C2107 Tutorial 1 (Intro & Encryption)

School of Computing, NUS

August 12, 2025

Remark: Not all questions would be covered by instructors during tutorial. Instructors might skipped some to focus on more important questions.

1. Alice was the Web administrator of the company VIC[1]. A malicious attacker sent an email to Alice. The email instructed Alice to click on a link so as to login to the human resource (HR) system to view a report. In the email, information on the "sender" indicated it was from the HR manager in VIC. Alice wrongly believed that the email was indeed sent by the manager and followed the instructions. In doing so, she revealed her password to the attacker. Using Alice's password, the attacker logged-in to the web-server, and invoked many processes. As a result, the server was overloaded.

   With respect to the security requirements mentioned in the lecture (confidentiality, integrity, authentication, availability, etc), discussed what aspects of security were compromised.

2. Suppose it takes 512 clock cycles to test whether a 64-bit cryptographic key is correct, when given a 64-bit plaintext and the corresponding ciphertext (known plaintext attack).

   (a) How long does it take to exhaustively check all the keys using a 4 GHz (single-core) processor?

   (b) How long does it take on a cluster of 1024 servers, each with a quad-core 4GHz processor.

   (c) Bitcoin Network hash rate is the number of Tera cryptographic hashes carried out by all the bitcoin "miners" in the world in one second (so, hash rate of 1 means 1T operations per second). Find out current hash rate. Suppose one cryptographic hash is equivalent to one test of the key, how long would the Bitcoin Network take to check all the 64-bit keys?

   (Note: Let's approximate 1 year $\approx 2^{25}$ *seconds. In this course, we follow the convention where 1K* $= 2^{10}$*, 1M* $= 2^{20}$*, 1G* $= 2^{30}$*, 1T*$= 2^{40}$*)*

3. Suppose it takes 512 clock cycles to test whether a 42-bit cryptographic key is correct, when given a plaintext m and the corresponding ciphertext c.

   How long does it take to exhaustively check all the keys using a 4GHz (single-core) processor?

A walkie-talkie system *realtime Secure Walkie Talkie* (rSWT)[1] secures its communication using symmetric keys encryption. rSWT uses two encryption schemes, AES block cipher, and another fast stream cipher developed by the company called FAST[1] . The cipher FAST is really fast, but its key length is only 42 bits.

To setup a group of walkie-talkies, the user enters k, a 128-bit key, into each walkie-talkie. The key k is called the *long-term Key*. When a walkie-talkie wants to broadcast a plaintext m, which is high quality audio, to other walkie-talkies, the followings are carried out.

(a) A 128-bit v is randomly chosen.

(b) Computes $t = \text{AES}_{\text{enc}}(k, v)$, where AES_{enc} is encryption of AES block cipher (without mode of operation).

(c) Obtains $\widetilde{k}$, which is the first 42 leading bits of t. This $\widetilde{k}$ is called the *session key*.

(d) Computes $c = \text{FAST}_{\text{enc}}(\widetilde{k}, \mathbf{0}_{64} \| m)$, where $\mathbf{0}_{64}$ is a string of 64 zeros, $\|$ is string concatenation, and FAST_{enc} is the deterministic encryption of FAST. Note that c does not contain any IV (initialisation vector).

(e) Sends $(v \| c)$ over the air.

When a new message is to be sent, the above will be repeated. As such, the session key likely to be different for different messages.

The receiver extracts $\widetilde{k}$ from v using the long term key k, and then decrypts using FAST. If the leading 64 bits of the decrypted message is not all zeros, then the receiver plays an error message. Otherwise, the receiver plays the decrypted audio.

In practice, an attacker can eavesdrop signal transmitted over the air. Hence, any reasonable threat model should assume that the attackers can eavesdrop and can obtain $v \| c$.

Now, with $v \| c$ and knowledge that the leading 64 bits are zeros, can the attacker carry out exhaustive search to find the key?

We know that 42-bit is too short and thus the key can be broken. However, as calculated earlier, it would still take very long time by a laptop. In their marketing efforts, rSWT claims that 42-bit is sufficient for realtime applications. This is what appeared in their advertisement:

"*42-bit is sufficient. By the time the message is maliciously decrypted, the message becomes useless*".

In this question, you play the role of an attacker. You want to design a hand-held device that is able to crack and obtain the plaintext in *realtime*. Specifically, when given the v and c, the device should derive the 42-bit

[1] These names are fictional.

session key readily within 0.1 second. The hand-held device can have computing resource comparable to a high-end laptop.

(a) Suggest a way to get the session key, assuming that the device has huge storage, say 100TB.

(b) (optional, techniques required would be covered later) Give another method where the device has lower storage, say 1 TB.

(Hint: Use a pre-computed table.)

4. Lecture 1 mentioned that Winzip encrypts the compressed file. Why it is meaningless to carry out the two operations in the other way, that is, encrypts the file, and then compresses the encrypted file?
(Hint: Consider the effectiveness of compression on "random" sequences, and a requirement of encryption scheme.)

5. Bob encrypted a music mp3 file using Winzip, which employs the 256-bit key AES. He chose a 6-digit number as password. Winzip generated the 256-bit AES key from the 6-digit password using a (deterministic) function, say SHA1[2].

   Alice obtained the ciphertext. Alice also knew that Bob used a 6-digit password and knew how Winzip generated the AES key.

   (a) Given a 256-bit string, can Alice determine whether this string was indeed the correct AES key?

   (b) How many guesses did Alice really need in order to get the mp3 file?

6. Find out more about these terminologies and well-known persons in cryptography

   (a) *NSA, NIST, cryptography backdoor, Decryption order* ;

   (b) *Edward Snowden.*

[2]We haven't introduce SHA1 yet. Here, just treat it as some routine that take in 6-digit as input, scramble them and output a 256-bit string.

Hands-on Exercise: Linux Set-Up[3]

Are you aware that we can use our laptop to simulate another laptop? Yes, using Virtual Machine (VM).

For CS2107 assignment, a Linux system would be useful. If you are using Mac or Window, don't buy a new laptop for that. Set up a **Linux host** using VM (Virtual Machine) using your current laptop. While it could be possible to complete CS2107 assignment without using VM, the experience could be useful for other modules.

There are different versions of Linux. An **Ubuntu desktop** is recommended since it is user friendly enough even for new users. If you are serious about security, at some point you way need Kali Linux which come with many security tools. Important to note that many tools in Kali are offensive tools and illegal if not handled properly (test those tools on your own system).

1. **Window.** For Windows, you can use either **VirtualBox** or **VMWare**. Instructions on installation should be easily found. Here is one:
   `https://medium.com/@brianmwambia3/a-step-by-step-guide-to-setting-up-windows-10-virtual-machine-on-virtualbox-945b4f321d61`

2. **Mac.** VirtualBox only compatible with intel-based Mac. For Silicon-based MacOS, try VMware Fusion (free for personal use). Need to register with Broadcom. see below for installation of VMware Fusion.
   `https://blogs.vmware.com/teamfusion/2024/05/fusion-pro-now-available-free-for-personal-use.html`

Use the "NAT" or "bridge adapter" connection/networking mode for your VM, so that it can access the Internet.

It is also expected that you have rudimentary proficiency in using a Linux system. Here is a beginner guide from Ubuntu.
`https://ubuntu.com/tutorials/command-line-for-beginners#1-overview`

However, more knowledge might be needed, and it is expected that you do some self-exploration. You may thus want to refer to this freely-downloadable good book on Linux: "**The Linux Command Line**":
`https://linuxcommand.org/tlcl.php`

If we indeed require Linux for the assignment, there would be open consultation session after assignment is released. Nonetheless, do self exploration now in setting up your Linux system. Setting up test environment in fact is a "skillset" required for security professionals.

[3]Will not discuss during tutorial