**11/13/2015** **CSE 565: Computer Security** **Due: 11/24/2015 (in Class)**

# Project 3

# E-mail Forensics with DKIM

## 1. Background

Digital Forensics is an inevitable topic in Computer Security. It is about finding useful information on digital media such as a computer system, storage media, and communication system that might help solve a civil or cyber crime. The evidence is sometimes referred to as digital artifact.

The main objective of this project is to give you the basic idea of how digital investigation is done in the real world. As E-mail is widely used for communication in the real world, you will work with those to do some hands-on forensics investigation.

## 2. Prerequisites

This project is in two phases. For all the phases you will need to use hotmail (Windows Live), Gmail and Yahoo accounts. If you do not have one, you can sign up for one at https://login.live.com, https://mail.google.com, https://login.yahoo.com.

For all the phases, you will use a Linux (Ubuntu) physical/virtual machine as the platform. Use **only** the platform and software indicated in this project description. No exception is made without explicit prior permission of the instructor. You should have openssl installed on the machine. Information of installation of virtual machine was included in Project 2 handout.

## 3. Operational Details and Deliverables

In Phase 1, you will investigate a "potential" E-mail Spoofing case. A brief description of it is found at: http://www.webopedia.com/TERM/E/e_mail_spoofing.html.

In Phase 2, you will first collect a repository of 10 email messages sent from team member 1 to team member 2 using hotmail (Windows Live) account **only**. Later write a simple C code to apply the core DKIM algorithms on the email repository, which are RSA (1024 & 2048 bits) and SHA1/SHA256.

### 3.1 Phase 1: Investigate E-mail Spoofing

You will investigate two e-mail headers to identify a "potential" E-mail Spoofing. First send two e-mails to your UB e-mail account following the directions given below:

I. Set up your Gmail account to allow you to send e-mail using your UB e-mail account.
II. Send an e-mail from your Gmail account to your UB account using your UB e-mail ID.
III. Send an e-mail from your UB account to your UB account using your UB e-mail ID.
IV. For both Gmail and UB e-mail, access mailboxes using web browser only. For UB e-mail use: https://ubmail.buffalo.edu/.

### 3.2 Phase 1: Deliverables

Include the following sections in your project report for Phase 1:

**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.

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

### 3.3 Phase 2: Separation of HAM and SPAM

The operational details of this phase are given below:

I. Collect a repository of 10 email messages of various sizes (in kb) sent from team member 1 to team member 2 using hotmail (Windows Live) account **only**. Store the email messages as msg.1 … msg.10.
II. Write a simple C code **using Fork – Exec system calls** to apply the core DKIM algorithms on the email repository collected above. The steps are as follows:
   - Generate the public – private RSA key pairs for both 1024 and 2048 bits using openssl command. Store them as **rsaprivatekey1024.pem** & **rsapublickey1024.pem** and **rsaprivatekey2048.pem** & **rsapublickey2048.pem.**
   - Apply both SHA digests SHA1 and SHA256 and sign the messages using the RSA private keys generated above (applying the SHA digest and signing can be achieved by using a single openssl command, i.e., SHA1/SHA256 + 1024/2048 RSA Signing).

| Algorithm | Output folder (store the cipher generated here as cipher.1 … cipher.10) |
|---|---|
| SHA1 + rsaprivatekey1024 signing | Output/1024SHA1 |
| SHA256 + rsaprivatekey1024 signing | Output/1024SHA256 |
| SHA1 + rsaprivatekey2048 signing | Output/2048SHA1 |
| SHA256 + rsaprivatekey2048 signing | Output/2048SHA256 |

   - Calculate the time taken in **milliseconds** for applying each of the above algorithms for 5 test runs using any standard C programming time libraries. Plot a graph depicting the same for each of them using the test run data.
     ***Note***: You will be graded based solely on how readable your graphs are.
   - As a next step **verify** the RSA signature using the **rsapublickey1024 and rsapublickey2048 public keys** generated above and taking the ciphers from the output folders above along with the original email messages as input for verification.
   - Calculate the time in **milliseconds** for verification of each of the algorithms in the table above for the 5 test runs using any standard C programming time libraries. Plot a graph depicting the same for each of them using the test run data.

### 3.4 Phase 2: Deliverables

Include the following sections in your project report for Phase 2:
**Section 2.1:** A brief description of how DKIM works. What are the components?

**Section 2.2:** Graph plots along with the source code written. Attach the source code 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.
**Section 2.4**: Also answer the following questions:

1) Briefly describe the problems with using S/MIME or PGP in emails.
2) How is DKIM different from these email signature schemes?
3) What are other broad categories of Domain Validations used? What does DKIM fall under?
4) Briefly explain what does DKIM do for the signer and for the receiver?
5) Does DKIM signature signify that all the fields in the header information are not forged?

**4. Project Guidelines**

The project has no demo component. Grading will be entirely based on the project report which should not be more than 5-6 pages. So, provide as much details as possible using diagrams and screenshots. For the projects, you continue to work in groups as formed before. This is a repeat project from previous years due to its high value as a learning component. If you refer to any particular paper or web document for this project, remember to mention the source in your report. Otherwise, such actions will be construed as plagiarism. In any case, do the complete project on your own and enhance your skills. Do not take any shortcuts or outsource the effort. The hardcopy report is due in class on the submission day. If you cannot come to class (in case you like to start your Thanksgiving break early, but not encouraged) on that day, you could submit the report anytime prior to the date. No softcopy is accepted. Enjoy the much deserving Thanksgiving break!