# C2107 Tutorial 1 (Intro & Encryption)

School of Computing, NUS

September 1, 2025

1. The violated security aspects and offending actions are as follows.

   **Confidentiality:**

   **A3.** Alice revealed her password.

   **Authenticity:**

   **A1.** The attacker spoofed the email.

   **A2.** Alice visited and interacted with the spoofed website specified in the link.

   **A4.** The attacker logged-in to the Web server.

   **Availability:**

   **A6.** The Web server got overloaded.

   **Integrity:**

   **A5.** The attacker invoked many processes on the Web server (*Remark*: a violation of the server's process integrity).

2. (a) Notice that a 4GHz processor has $2^2 \cdot 2^{30} = 2^{32}$ cycles per second.
   From the problem description, testing 1 key takes $512 = 2^9$ cycles.
   In 1 second, the processor can thus check $2^{32} \,/\, 2^9 = 2^{23}$ keys.
   To check all $2^{64}$ keys, the processor needs $2^{64} \,/\, 2^{23} = 2^{41}$ seconds.
   Since 1 year $\approx 2^{25}$ seconds, the total time needed is therefore: $2^{41} \,/\, 2^{25} \approx 2^{16} \approx 2^6 \cdot 2^{10}$ years $\approx$ 64K years.

   (b) Given 1024 servers, each with a quad-core processor, we thus have $1024 \cdot 4 = 2^{10} \cdot 2^2 = 2^{12}$ processors.
   The total time needed is now reduced by a factor of $2^{12}$ to become: $\approx 2^{16} \,/\, 2^{12} \approx 2^4 \approx 16$ years.

   (c) Current hashrate (Aug 2025) approx 900EH/s. Let's take it as 1024 EH/s = 1 ZH/s. That is, $2^{70}$ hashes per second. So, can break in $2^{64-70} = 2^{-6}$ seconds. What about 128-bit keys? For 128-bit keys it would take extremely long time. Even if we significantly increase the hashrate by a large factor, say 1,000,000 times, which could be more than total current compute power in the world. Thus, it was widely accepted that exhaustively searching 128-bit keys is not

feasible. Note that NIST suggested 128-bit for AES. Nonetheless, there is a complication due to quantum computer. It turns out that one could speed up any exhaustive search by a square root factor using quantum computer (we still don't have quantum computer with sufficient "qbits"). Hence, post-quantum requirement is 256 bit for AES.

3. (a) Notice that a 4GHz processor has $2^2 \cdot 2^{30} = 2^{32}$ cycles per second.
From the problem description, testing 1 key takes $512 = 2^9$ cycles.
In 1 second, the processor can thus check $2^{32} \,/\, 2^9 = 2^{23}$ keys.
To check all $2^{42}$ keys, the processor needs $2^{42} \,/\, 2^{23} = 2^{19}$ seconds.
Approx 145 hours.
The plaintext thus cannot be recovered in realtime.

   (b) Tradeoff the time with space, i.e. solving the problem in less time by using more storage space, as follows:
Construct a table of $\text{FAST}_{\text{enc}}(\widetilde{k}, 000\ldots000)$ for all possible $2^{42}$ values of $\widetilde{k}$. Notice that there are $2^{42}$ entries in the table, where each entry is 64-bit long as the result of encrypting the 64-bit of zeros. Hence, the derived table will take $2^{42} \cdot 64$ bits $= 2^{42} \cdot 8$ bytes $=$ 32TB.
To break a SWT communication, first extract the first 64 bits of the captured ciphertext, then perform a lookup operation on the constructed table in order to determine the employed $\widetilde{k}$.
Given $\widetilde{k}$, the attacker can perform a decryption process using the stream cipher, thus recovering the plaintext in realtime.

   Remark:

   i. We still need a data-structure to lookup the key. One method is to employ hash-table that can have "constant-time" look up. We omit the details (algorithm & data-structure course).
   ii. There is another technique know as *time-space tradeoff* to further tradeoff time with space. Yet another technique *Rainbow table* further improve the efficiency.

4. Compressing an encrypted file will yield very little or no compression gain. This is since the encrypted file will resemble a "random" sequence (due to a requirement of a good encryption scheme).
A compression algorithm, which takes advantage of repeating patterns, therefore will not work well on an encrypted file.

5. (a) Yes. Check whether the decrypted plaintext follows mp3 format. (b) The total number of guesses needed is only $10^6 = 1$ million.
(Note: Why not $2^{256}$?)