN6U5I7GBbeH/kopNeg+mIjA2Wyh8+Zh/japU0aMI4rn3s+aTWN37upF5mIHE

t0vg==

ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;

h=to:subject:message-id:date:from:mime-version:dkim-signature

:arc-authentication-results;

bh=a7+jBH/vFqChO8eE114IISzfaYaECICs177aCG5sIW0=;

b=0N/rubJMda0NLxZ60goNntxVq3G79eL4Cg8AUwx539nXd1bAnYVHcvl3NRhchjB75M

r9ez2U7OcYcsqpidTvET/5die9bleh8hI1BHsDhL75Kelrnu6RkHNFipE3fvFVYYeTLR

kALYKBH4RjZ6puGJim/lLQ686Tye54k0veCPaZpyFBjaqyrIBVAyHTL0F+1E949kFzSI

zDe/2ykIPN4/luZN6gxLS5ZhMLxd+uyLh4aU62qd1KNPOHQmPzwalk6YDQFz2rM32VGT

QnKfgiFyKqJl3q+gRcGl0m0w6IXwaGlWrMjTQyy5r4UxWK3s0KRiFep8GAST7G0HmD3I

+/1A==

ARC-Authentication-Results: i=1; mx.google.com;

dkim=pass header.i=@gmail.com header.s=20161025 header.b=OlOh2o8Z;

spf=pass (google.com: domain of shefali92@gmail.com designates 209.85.215.54 as permitted sender) mailfrom=shefali92@gmail.com

Return-Path: <shefali92@gmail.com>

Received: from mx1.mail.buffalo.edu (mx1.mail.buffalo.edu. [128.205.1.214])

by mx.google.com with ESMTP id t18si3319609qtb.105.2017.11.29.19.12.33

for <sharma92@g-mail.buffalo.edu>;

Wed, 29 Nov 2017 19:12:33 -0800 (PST)

Received-SPF: pass (google.com: domain of shefali92@gmail.com designates 209.85.215.54 as permitted sender) client-ip=209.85.215.54;

Authentication-Results: mx.google.com;

dkim=pass header.i=@gmail.com header.s=20161025 header.b=OlOh2o8Z;

spf=pass (google.com: domain of shefali92@gmail.com designates 209.85.215.54 as permitted sender) smtp.mailfrom=shefali92@gmail.com

Received: from mail-lf0-f54.google.com (mail-lf0-f54.google.com [209.85.215.54]) by mx1.mail.buffalo.edu (mx) with ESMTP id 5023E1004CF for <sharma92@buffalo.edu>; Wed, 29 Nov 2017 22:12:33 -0500 (EST)

Received: by mail-lf0-f54.google.com with SMTP id 94so6251310lfy.10

for <sharma92@buffalo.edu>; Wed, 29 Nov 2017 19:12:33 -0800 (PST)

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;

d=gmail.com; s=20161025;

h=mime-version:from:date:message-id:subject:to;

bh=a7+jBH/vFqChO8eE114IISzfaYaECICs177aCG5sIW0=;

b=OlOh2o8ZQFAehLt0ILB5Pzm/0ZB4zl2XWzQtgFBpa+9s/bEwqH6LlzLSDYpUs8hTb6

2BJB8R+9wkou0dLcnOohxBTCviblD3/gE9h48ZXwny5HPSunvjh/WL10Hw3BUDg+wn7J

Mw4vCd+29hJRf0q9oWph96r7nQ0mcjCaQ6L/oIzhyKW77MlMhJU2CKJ7Re7g7esbFFeG

vE9zuXOUzqxqt2yh4bxI+fAckpnndMvTCPrAvcLNhSMwD561kt2tNpaGi8pQ9Bj4+vQE

aK02nGbs89UiN3yBHInAOR3FxPmoMH76llu9+D0GyvcC7th3UjHjx/MgZ6VT0bUaVbkd

+QtA==

X-Google-DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;

d=1e100.net; s=20161025;

h=x-gm-message-state:mime-version:from:date:message-id:subject:to;

bh=a7+jBH/vFqChO8eE114IISzfaYaECICs177aCG5sIW0=;

b=MGTBhMvqvfkoQXSKHmLpsbarcEMOBY320DnXO1ntWI8eGdLrAdjfKxH85DVhDeNJ9y

0HhXTSdpykrJQ2NWj0Y0BelE2TmJ3olIHxt7V45MNswPIZPy4kgajf/nmlaY52mj7n7e

W026LE+nTjcvATJQjfVoKolE/yPeP09A9aaG1PzQx1egDENfTseZsDuwRiRrLqTqEPZU

zjloU/IT1OnGdWSC61XkeWiCB+7tM1HsT4Pb0pN8gsjDyk+f/I57jxpX16C8qEkjzR3n

VGYrM2MJtcclpafs4oFDoGuEyHRmPm38CKkK2axz27ED/e0qNjI7PoedC8sJklNElBz4

vyRg==

X-Gm-Message-State: AJaThX4s6clRc8gmk3Ea3xgnInlkzo+C8q32TE8ZBNn8hoAws+fUUYDC VoPhiA35VgyXWSwok+DNJ6Ybd5HVuOwpPfpO9Nk=

X-Received: by 10.25.201.83 with SMTP id z80mr1896982lff.181.1512011552189; Wed, 29 Nov 2017 19:12:32 -0800 (PST)

MIME-Version: 1.0

Received: by 10.25.155.83 with HTTP; Wed, 29 Nov 2017 19:12:31 -0800 (PST)

From: Shefali Sharma <shefali92@gmail.com>

Date: Wed, 29 Nov 2017 22:12:31 -0500

Message-ID: <CADuqHwJRO0ecCBd3sVUaPzztdj5o8bHdUEgiaOi7UhV4rLLnng@mail.gmail.com>

Subject: Test Email

To: sharma92@buffalo.edu

Content-Type: multipart/alternative; boundary="001a114b942676f0c9055f2a9f7f"

--001a114b942676f0c9055f2a9f7f

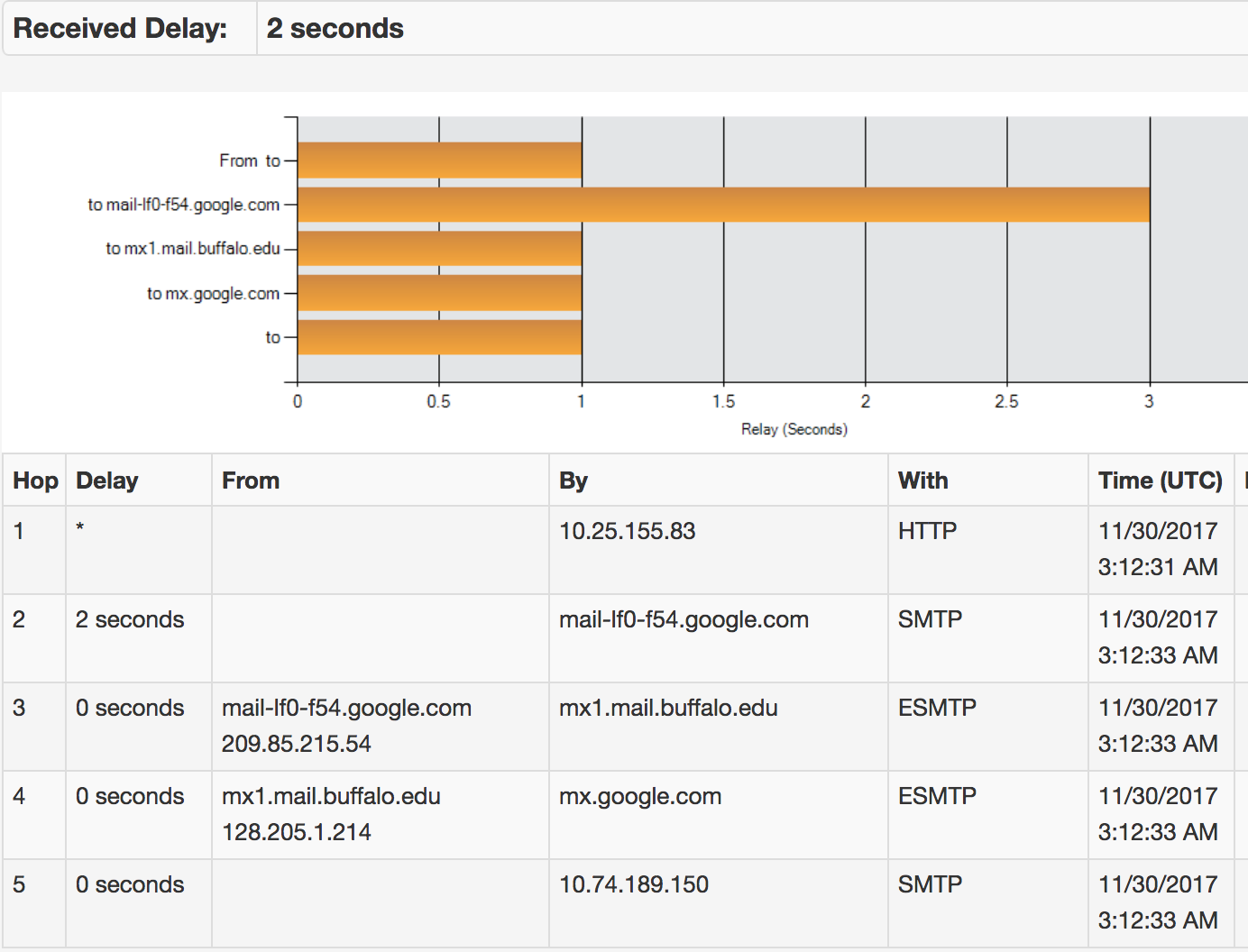
Content-Type: text/plain; charset="UTF-8"

--001a114b942676f0c9055f2a9f7f

Content-Type: text/html; charset="UTF-8"

<div dir="ltr"><br></div>

--001a114b942676f0c9055f2a9f7f—



**Header for email 2: (sent from Shefali Sharma <sharma92@buffalo.edu>**

**to sharma92@buffalo.edu)**

MIME-Version: 1.0

Received: by 10.74.195.197 with HTTP; Wed, 29 Nov 2017 19:13:47 -0800 (PST)

Date: Wed, 29 Nov 2017 22:13:47 -0500

Delivered-To: sharma92@buffalo.edu

Message-ID: <CAHgK+OECJGVwT0BxppNgsyYbxu7ev\_uNoNbGNh-D7UCD\_8pdQA@mail.gmail.com>

Subject: Test Email

From: Shefali Sharma <sharma92@buffalo.edu>

To: sharma92@buffalo.edu

Content-Type: multipart/alternative; boundary="089e08283bc4f6de34055f2aa376"

--089e08283bc4f6de34055f2aa376

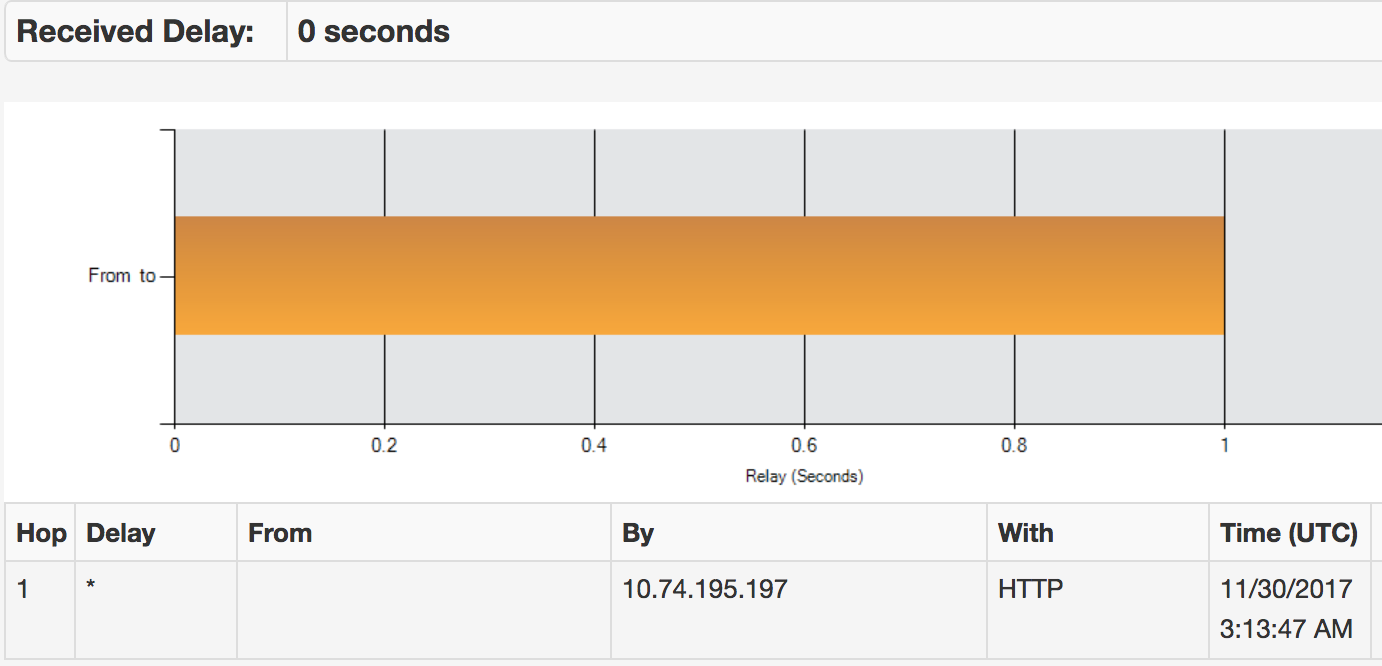
Content-Type: text/plain; charset="UTF-8"

--089e08283bc4f6de34055f2aa376

Content-Type: text/html; charset="UTF-8"

<div dir="ltr"><br></div>

--089e08283bc4f6de34055f2aa376--



**Section 1.1:**

**Explain the full header of both e-mails received in your UB mailbox. Try to give as much details as possible. Try to describe it using the timestamps provided on the header.**

* **Delivered-To:** It contains the E-Mail address of the receiver to whom the E-Mail is being addressed. The value of this field was sharma92@g-mail.buffalo.edu for the mail sent from Gmail while for the one sent from UB account it was sharma92@buffalo.edu.
* **Received:** These fields define all the intermediate nodes through which the email went before reaching the receiver. To detect the path by which emails were sent, we need to check Received header in the bottom-up fashion. This field includes the sender’s IP address, optional details on Transport Layer Security, the server that processed the message and a timestamp. For the mail sent by Gmail, we have more intermediate nodes while for UB account mail we have only one intermediate node.
* **Return-Path:** This field defines the return path for this email i.e the id that reply would be sent to.
* **Authenticated Received Chain** (**ARC**) is an email-authentication system designed to allow an intermediate mail server like a mailing list or forwarding service to sign an email's original authentication results. This allows a receiving service to validate an email when the email's SPF and DKIM records are rendered invalid by an intermediate server's processing.
* **Received-SPF:** MTA adds the above advisory header to message so that SPF results can be verified. These headers must be added above “Received” header for the given MTA so that it becomes clear that trusted MTA added header to the message.
* **Sender Policy Framework** (**SPF**) is a simple email-validation system designed to detect email spoofing by providing a mechanism to allow receiving mail exchangers to check that incoming mail from a domain comes from a host authorized by that domain's administrators.[[1]](https://en.wikipedia.org/wiki/Sender_Policy_Framework#cite_note-1) The list of authorized sending hosts for a domain is published in the Domain Name System (DNS) records for that domain in the form of a specially formatted TXT record. Email spam and phishing often use forged "from" addresses, so publishing and checking SPF records can be considered anti-spam techniques.
* **Domain Keys Identified Mail** (**DKIM**) is an email authentication method designed to detect email spoofing. 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.
* **Authentication-Results:** It contains values for the authentication results for the sender’s server. It may include SPF or DKIM results. Spf=pass indicates that domain name is authenticated by the mx.google.com’s mail exchange server.
* **Content-Type:** It describes the data in the body such that the receiving user agent can handle the data in the right way. The Content-type header includes type and sub-type fields to indicate the actual nature of the data. The sub-type field specifies the data format. Content-type: Plain text indicates plain-text with UTF-8 encoding character set. Content-type: multipart/alternative indicates that the body has multiple parts along with boundary to identify different parts.
* **MIME-Version:** This field gives the MIME version being used for this particular email. 1.0 – version of MIME is supported.
* **To:** It contains the recipient’s name and address. This may be different from the Delivered-To field as it is in the mail sent from Gmail.
* **From:** This field defines the name and email ID of the sender for the email.
* **Subject:** It contains the actual mail content which the sender would like the receiver to know.
* **Date:** This field is the timestamp for the email, telling when the email was sent, including its date and time.
* **Message-ID:** It is the unique identifier of an E-Mail message. It follows a specific set of rules based on the timestamp and the sender’s domain.

By comparison, we can see that the email sent from Gmail took more hops to reach the destination and took more time as sending date and time is Wed, 29 Nov 2017 19:12:31 -0800 (PST) and received 2 seconds later at Wed, 29 Nov 2017 19:12:33 -0800 (PST).

While mail sent from buffalo account to buffalo mail id was received in the same second my perhaps some microsecond difference as sending time is Wed, 29 Nov 2017 19:13:47 -0800 (PST) and receiving time is Wed, 29 Nov 2017 19:13:47 -0800 (PST).

**Section 1.2: What are the unique identifiers of these messages contained on the header? Are these fields spoof-vulnerable? Justify your answer.**

Unique identifiers in the headers above are message-id. Message-IDs are required to have a specific format which is a subset of an email address and to be globally unique. That is, no two different messages must ever have the same Message-ID. A common technique used by many message systems is to use a time and date stamp along with the local host's domain name

These are supposed to be unique but can easily be forged. It is quite easy to spoof this message ID because the message ID has certain set of rules that it follows, but by spoofing just this message ID the email cannot be simply classified as spoofed because other fields may not be easy to spoof. There also may be SPF or DKIM authentication sent along with the mail, which would make spoofing harder.

**Section 1.3: Show diagrammatically the network path traversed by both the e-mails. Use IP address as well as server name to identify intermediate nodes. Are the paths same or different? Justify your answer**

**Network Diagram for the flow of message from Gmail to UB account:**

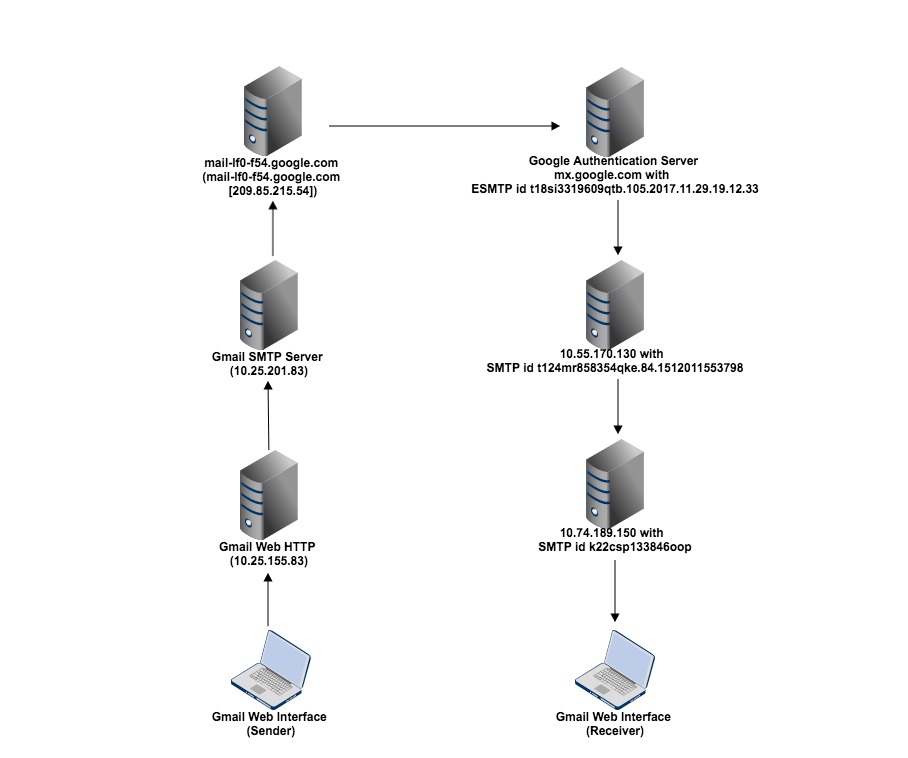


Figure 1

**Network diagram for the flow of message from UB account to UB account:**

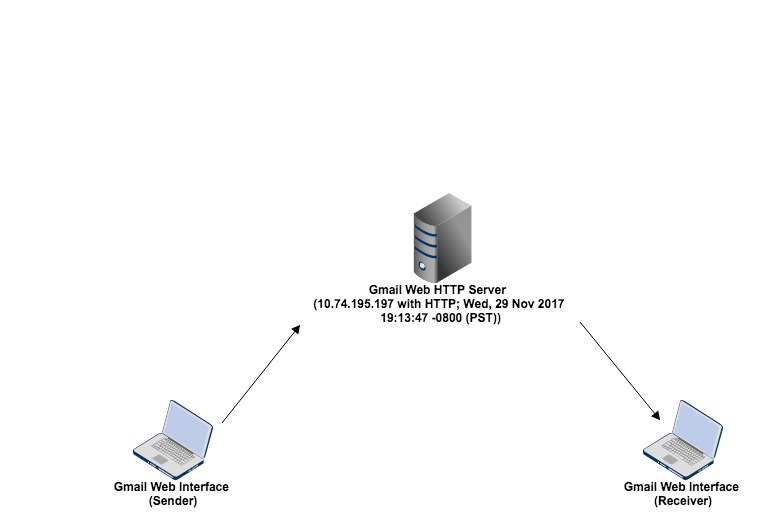


Figure 2

While sending a message from Gmail account to UB account, the mail is sent from Gmail ID. The message hence has to move from the path shown in Figure 1, where it moves from Gmail web interface to HTTP client to google servers and then UB SMTP server and so on. From UB server, it is then routed to the UB mail ID of the recipient. But, when we send a message from UB account to UB account, since the message does not have to pass through any other server, it is directly routed from UB mail server back to the UB user ID that belongs to the same UB server. **The paths traversed from both the servers is hence different.**

**Section 1.4:**

**Can we characterize one of these e-mails as a spoofed e-mail? Why or Why not? Justify the cases of both e-mails.**

Neither of the emails sent can be classified as email spoofs, rather it a case of email aliasing.

In case 1, Gmail authenticates the UB email ID before adding it to an existing account. ESMTP will not forward any emails originating from a blacklisted domain or if the mail seems to be forged. Since these emails were not tampered with or replicated they can be tagged as original ones. Spoofing can be detected if the source IP address remains the same with change in authenticated domain names. Spoofing can also be detected using inconsistencies in timestamp and message ID. Also, the Sender Policy Framework(SPF) for this email is pass, so we can somewhat trust the sender.

In case 2, the message is travelling inside the same server after being sent from UB email ID, from where it is redirected back to the UB mail. Since it does not travel through any other network entity, it is not forged or spoofed and that is why this mail did not contain any authentication test, due to the fact that mail was sent from @buffalo.edu domain to @buffalo.edu domain.

**Section 1.5:**

**Using your knowledge from Section 1.1 through Section 1.4, describe briefly how you can deduce a conclusion on a suspected e-mail spoofing case using the header information.**

We should look at the email header to find out if it is a email spoofing case. The header contains information on the route the email message took as it was being transmitted across the Internet on its way into your email inbox.

* Look for any differences between the name that looks like the name of the person you know and the actual e-mail address in the FROM field. If the friendly name is “BIG BANK of US” but the e-mail address is jimmy@con\_artists.com, or if the name in the FROM field is missing or spelled incorrectly, the e-mail is probably spoofed. The experienced email spoofer won’t make this mistake though.
* The Tracert command with display the route, along the network, from your computer to the IP address specified. Things to look for are dubious names of servers or signs that indicate geographical locations (e.g., SEA for Seattle).
* If the message header shows the use of some script to make the message instead of an email client application.
* The SPF record helps identify the mail server’s authenticity to send emails on behalf of a domain. If the value of SPF field is pass, it means the client is authorized to inject a given mail with the given identity. Otherwise, the email could be spoofed if it shows fail.
* If the message is relayed via an unusual and random location, it could mean that the message was spoofed/modified there and then sent.

**Phase 2: Deliverables**

**Section 2.1 A brief description of how DKIM works. What are the components?**

DKIM (DomainKeys Identified Mail) was first developed by Yahoo to be used as a way of email validation and user authentication.

DKIM creates a domain-level authentication framework for email by using public-key technology and DNS record to prove the source and content of a message.

**Sending Servers**

There are two steps to signing an email with DKIM:

1. The domain owner generates a public/private key pair to be used for signing outgoing messages (multiple key pairs are allowed). The public key is published in a DNS TXT record, and the private key is made available to the DKIM-enabled outbound email server.
2. When an email is sent by an authorized user of the email server, the server uses the stored private key to generate a digital signature of the message, which is inserted in the message as a header, and the email is sent as normal.

**Receiving Servers**

1. The DKIM-enabled receiving email server extracts the signature and claimed from: domain from the email headers.
2. The public key is retrieved from the DNS system for the claimed from: domain.
3. The public key is used by the receiving mail system to verify that the signature was generated by the matching private key. A match effectively proves that the email was truly sent from, and with the permission of, the claimed domain and that the message headers and content have not been altered during transit.
4. The receiving email system applies local policies based on the results of the signature test. For example, the message might be deleted if the signature does not match.

**Functional Components of DKIM:**

* **MUA (Message User Agent):** Message is formatted and submitted to MHS via MSA. It processes the message for storage or delivery to the receiver.
* **MSA (Message Submission Agent):** It performs signing before enforcing the message into Internet standards and policies of hosting domain before relaying to MTA.
* **MTA (Message Transfer Agent):** This deals with the relaying message from MSA to MDA through one or more MTAs.
* **Verifying:** Verifying is performed by an authorized module within the verifying ADMD. Within a delivering ADMD, verifying might be performed by an MTA, MDA, or MUA.
* **MDA (Message Digest Agent):** This is the verifier of the message. It performs the verification test on the message and if the test is successful, it transfers the message from MHS to MS
* **MHS (Message Handling service):** A network consisting of the various components namely MSA, MDA and MTAs is together called a MHS.
* **Signing:** Signing will be done by a service agent within the authority of the message originator's Administrative Management Domain (ADMD). Signing might be performed by any of the functional components, in that environment, including: Mail User Agent (MUA), or Mail Submission Agent (MSA), Internet Boundary MTA. DKIM permits signing to be performed by authorized third-parties.

**Section 2.2: Graph plots along with the source code written. Attach the source code as an appendix.**

Figure 3. Graph showing the time taken in milliseconds for applying each algorithm(signing) for 5 test runs

Figure 4. Graph showing the time taken in milliseconds for applying each algorithm(verification) for 5 test runs

The source code used for the signing and verification is attached as an Appendix.

**Section 2.3: Explain which of the combination of core DKIM algorithms provides the best performance using email message size as the criteria.**

From the above plots, we can see that on an average, the RSA 1024 + SHA1 algorithm performs better than others with a smaller key, when run on a data of similar length. However, this algorithm is preferred to be used only when the security is not critical.

**Section 2.4: Also answer the following questions:**

1. **Briefly describe the problems with using S/MIME or PGP in emails.**Problems with using PGP:

* It is complex to use, due to which messages might be sent in an insecure manner.
* It does not integrate seamlessly with email clients.
* Before we can communicate via PGP, we first need to exchange keys. PGP makes this a problem for the users. It takes very long to encrypt/decrypt messages.

Problems with using S/MIME:

* If the certificate is lost, messages that were encrypted with that key cannot be decrypted.
* Issues may arise with trusting Certificate Authorities.

1. **How is DKIM different from these email signature schemes?**
   * In DKIM, the message signature is written to the message header fields so that it does not confuse recipients or MUA software.
   * Using domain names, rather than email addresses, provides advantage.
   * DKIM doesn’t provide repudiation or confidentiality, PGP and S/MIME provide.
   * DKIM does not need certificates issued by authorities as it uses uses DNS-based self-certified keys and the scope of DKIM is limited.
   * In DKIM, there is no dependency on public and private keys by certificate authorities.
2. **What are other broad categories of Domain Validations used? What does DKIM fall under?**The two broad categories are:
   * Using Digital Signatures
   * Using IP address

DKIM uses Digital Signatures.

1. **Briefly explain what does DKIM do for the signer and for the receiver?**

The responsible organization adds a digital signature to the message, associating it with a domain name of that organization.  Typically, signing will be done by a service agent within the authority of the message originator's Administrative Management Domain (ADMD). Signing might be performed by any of the functional components, in that environment, including: Mail User Agent (MUA), or Mail Submission Agent (MSA), Internet Boundary MTA. DKIM permits signing to be performed by authorized third-parties.

After a message has been signed, any agent in the message transit path can choose to validate the signature. Typically, validation will be done by an agent in the ADMD of the message recipient. Again, this may be done by any functional component within that environment. Notably this means that the signature can be used by the recipient ADMD's filtering software, rather than requiring the recipient end-user to make an assessment.

1. **Does DKIM signature signify that all the fields in the header information are not forged?**

In case of DKIM, signing agents may remove some of the header fields at the time of signing. The sender signs the message header and since all of the content of the message header may not be relevant and the sender can choose not to sign some of the information. Headers fields are most vulnerable to change in transit, so leaving insignificant fields unsigned will increase the chance of successful verification. However, these fields could be tampered in transit. Hence, a DKIM signature does not signify that all fields in the header are not forged.

**APPENDIX**

* Please find code in **signingEmail\_RSA\_SHA.cpp**.

**References:**

1. <http://mxtoolbox.com/Public/Tools/EmailHeaders.aspx>
2. <https://en.wikipedia.org/wiki/Authenticated_Received_Chain>
3. <https://en.wikipedia.org/wiki/Sender_Policy_Framework>
4. <https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail>
5. <http://www.dkim.org/info/dkim-faq.html>
6. <https://www.emailonacid.com/blog/article/email-development/what_is_dkim_everything_you_need_to_know_about_digital_signatures>
7. <https://www.icewarp.com/support/online_help/341.htm>
8. <https://blog.mailrelay.com/en/2012/09/27/what-is-the-dkim-system-and-how-does-it-work>
9. <https://luxsci.com/blog/email-encryption-showdown-smtp-tls-vs-pgp-vs-smime-vs-portal-pickup.html>