CSE 565 : Project 3 Report
E-mail Forensics with DKIM

Submission by: Charanya Sudharsan
Shri Sai Sadhana Natarajan
Sneha Parshwanath

PHASE 1

SECTION 1.1

Gmail to UB e-mail

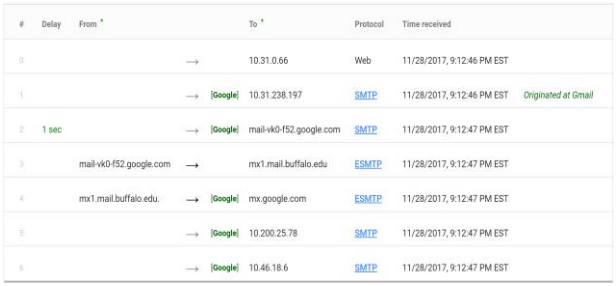
```
Delivered-To: csudhars@g-mail.buffalo.edu
Received: by 10.46.18.6 with SMTP id t6csp3868155lje;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
X-Google-Smtp-Source: AGs4zMbixGWQt9OaUy6FdlBi2f2RgXBOdbQVUfUd1Z2RWA/HnpCsKJpJw9td7D7QQb2EhWtRM5YP
X-Received: by 10.200.25.78 with SMTP id g14mr2151568qtk.119.1511921567916;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
ARC-Seal: i=1; a=rsa-sha256; t=1511921567; cv=none;
      d=google.com; s=arc-20160816;
      b=LhHlcmJm01VrHTNBI5LLCO+5tLmA/HwCSrXRf5IdDbMBsVfO1BedMIYJ+L3H+czsvw
       AiAnB57zgHPrJV103kFwNhaiFwCr2v63TPbVXAdF0bzDDEBjlyrCbMCiL6gFkrRhQfCK
       AFSGkHchefW8Wzh0ohLvzAZbhYuSYFBbJeABtn9cbTsGrBUIrSTmlhvFc2td/5nWn48/
       6RGPCFBmWpt4m4VIMPCUW4Vy0+jwrE/f0S2NhBc7XAyQ2NfbP5jW3XsJaQMVd7cMligN
       9P4u5K4IwbggAoKJSZeuW/waPdFrTVCe/3ktEkzncv95cVc2z1/kG/MmLWyWr4wkALZn
       oQyQ==
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=cGrEBKX7rQT8kebMAPh4ruNRH9pW5lMrSP1TYzJGauQ=;
      b=cCZjPgkBlCQyRr75XvijyTNdmOUMC5KYZ1Es+0MATmYybZU2zTPJlc2tWghRo8K8Vz
       0fIj24ODqS0TiLRxxQla12Z28/viuxlM7D4j2B+j1KcZwnUki20GT3NUGkwSWqb5b02L
       uxMvAtClXyiWeTDIvUjBiXOVcrlHCzj54KinRbpKh9Zs1gS+XPNlPdz5ZoQrrVPmKKnf
       nAJp3fWxN3ePrAVMyPpDrCbrw5/9RavDAkYKSzJpjYT0vaRp0ObwZitUoz67Jflv6/XD
       cSF/M0FsQgFv2Duk4CrYhSvzTnMSFnkN15agv7bZu+2yr8V0hAbXjQLxQPYBOx7Js/4I
       5mYg==
ARC-Authentication-Results: i=1; mx.google.com;
      dkim=pass header.i=@gmail.com header.s=20161025 header.b=DmrtSIGr;
      spf=pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) smtp.mailfrom=dues13@gmail.com
Return-Path: <dues13@gmail.com>
Received: from mx1.mail.buffalo.edu (mx1.mail.buffalo.edu. [128.205.1.214])
      by mx.google.com with ESMTP id t4si700755qtc.194.2017.11.28.18.12.47
      for <csudhars@g-mail.buffalo.edu>;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
Received-SPF: pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) client-ip=209.85.213.52;
Authentication-Results: mx.google.com;
      dkim=pass header.i=@gmail.com header.s=20161025 header.b=DmrtSIGr;
      spf=pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) smtp.mailfrom=dues13@gmail.com
Received: from mail-vk0-f52.google.com (mail-vk0-f52.google.com [209.85.213.52])
   by mx1.mail.buffalo.edu (mx) with ESMTP id 6B64E1000FF
   for <csudhars@buffalo.edu>; Tue, 28 Nov 2017 21:12:47 -0500 (EST)
Received: by mail-vk0-f52.google.com with SMTP id q189so300274vke.0
      for <csudhars@buffalo.edu>; Tue, 28 Nov 2017 18:12:47 -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=cGrEBKX7rQT8kebMAPh4ruNRH9pW5lMrSP1TYzJGauQ=;
      b=DmrtSIGrTVUGn0BQP0WjSK5FhFtUeghXfn6OZfNROudgmdcjXvGmSdo+QaeqvYYKOy
       tNXNmY46TcLF1vMP2v7rURse4zL3H45sHv+7tYdLsXAARsXWRKswi5622O5M46RiUTEI
       qE9UuyEVTJFig0dpIDeN37zs88pws+YSEaf1PX+p/yPe9USsMyS/CagIPURCEGz9TrkH
       2dWEcobVtIwlWSQk4Z6RdcHDRmzyAEabPgazY6Zz6FdiF39mxAwIE2za/UalSoHnDyTi
       /ImbK7vZeRH/r2WORivqXj+/5iCZLTffon0kK+bBfCD5ASaGByHa9AlYTTWoEsDbK7yJ
       3g3A==
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=cGrEBKX7rQT8kebMAPh4ruNRH9pW5lMrSP1TYzJGauQ=;
      b=nK/RXJdm+3l8WC5z6MWRdK6Dlig2z0pCPG7cjVKcExLG2Vx/pgmnLcx3rUA1+mUPz9
       NJQgrxz5K6bx13m//a26Ac2q6LMogHDOpmfEAmyj0WVo92ntwxeWlODUO9Xub0U+lD0z
       ed5U2W4wk8lPoOVKiYcL/WBsO6LcL0kV025kai+uprBp17byScF2JAXk6pXxIxNnPnmV
       hRQuJzxRuB9YY54LsFxsOEHUsBOtV3vlxLG5TrPE355xPWiBU7iwBOPYiVEp1fFZjlVB
```

Z5G0enkIMkEB5HjrO4Osa6cNBdnmX/b5pwN3L/peHV1W20nCL3RpXWiNt7s3vnpTeZoq
O++w==
X-Gm-Message-State: AJaThX7QzMxAinbi+Y1mQfqmNc2eZthYhvpmlr3UvWualB6nxJTyuxvO
qmE/Dvgyv+7BTeQOEy5r/aaN6kGBmpXVF+cYUT8=
X-Received: by 10.31.238.197 with SMTP id m188mr1102960vkh.190.1511921566848;
Tue, 28 Nov 2017 18:12:46 -0800 (PST)
MIME-Version: 1.0
Received: by 10.31.0.66 with HTTP; Tue, 28 Nov 2017 18:12:46 -0800 (PST)
From: Charanya S <dues13@gmail.com>
Date: Tue, 28 Nov 2017 21:12:46 -0500
Message-ID: <CAFYoVQe8f1XuiPt==44ca0UAo19-jF8Gr4SqoAQweWOc4EDQYg@mail.gmail.com>
Subject: Mail from Gmail to UBmail
To: csudhars@buffalo.edu
Content-Type: multipart/alternative; boundary="94eb2c14be8eeb9995055f15ab93"
--94eb2c14be8eeb9995055f15ab93
Content-Type: text/plain; charset="UTF-8"
Hey Charanya ,
This is a mail from Gmail to UB mail.
REgards,
Charanya
--94eb2c14be8eeb9995055f15ab93
Content-Type: text/html; charset="UTF-8"
Hey Charanya ,

 This is a mail from Gmail to UB mail.

 REgards,
 Charanya
--94eb2c14be8eeb9995055f15ab93--

Email Header Analysis :



1. **Common headers** (From, To, Created at (time & Date) , Subject , Message ID)

2. **Message ID :** <CAFYoVQe8f1XuiPt==44ca0UAo19-jF8Gr4SqoAQweWOc4EDQYg@mail.gmail.com>
   This is a unique identifier of gmail which is used to keep track of the mail that has been sent.

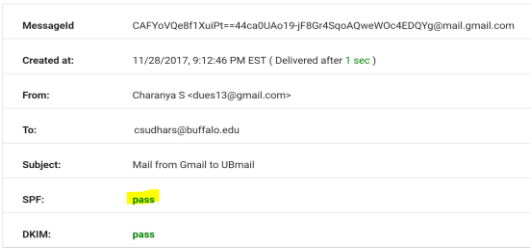
3. **DKIM Signature (Domainkeys Identified Mail):** Verifies sender's identity and check if mail is tamper proof . It has a domain name linked to the mail message. This line in the email header confirms that DKIM authentication is active.

   DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
   d=gmail.com; s=20161025;
   (Where 'd' is the sending domain, and 's' is the signing domain)

4. **Received Headers:** The Received stamps show the email and IP address of each sender and recipient, date and time of message transfer. The Mail Transfer Agents processes the email message and adds a Received stamp to the email header.

   a. Received: by 10.46.18.6 with SMTP id t6csp3868155lje;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
   b. X-Received: by 10.200.25.78 with SMTP id g14mr2151568qtk.119.1511921567916;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
   c. Received: from mx1.mail.buffalo.edu (mx1.mail.buffalo.edu. [128.205.1.214])
      by mx.google.com with ESMTP id t4si700755qtc.194.2017.11.28.18.12.47
      for <csudhars@g-mail.buffalo.edu>;
      Tue, 28 Nov 2017 18:12:47 -0800 (PST)
   d. Received-SPF: pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) client-ip=209.85.213.52;
   e. Received: from mail-vk0-f52.google.com (mail-vk0-f52.google.com [209.85.213.52])
      by mx1.mail.buffalo.edu (mx) with ESMTP id 6B64E1000FF
      for <csudhars@buffalo.edu>; Tue, 28 Nov 2017 21:12:47 -0500 (EST)
   f. Received: by mail-vk0-f52.google.com with SMTP id q189so300274vke.0
      for <csudhars@buffalo.edu>; Tue, 28 Nov 2017 18:12:47 -0800 (PST)

5. **SPF** : Sender Policy Framework (SPF) is an attempt to control forged e-mail. SPF is not directly about stopping spam – junk email. It is about giving domain owners a way to say which mail sources are legitimate for their domain and which ones aren't. While not all spam is forged, virtually all forgeries are spam. If it results as pass – the mail is not a spam .



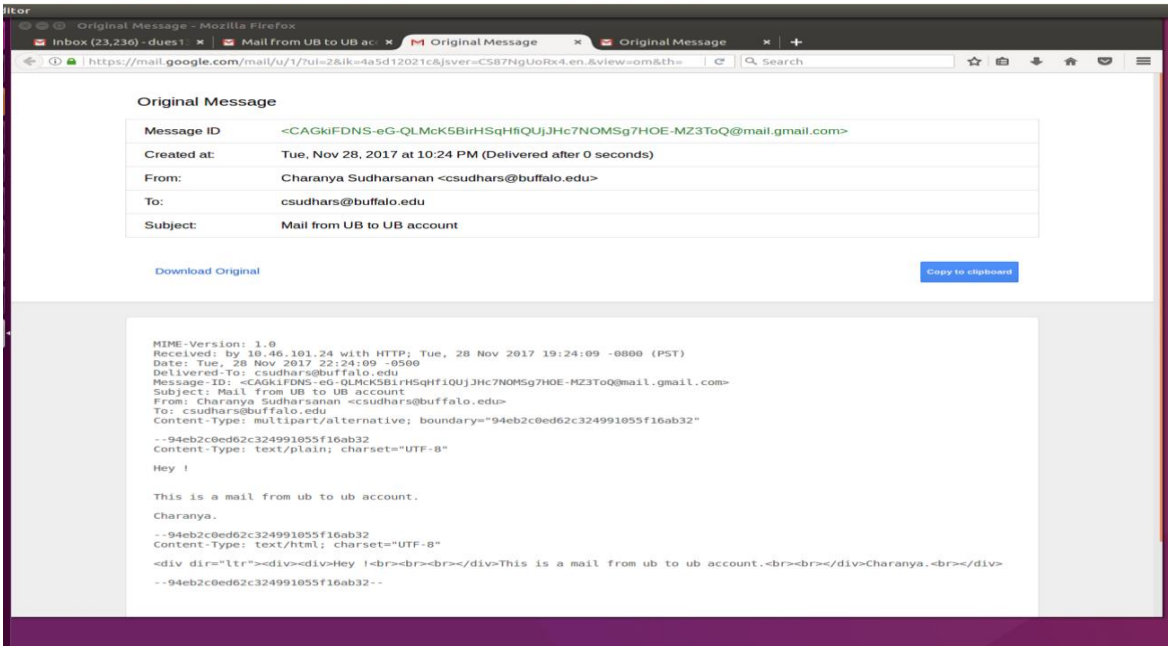
   Received-SPF: pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) client-ip=209.85.213.52;

6. **Time the mail reaches the receiver (here, the buffalo server)**
   Received: by 10.46.18.6 with SMTP id t6csp3868155lje;
   Tue, 28 Nov 2017 18:12:47 -0800 (PST)

7. **Return path** : Return-Path is the address where bounce messages (undeliverable notifications, etc.) should be delivered. Return-Path: <dues13@gmail.com>

UB TO UB E-MAIL

ubuntu [Running] - Oracle VM VirtualBox



MIME-Version: 1.0
Received: by 10.46.101.24 with HTTP; Tue, 28 Nov 2017 19:24:09 -0800 (PST)
Date: Tue, 28 Nov 2017 22:24:09 -0500
Delivered-To: csudhars@buffalo.edu
Message-ID: <CAGkiFDNS-eG-QLMcK5BirHSqHfiQUjJHc7NOMSg7HOE-MZ3ToQ@mail.gmail.com>
Subject: Mail from UB to UB account
From: Charanya Sudharsanan <csudhars@buffalo.edu>
To: csudhars@buffalo.edu
Content-Type: multipart/alternative; boundary="94eb2c0ed62c324991055f16ab32"
--94eb2c0ed62c324991055f16ab32
Content-Type: text/plain; charset="UTF-8"
Hey !
This is a mail from ub to ub account.
Charanya.
--94eb2c0ed62c324991055f16ab32
Content-Type: text/html; charset="UTF-8"
Hey !

 This is a mail from ub to ub account.

 Charanya.
--94eb2c0ed62c324991055f16ab32--

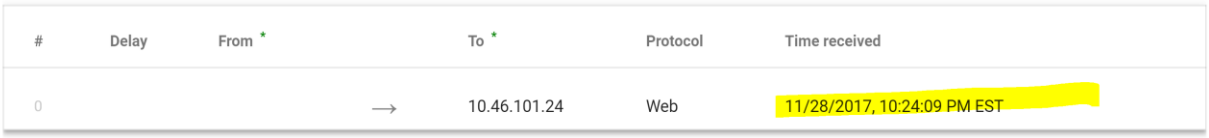
Email Header Analysis:

1. **Common headers:** (From, To, Created at ,Subject, Message ID)

MessageId	CAGkiFDNS-eG-QLMcK5BirHSqHfiQUjJHc7NOMSg7HOE-MZ3ToQ@mail.gmail.com
Created at:	11/28/2017, 10:24:09 PM EST (Delivered after)
From:	Charanya Sudharsanan <csudhars@buffalo.edu>
To:	csudhars@buffalo.edu
Subject:	Mail from UB to UB account

2. **Message – ID** is a unique value given to identify the mail
CAGkiFDNS-eG-QLMcK5BirHSqHfiQUjJHc7NOMSg7HOE-MZ3ToQ@mail.gmail.com

3. **Time** to reach Reciever (Buffalo Server)



#	Delay	From *		To *	Protocol	Time received
0			→	10.46.101.24	Web	11/28/2017, 10:24:09 PM EST

SECTION 1.2

Message-ID is a unique identifier for a digital message, most commonly a globally unique identifier used in email. 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.

To check forgery, you should verify that the time in the Message Id field matches the time in the Date field for the message. The time in the Date field should be in the Sender's local time and the Message Id will be in Google's local time, US Pacific Time. The message id headers can prove useful when trying to determine if a email is authentic. Although they can't always prove that message is authentic, they can often show that a message has been forged. Message ID is not spoof vulnerable as every single message has a globally unique identifier generated every time .

DKIM Signatures :

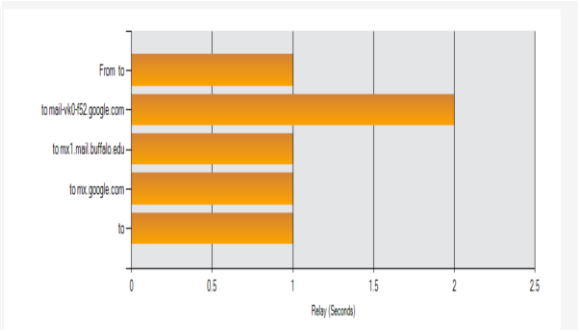
DomainKeys Identified Mail (DKIM) is a popular email authentication technology that allows for a domain to prove it is responsible for a message and it was not altered as it traveled the delivery path.
The DKIM signature will be generated in a unique textual string, the 'hash value'. Before sending the email, the hash value is encrypted with a private key, the DKIM signature. Only the sender has access to this private key. When the email is encrypted the email is sent with this DKIM signature. Email receivers, like Gmail and Microsoft (Hotmail, Outlook etc), detect the DKIM signature. This DKIM signature reveals which domain was used to sign the email in the encryption process. To validate the DKIM signature, the email receiver will run a DNS query to search for the public key for that domain. The variables provided in the DKIM signature are used to determine where to look for this key. If the key was found, it can be used to decrypt the DKIM signature back the to original hash values. These values are compared to the new values retrieved from the received mail. If they match, the DKIM was valid. Each "tag" in a DKIM Signature is associated with a value.

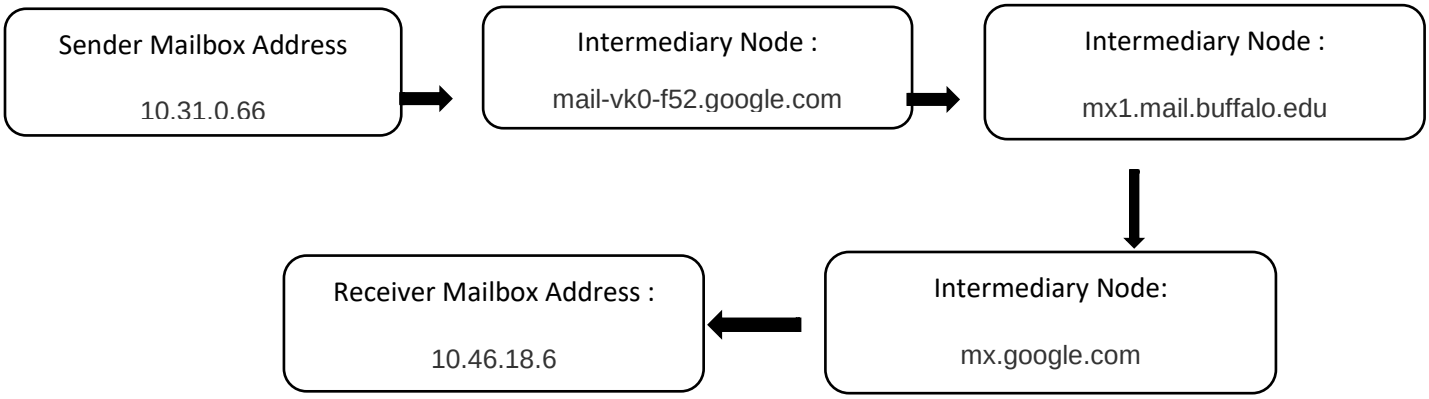
- b = the actual digital signature of the contents (headers and body) of the mail message
- bh = the body hash
- d = the signing domain
- s = the selector
- v = the version
- a = the signing algorithm
- c = the canonicalization algorithm(s) for header and body
- q = the default query method
- l = the length of the canonicalized part of the body that has been signed
- t = the signature timestamp
- x = the expire time
- h = the list of signed header fields, repeated for fields that occur multiple times
  Here, tags 'b' and 'bh' are uniquely generated for every message, hence making it not vulnerable to spoof. Similarly the time and date stamp together is definitely unique for every message too making it not spoof vulnerable.

SECTION 1.3

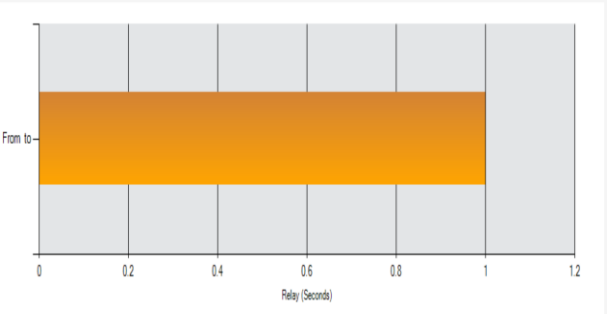
Gmail to UBmail:
Flow of mail from Gmail to UBMail :

Hop	Delay	From	By	With	Time (UTC)	Blacklist
1	*		10.31.0.66	HTTP	11/29/2017 2:12:46 AM	
2	1 Second		mail-vk0-f52.google.com	SMTP	11/29/2017 2:12:47 AM	
3	0 seconds	mail-vk0-f52.google.com 209.85.213.52	mx1.mail.buffalo.edu	ESMTP	11/29/2017 2:12:47 AM	
4	0 seconds	mx1.mail.buffalo.edu 128.205.1.214	mx.google.com	ESMTP	11/29/2017 2:12:47 AM	
5	0 seconds		10.46.18.6	SMTP	11/29/2017 2:12:47 AM	

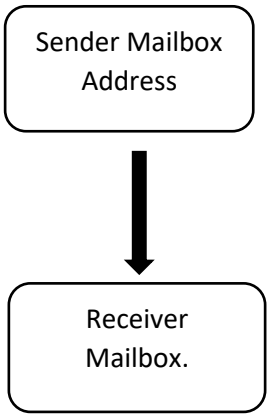




UBmail to UBmail:
Flow of mail from UBmail to UBMail :



Hop	Delay	From	By	With	Time (UTC)	Blacklist
1	*		10.46.101.24	HTTP	11/29/2017 3:24:09 AM	



Analysis:
The Network paths taken by both the emails are different . First Email needed a cross domain reference as it was sent from one server to another (Gmail to UB), Hence the use of several intermediary nodes. These nodes verify the DKIM signature. If the DKIM test results out as pass , then the signature is verified (No Email spoofing) . Whereas in the second case there is no need for any cross domain reference as the mail is sent within a single server, hence no DKIM signature verification for email spoofing is needed too.

SECTION 1.4

Gmail to UBMail

Due to the presence of DKIM and SPF signature , we can say that this email has not been spoofed.
If it results as pass – the mail is not a spam .This identifies and authorizes it as a permitted sender. If this results in a Pass , the sender is acceptable to the verifier.

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
d=gmail.com; s=20161025;
dkim=pass header.i=@gmail.com header.s=20161025 header.b=DmrtSIGr;
spf=pass (google.com: domain of dues13@gmail.com designates 209.85.213.52 as permitted sender) smtp.mailfrom=dues13@gmail.com

UBMail to UBMail

This case does not have any authentication mechanism associated with it . It uses basic SMTP and doesn't have DKIM or SPF . Hence this mail can be spoofed . An Intruder having access to mail.buffalo.edu server can forge the mails by changing the headers and send out spoof emails without much hassle.

SECTION 1.5

Following Elements help us detect if spoofing has occurred.

Message ID : Message ID is basically used to identify each mail and denotes a unique identity of an email. Spoofing of this unique ID needs special & technically clever skills and cannot be spoofed as easily as the other elements are spoofed. This factor of it can help investigators as there is less or no possibility of spoofing the Message ID. Message-ID is another element which can make the investigation more convenient as any type of forging done with the email will definitely reflect through its message ID. Moreover, details like time-stamp, date, etc. can also be tallied through this message-ID.

SPF : With an SPF record, the address of your stated, trusted originating mail server(s) is always compared against the originating mail server's address information in the email headers. If they don't match (and they won't if they've been spoofed), it's easy to spot and dump out a forged email. That email will not be delivered, thereby frustrating the spammer's efforts.

DKIM : DomainKeys 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. The two main components of DKIM are cryptography and DNS. DNS is used to publish a public key. As DKIM is usually used to secure the communication in infrastructure-level and not per user, the entire verification process is usually hidden for end users.

IP : The most obvious method for detecting "SPOOFED" email is to look in the FROM field of the email. If the e-mail address displayed is different from the known e-mail address of the person who supposedly sent it, then you know it's a spoof. Looking up the IP address we can verify senders location and mail server . If this does not match with the header , mail is spoofed. We could also find the sender score using the sender's IP address using the return path site. Sender Score is a number between **0 and 100** that identifies your sender reputation and shows you how mailbox providers view your IP address. Mailbox providers take a lot of metrics into consideration to determine your sender reputation including spam complaints, mailing to unknown users, industry blacklists, and more.

PART 2

Section 2.1

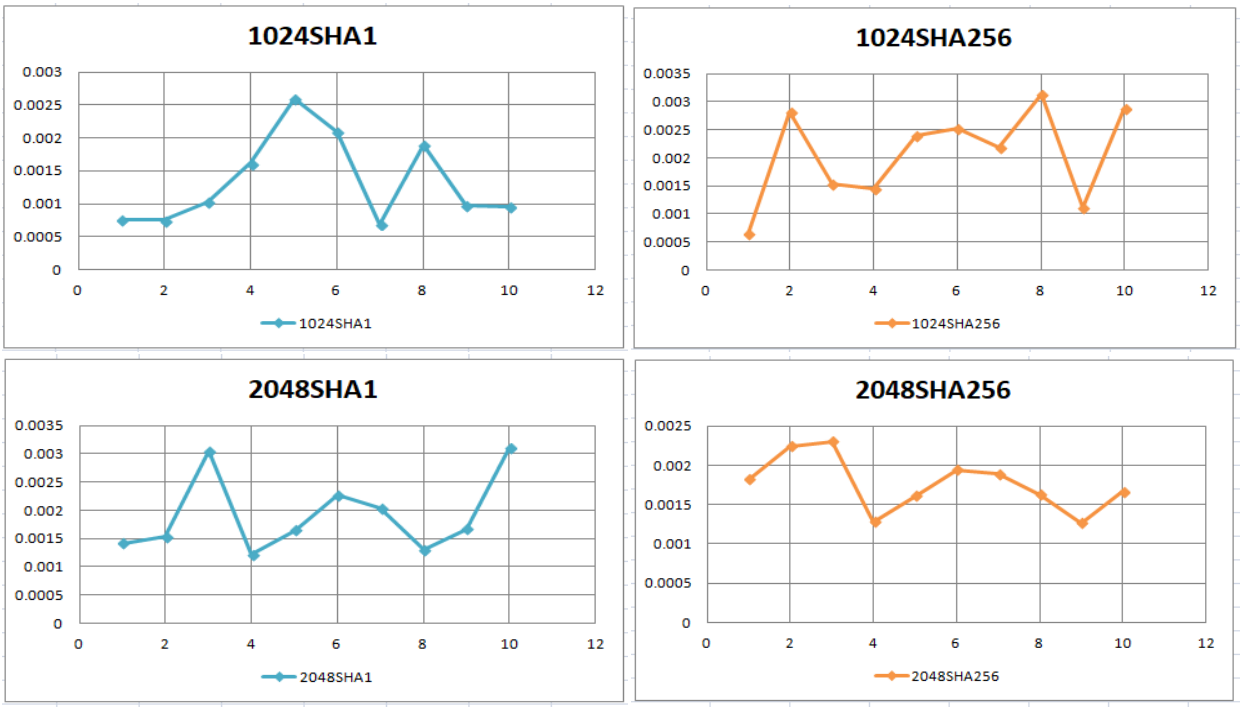
DKIM provides a transparent email authentication method for validating a domain name identity that is associated with a message through cryptographic authentication. It consists of two operations – signing and verifying. The email message content and selected RFC 5322 header is signed by the administrative domain from which the email message originates using a private key. The MDA is able to verify the authenticity of sender domain by accessing the

corresponding public key through DNS, verifying the signature and then passes the email to the recipient email client. This method protects against email spoofing.RSA+SHA-256 is the default signing algorithm used.

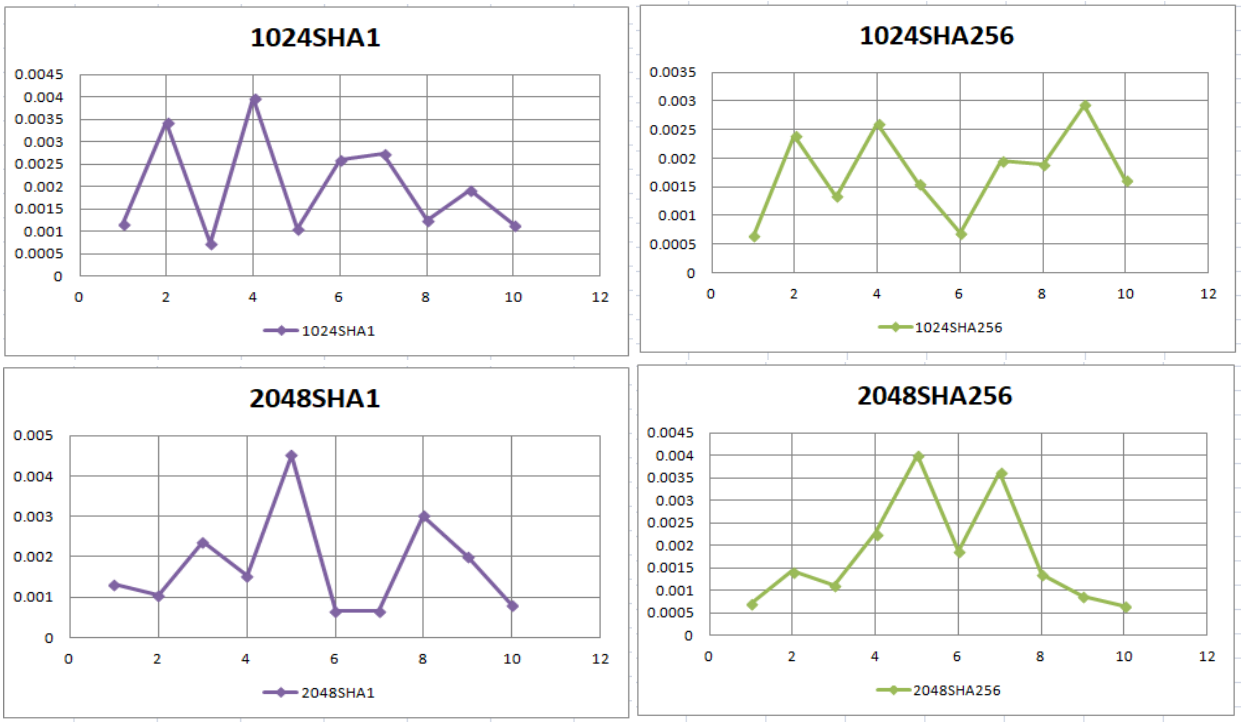
Tag	Description
v	Version - this specifies the version of the DKIM spec we are compliant with
h	A colon-separated list of header field names that identify the header fields presented to the signing algorithm
k	this defines the key type being used and defaults to RSA
l	length of the canonicalized part
p	the base64 encoded ASN.1 DER-encoded RSA public key
s	DKIM defines a selector (a name associated with a key) that is used by the verifier to retrieve the proper key during signature verification.
t	signature timestamp
c	Canonicalization algorithm(s) for header and body
q	Default query method
d	domain name used as an identifier to refer to the identity of a responsible person or organization
bh	The hash of the canonicalized body part of the message
b	The signature data in base64 format

Section 2.2**: Graph plots**

Signing:



Verifying:



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

We can conclude that 2048SHA256 takes more time than any other algorithm from the above graphs. The time taken to apply these is directly proportional to the size of the messages. 1024-bit algorithm is comparatively faster than 2048-bit algorithm.SHA256 is more secure than SHA1. Therefore, we can employ 1024SHA1 to get maximum performance in less time. However, if maximum security is the goal it is better to employ 2048SHA256 is it provides high security.

Section 2.4

1) Briefly describe the problems with using S/MIME or PGP in emails.
   - Both S/MIME and PGP requires the originator's s private key to sign the email message. This is an overhead to the sender. They do not involve domain owner.
   - For S/MIME, both sender and the receiver must employ S/MIME.
   - S/MIME and PGP only signs the message contents hence the header RFC 5322 header information can be compromised.
   - PGP an S/MIME requires prior exchange of keys and maintaining the collection of keys for every communication pair.
   - Encryption and decryption processes are not transparent to the end users.
   - S/MIME and PGP certificates are expensive.
   - It is more susceptible to e-mail spoofing.
   -
2) How is DKIM different from these email signature schemes?
   - DKIM is transparent to the end users. It is not implemented in client programs (MUAs).
   - Email is signed by a private key of the administrative domain from which the email originates (not the sender's private key).

- It allows senders to prove that they did send a particular message and prevents masquerade attack.
- The message contents + selected header RFC 5322 fields are signed for verification.
- It applies to all mails from cooperating domains.

3) What are other broad categories of Domain Validations used? What does DKIM fall under?

a) IP addressing: This includes SPF (Sender Policy Framework), Sender ID and CSV (Certified Sender Validation).

b) Digital Signatures: Here the sender/sending domain apply a digital signature on the message for verification.DKIM falls under this category.

4) Briefly explain what does DKIM do for the signer and for the receiver?

DKIM allows the signer to take responsibility for sending the message. It ensures that domain can protect its reputation and defend itself against false accusations. Receiving domains which trust the sending domain can handle mails sent by this domain with higher preference.

DKIM helps the verifier to authenticate that the email came from the domain it said it did. The receiving domain can blacklist and whitelist sending domains with help of DKIM.A signed mail is more trusted than unsigned mails. The receiver can also process mails from trusted sending domains faster.

5) Does DKIM signature signify that all the fields in the header information are not forged?

No, DKIM signature cannot assert or signify that all the fields in the header information are not forged. This is because, DKIM enforces that some header fields like from header field must be signed but some fields like Return-Path header field must not be signed. The choice of which headers to be signed is left to the discretion of the signer, hence there may be some unsigned header fields that can be forged/modified.

APPENDIX

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>

int main(int argc, char *argv[]){
        char* bitSize=argv[1];
        char* shaType=argv[2];
        int status = 1, i=0, k=0;
        FILE *fp;
        printf("Signing the message");
        while(i<10){
                double time;
                clock_t start;
                char outputfile[100], keyfile[100], mailfile[100], sha[100];
                sprintf(sha, "-%s", shaType);
                sprintf(outputfile, "../emails/Output/%s%s/cipher%d.txt", bitSize, shaType, i+1);
                sprintf(mailfile, "../emails/msg%d.txt", i+1);
                sprintf(keyfile, "../rsaprivatekey%s.pem", bitSize);
                char* command[] = {"openssl", "dgst", shaType, "-out", outputfile, "-sign", keyfile, mailfile, NULL};
                start = clock();
                for(int j = 0; j < 5; j++){
                        if(fork()==0)
                                execvp(command[0],command);
                        else wait(&status);
```

```
                }
        time = (clock() - start)/CLOCKS_PER_SEC;
        fp=fopen("/home/gurukrupa/Documents/comp.csv", "w");
        fprintf(fp,"%lf\n", time);
        i++;
        fclose(fp);
        }
        printf("Verifying the message");
        while(k<10){
                double time;
                clock_t start;
                char outputfile[100], keyfile[100], mailfile[100], sha[100];
                sprintf(sha, "-%s", shaType);
                sprintf(outputfile, "../emails/Output/%s%s/cipher%d.txt", bitSize, shaType, k+1);
                sprintf(mailfile, "../emails/msg%d.txt", k+1);
                sprintf(keyfile, "../rsapublickey%s.pem", bitSize);
                char* command[] = {"openssl", "dgst", shaType, "-verify", keyfile, "-signature", outputfile, mailfile ,NULL};
                start = clock();
                for(int m = 0; m < 5; m++){
                        if(fork()==0)
                                execvp(command[0],command);
                        else wait(&status);
                }
        time = (clock() - start)/CLOCKS_PER_SEC;
        fp=fopen("/home/gurukrupa/Documents/comp.csv", "w");
        fprintf(fp,"%lf\n", time);
        k++;
        fclose(fp);
        }

return 0;
}
```

References

https://en.wikipedia.org/wiki/DomainKeys_Identified_Mail
https://rietta.com/blog/2012/01/27/openssl-generating-rsa-key-from-command/
http://www.dkim.org/
https://returnpath.com/senderscore-privacy-policy/
https://www.emailonacid.com/blog/article/email-development/what_is_dkim_everything_you_need_to_know_about_digital_signatures
http://forensicswiki.org/wiki/Using_message_id_headers_to_determine_if_an_email_has_been_forged
https://mxtoolbox.com/Public/Tools/EmailHeaders.aspx?huid=b5ce6d4f-8b25-4fb6-a26b-90a997d018db

Paper: http://internetmessagingtechnology.org/pubs/CEAS-2007-078-DKIM.pdf
Book: Cryptography and Network Security, 7e, William Stallings

Contributions

Part 1 : Charanya Sudarsan Part 2: Sneha Parshwnath and Sri Sai Sadhana Natarajan