Bahan kuliah II4020 Kriptografi

07 - Serangan Terhadap Kriptografi

Oleh: Rinaldi M

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2026

Pendahuluan

- Keseluruhan *point* dari kriptografi adalah menjaga kerahasiaan pesan atau kunci dari penyadap (*eavesdropper*) atau dari kriptanalis (*cryptanalyst*).

- Kriptanalis dapat pula merangkap sebagai seorang penyadap

- Kriptanalis berusaha memecahkan cipherteks dengan melakukan serangan terhadap sistem kriptografi.

- Tujuan serangan adalah untuk mengungkap plainteks dari cipherteks atau mendapatkan kunci dekripsi.

Serangan (*attack*)

- **Serangan** diartikan sebagai setiap usaha (*attempt*) atau percobaan yang dilakukan oleh kriptanalis untuk menemukan kunci atau menemukan plainteks dari cipherteksnya.
- Asumsi: kriptanalis mengetahui algoritma kriptografi yang digunakan

Prinsip Kerckhoff: Semua algoritma kriptografi harus publik; hanya kunci yang rahasia.

Jadi,keamanan sistem kriptografi terletak pada kunci!

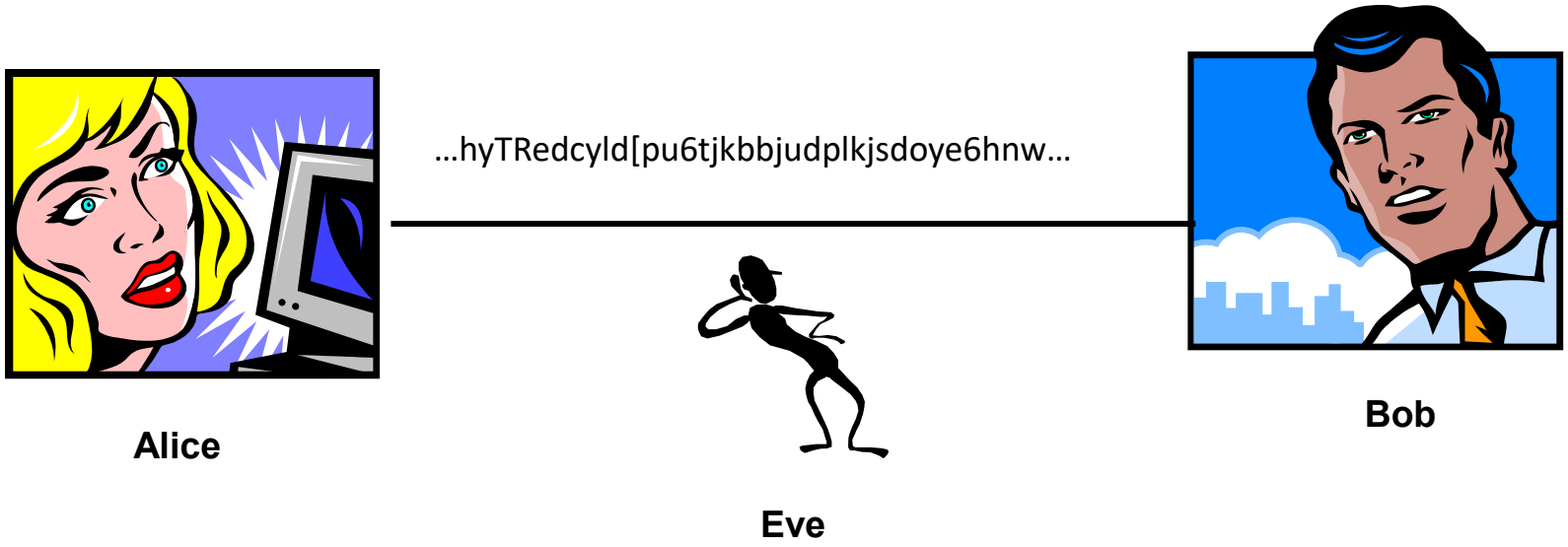
Jenis-jenis Serangan

Berdasarkan keterlibatan penyerang dalam komunikasi pesan**:**

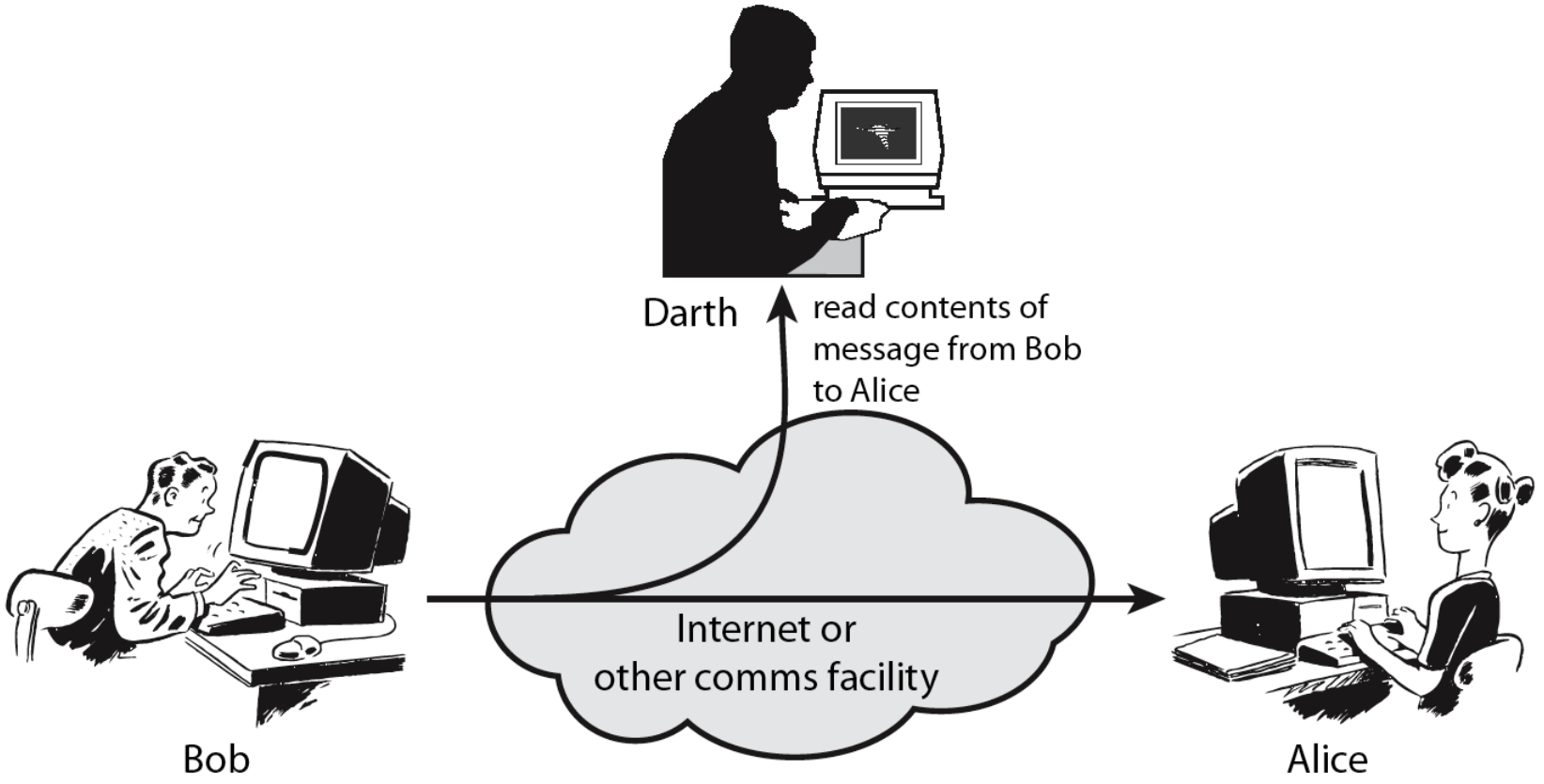
1. **Serangan pasif**
2. **Serangan aktif**

1. Serangan pasif (*passive attack*)

- penyerang tidak terlibat dalam komunikasi antara pengirim dan penerima
- penyerang hanya melakukan penyadapan untuk memperoleh data atau informasi sebanyak-banyaknya
- enkripsi pesan mencegah penyadap memahami isi pesan

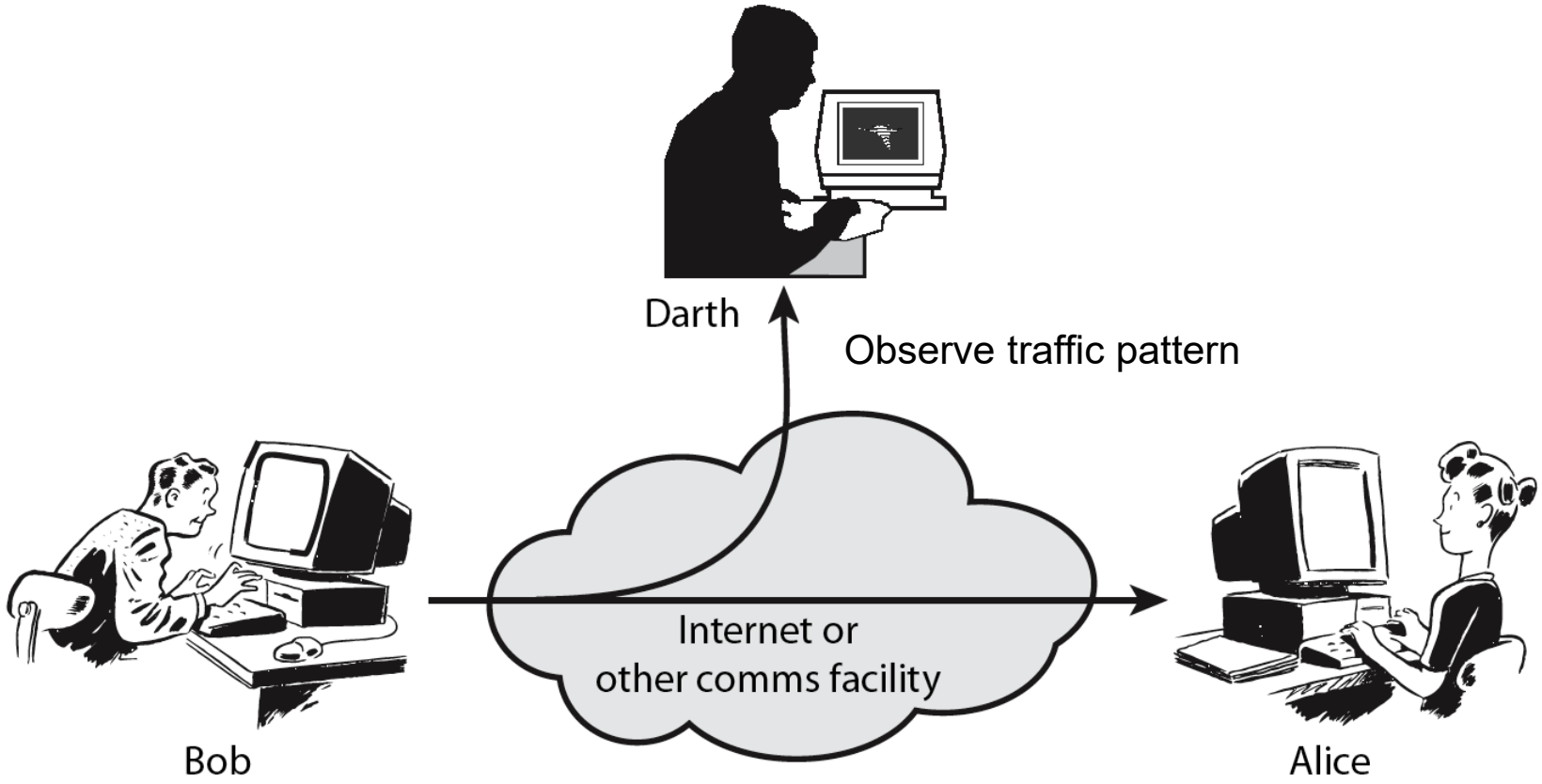


Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



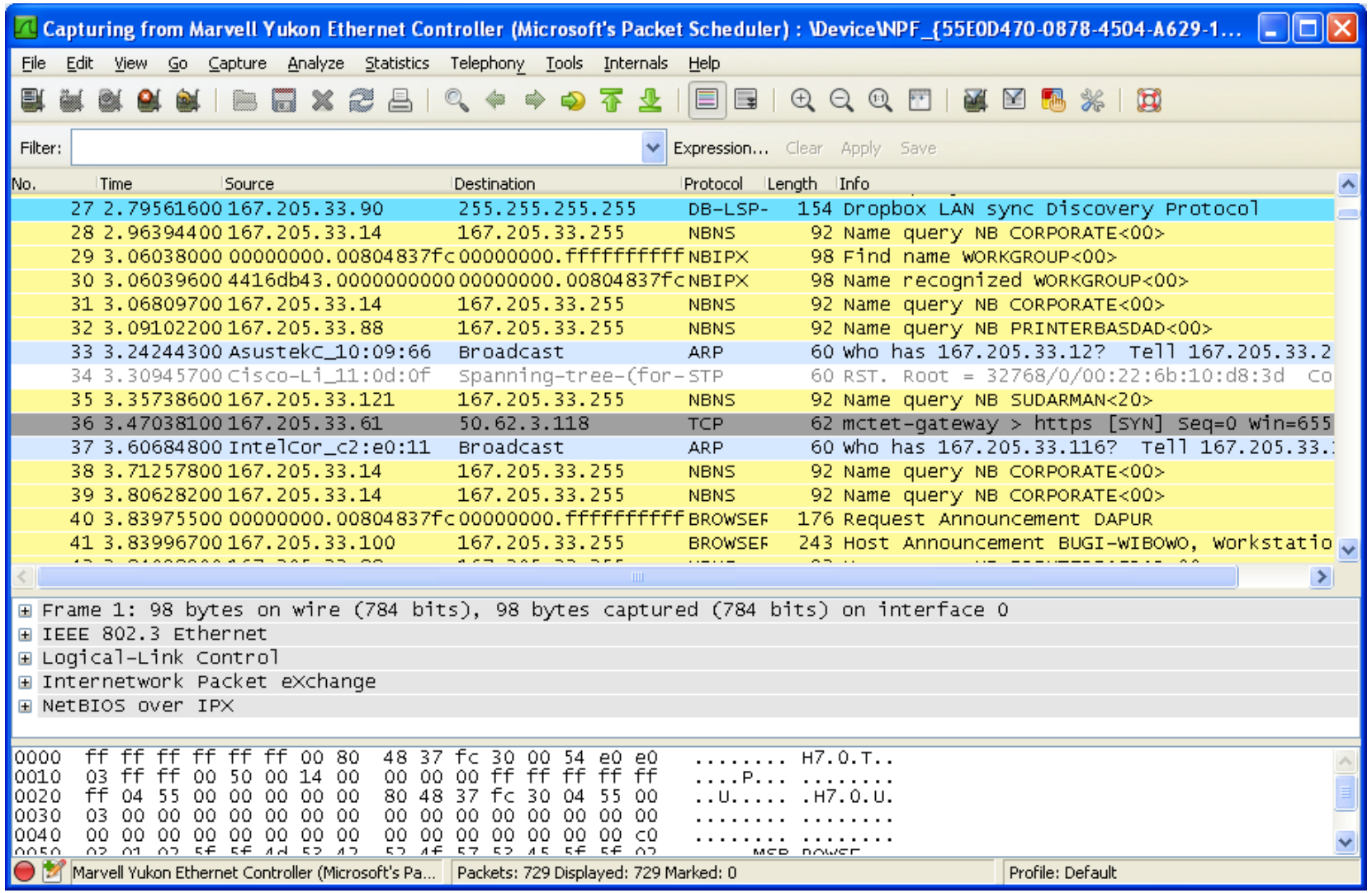
Passive Attack : Interception

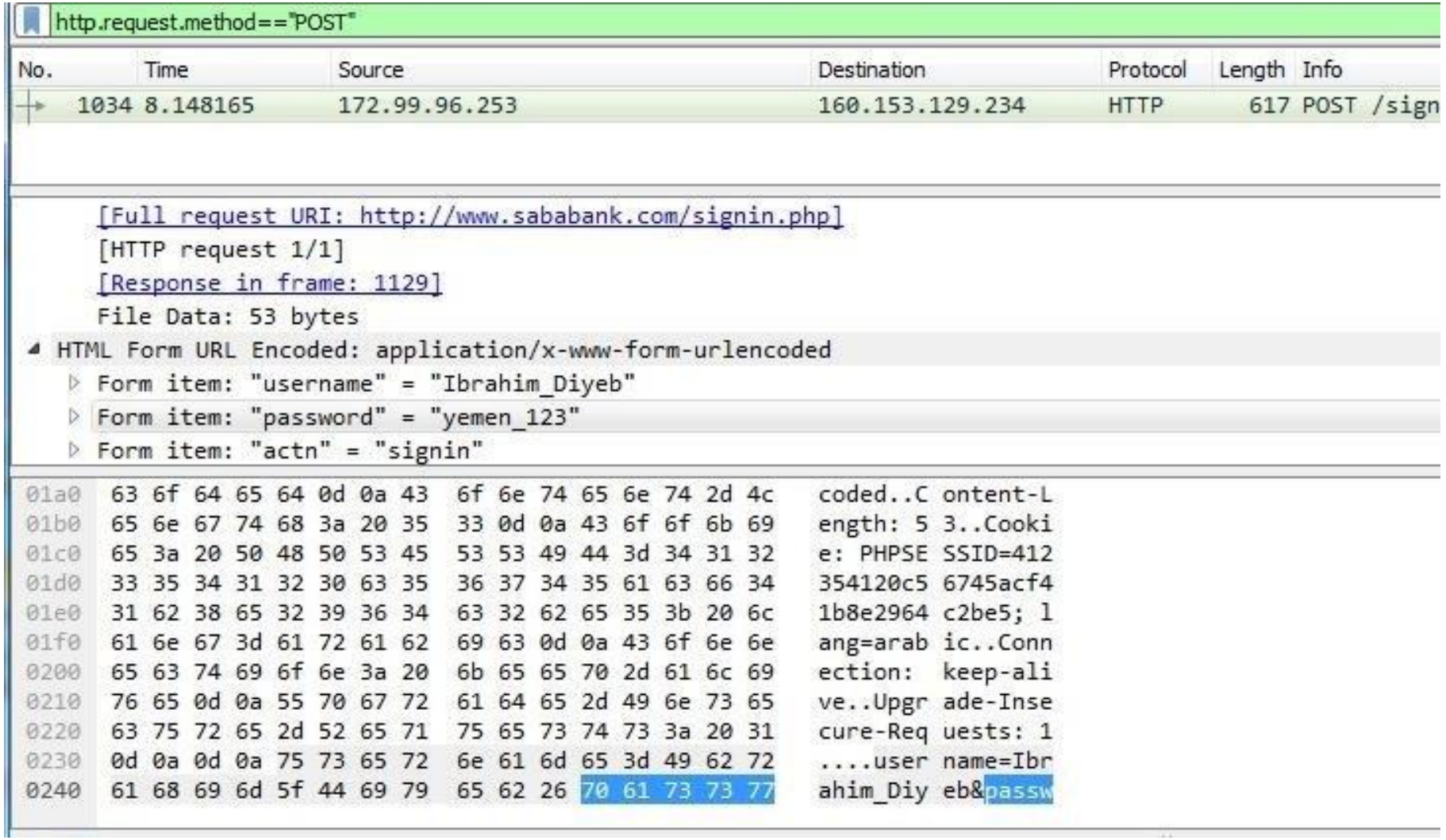
Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



Passive Attack : Traffic Analysis

Screenshot Wireshark (memantau *network traffic*)

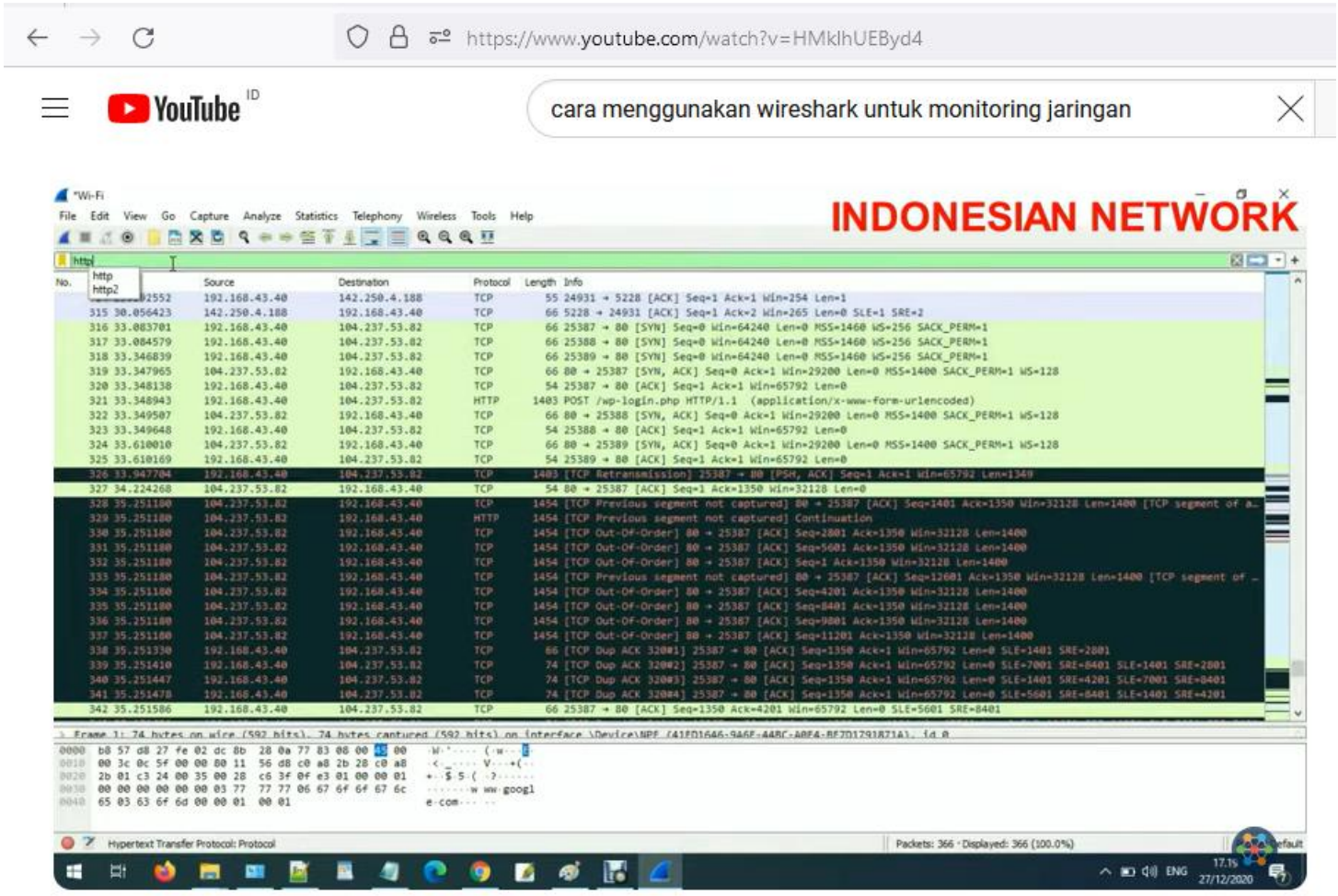




Filtering dengan Wireshark dapat menampilkan plainteks berupa *username* dan *password*

Sumber gambar: https://www.researchgate.net/figure/Wireshark-Filtering-Showing-Clear-Text-of-user-Name-and-Password_fig3_326419957

Cek video TUTORIAL CARA MENGGUNAKAN WIRESHARK | MENEMUKAN U53RN4M3 P455W0RD PADA LALU LINTAS PACKET JARINGAN: https://www.youtube.com/watch?v=HMklhUEByd4

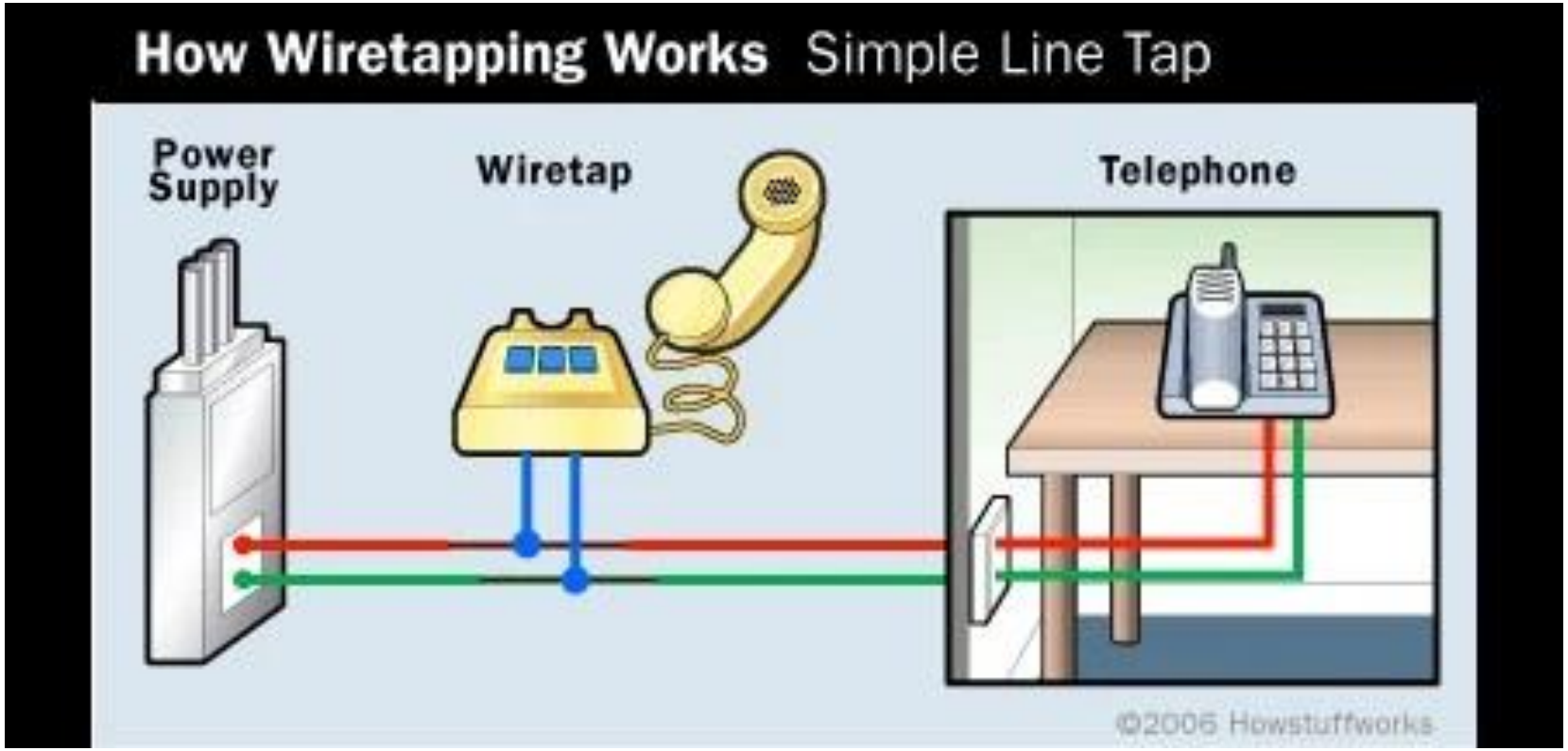


TUTORIAL CARA MENGGUNAKAN WIRESHARK | MENEMUKAN U53RN4M3 P455W0RD PADA LALU LINTAS PACKET JARINGAN

Metode penyadapan:

1. *Wiretapping*
2. *Electromagnetic eavesdropping*
3. *Acoustic Eavesdropping*

- *Wiretapping*

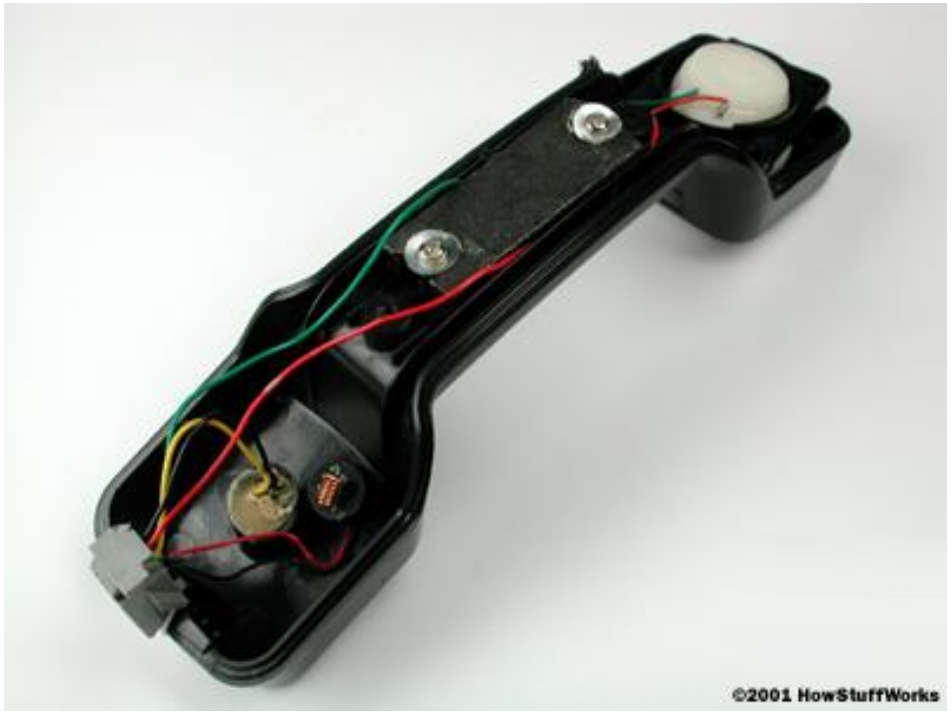


Baca di sini:

1. https://hackaday.com/2008/06/19/wiretapping-and-how-to-avoid-it/
2. https://edition.cnn.com/videos/us/2017/03/07/what-is-wiretapping-jpm-orig.cnn

How Wiretapping Works

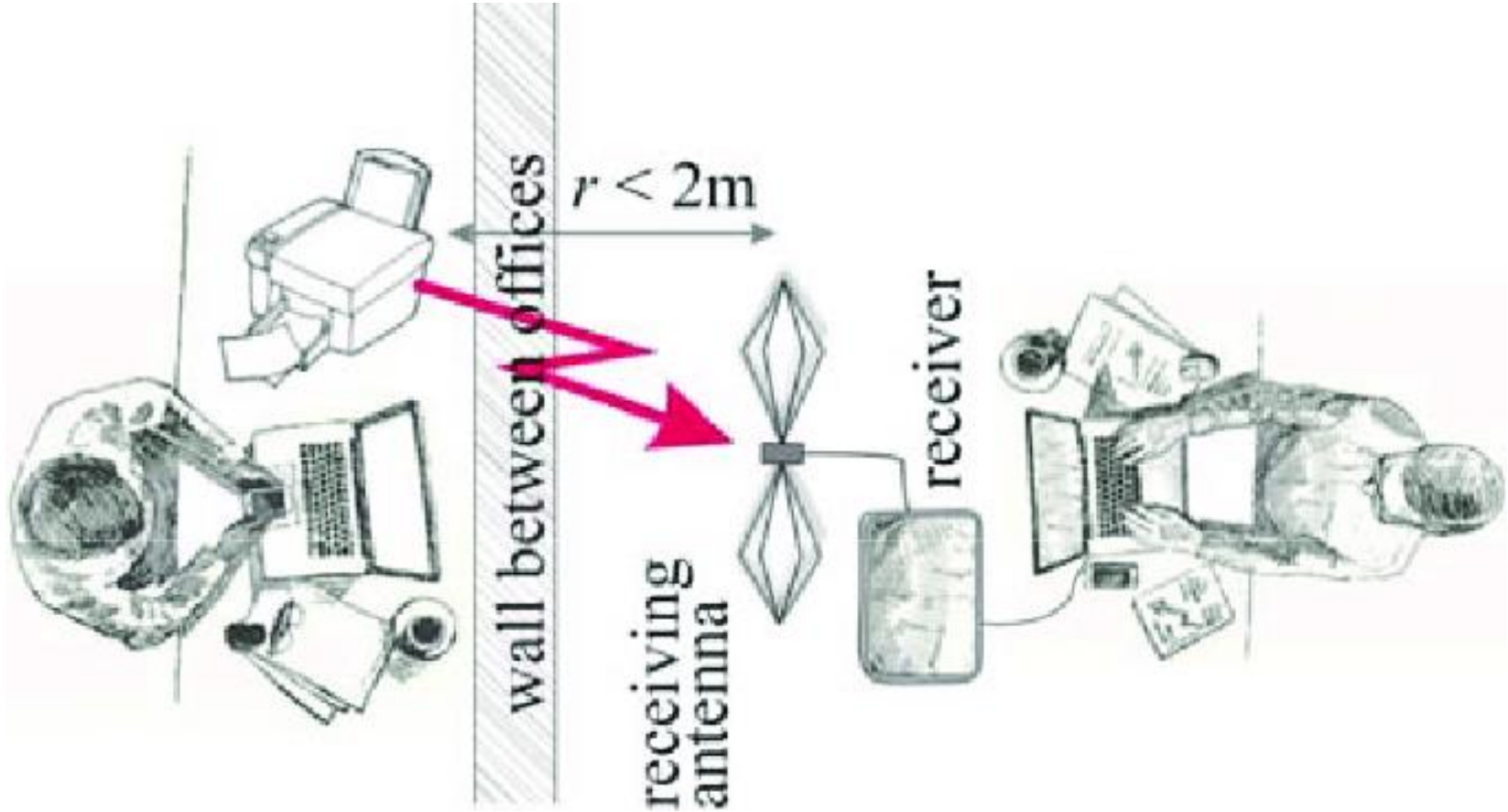
(sumber: http://www.howstuffworks.com/wiretapping.htm)



When you open up a phone, you can see that the technology inside is very simple. The simplicity of design makes the phone system vulnerable to surreptitious eavesdropping.

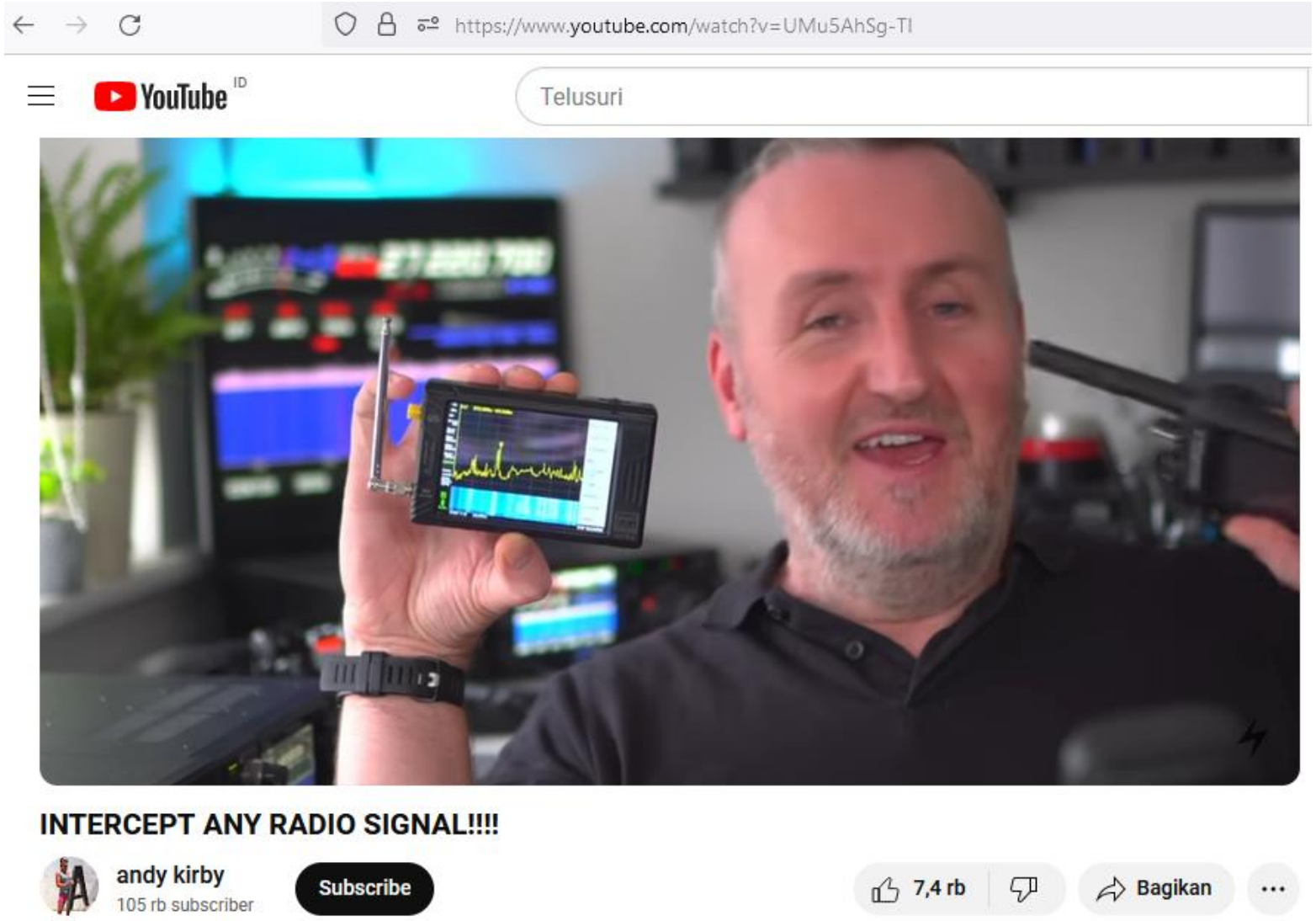
Inside a standard phone cord, you'll find a red wire and a green wire. These wires form a circuit like the one you might find in a flashlight. Just as in a flashlight, there is a negatively-charged end and a positively-charged end to the circuit. In a telephone cord, the green wire connects to the positive end and the red cord connects to the negative end.

Electromagnetic eavesdropping

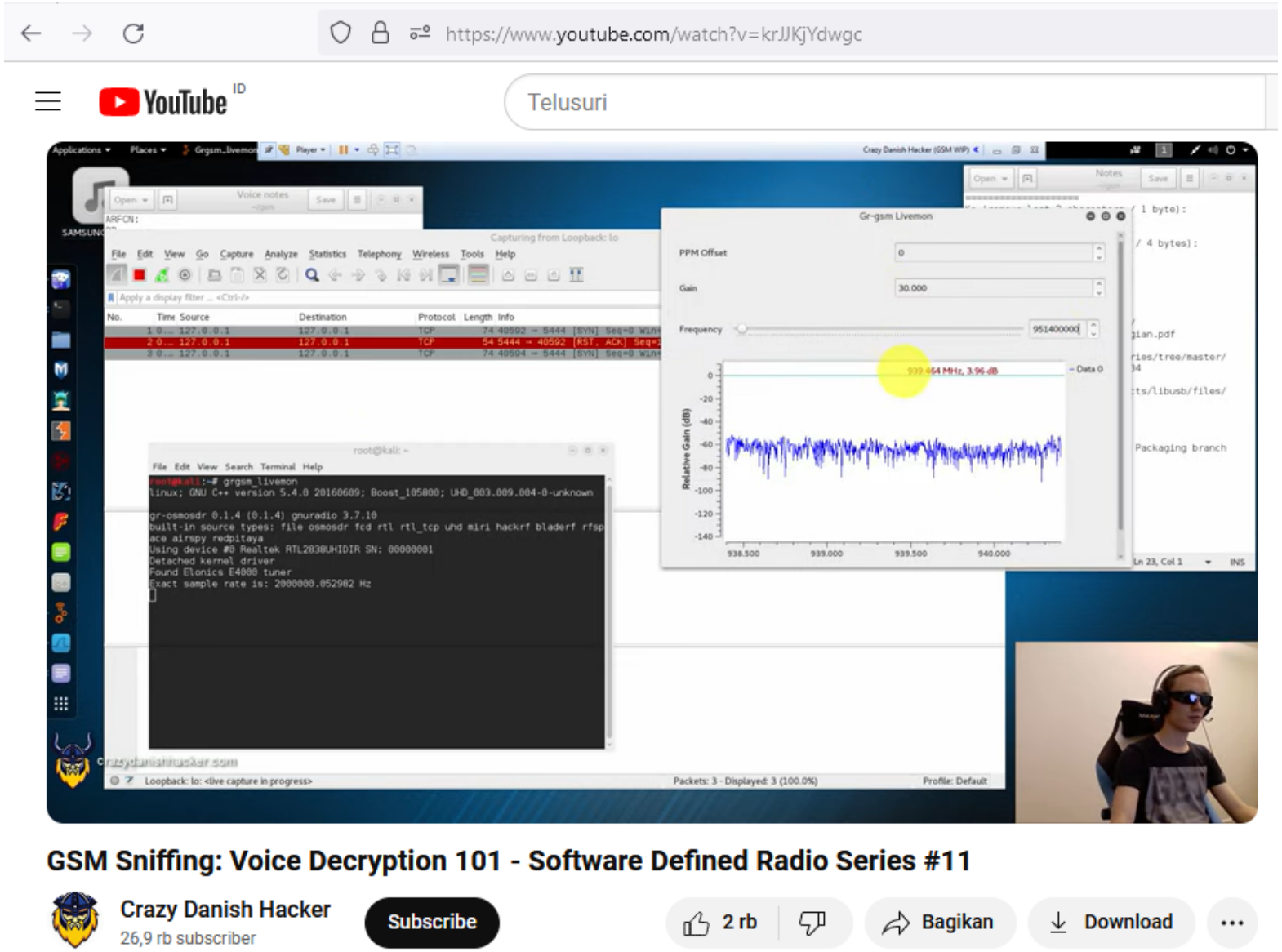


Sumber: https://www.researchgate.net/figure/An-anechoic-chamber-Figure-4-Example-for-an-electromagnetic-eavesdropping-process_fig3_324680618

Cek video **INTERCEPT ANY RADIO SIGNAL!!!!:** https://www.youtube.com/watch?v=UMu5AhSg-TI



Cek video GSM Sniffing: Voice Decryption 101 - Software Defined Radio Series #11:
https://www.youtube.com/watch?v=krJJKjYdwgc

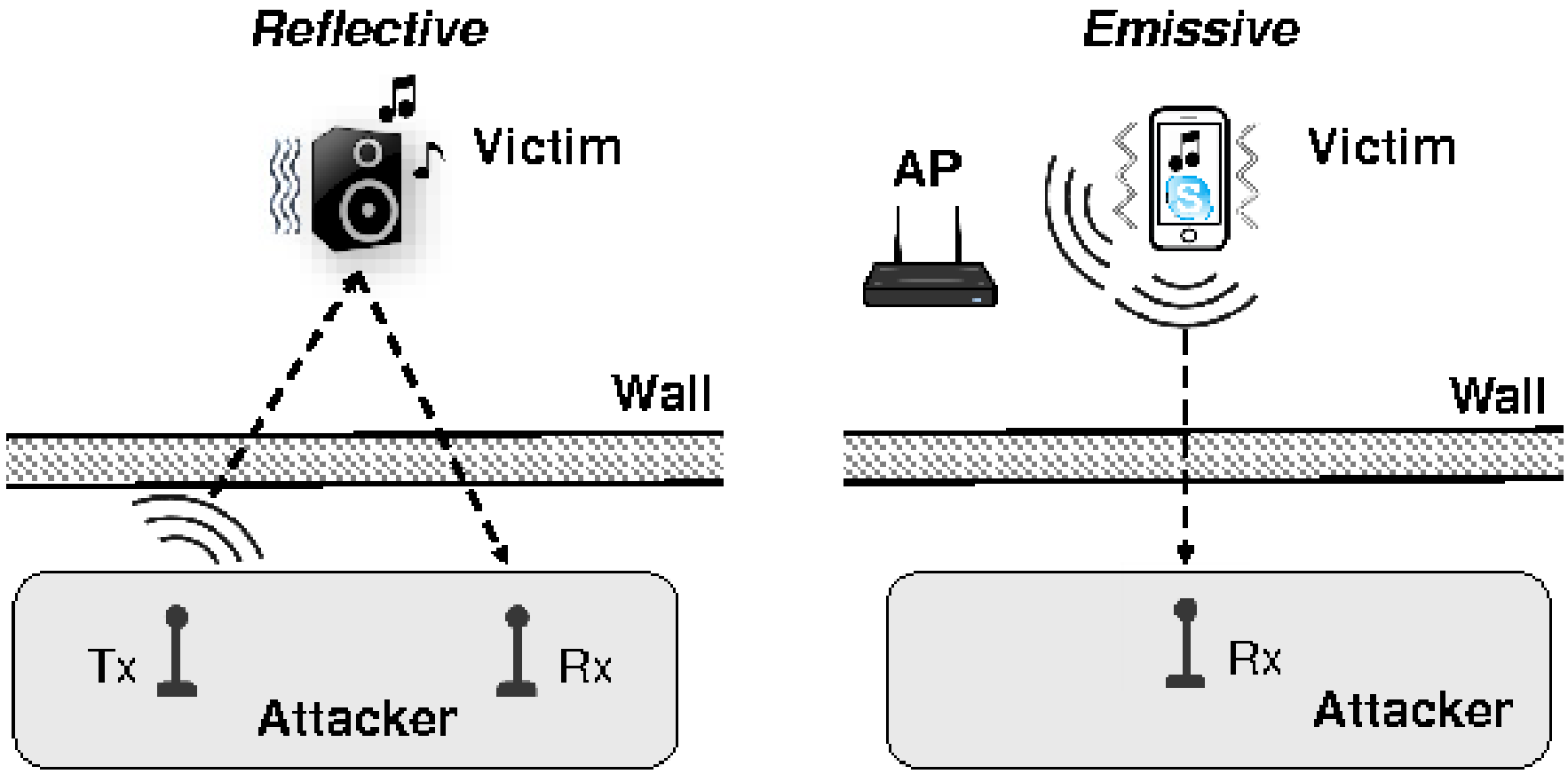


Acoustic Eavesdropping

15506-41DG
'Office: 9am' Disc
© JupiterImages

Creatas

www.comstock.com



Sumber: https://www.semanticscholar.org/paper/Acoustic-Eavesdropping-through-Wireless-Vibrometry-Wei-Wang/8afd80726c54ed7b95d30d1230bef633d128c930

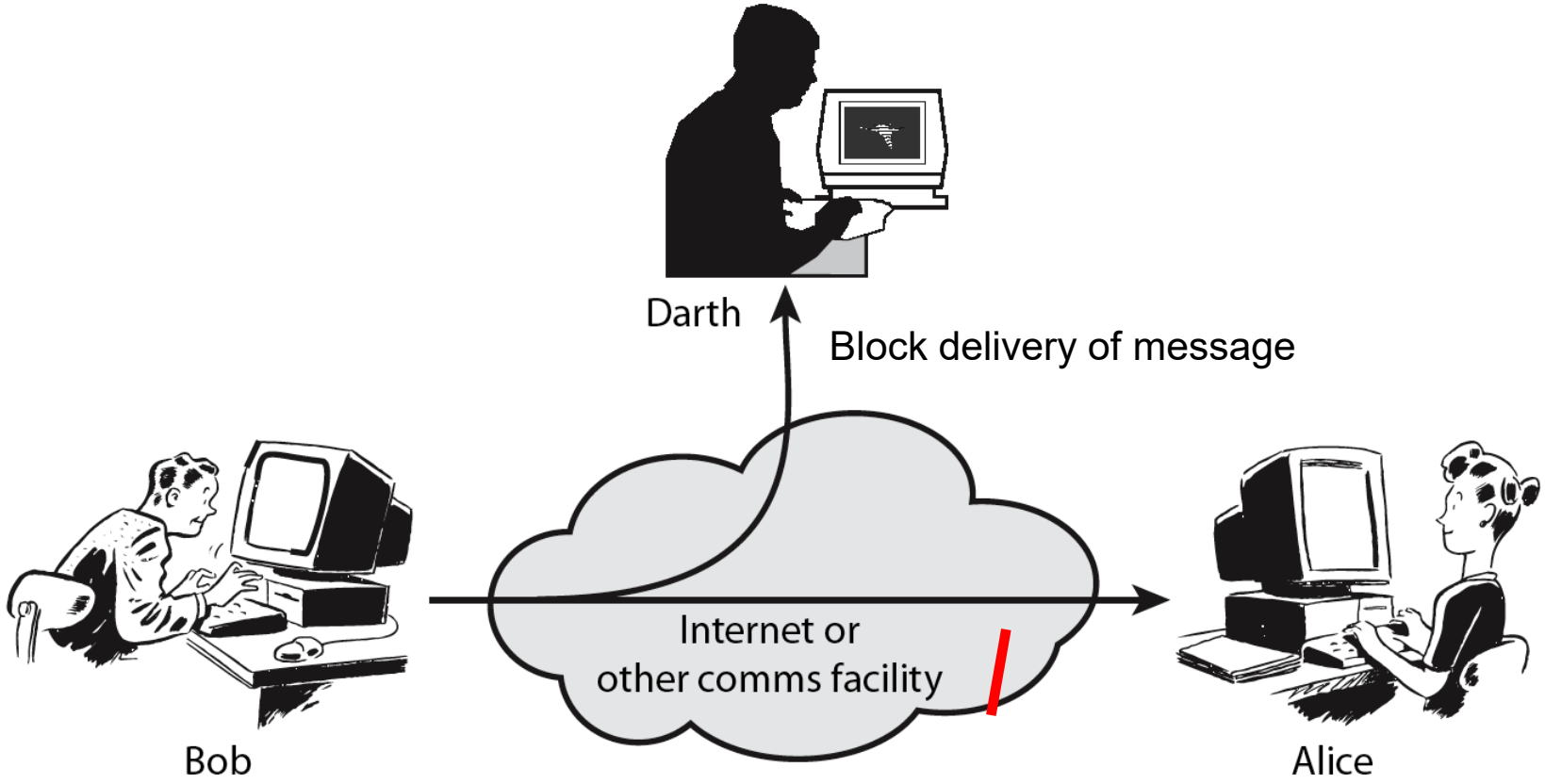
2. Serangan aktif (*active attack*)

- penyerang mengintervensi komunikasi dan ikut mempengaruhi sistem untuk keuntungan dirinya

- penyerang dapat mengubah aliran pesan seperti:
  - menghapus sebagian cipherteks,
  - mengubah cipherteks,
  - menyisipkan potongan cipherteks palsu,
  - me-*replay* pesan lama,
  - mengubah informasi yang tersimpan, dsb

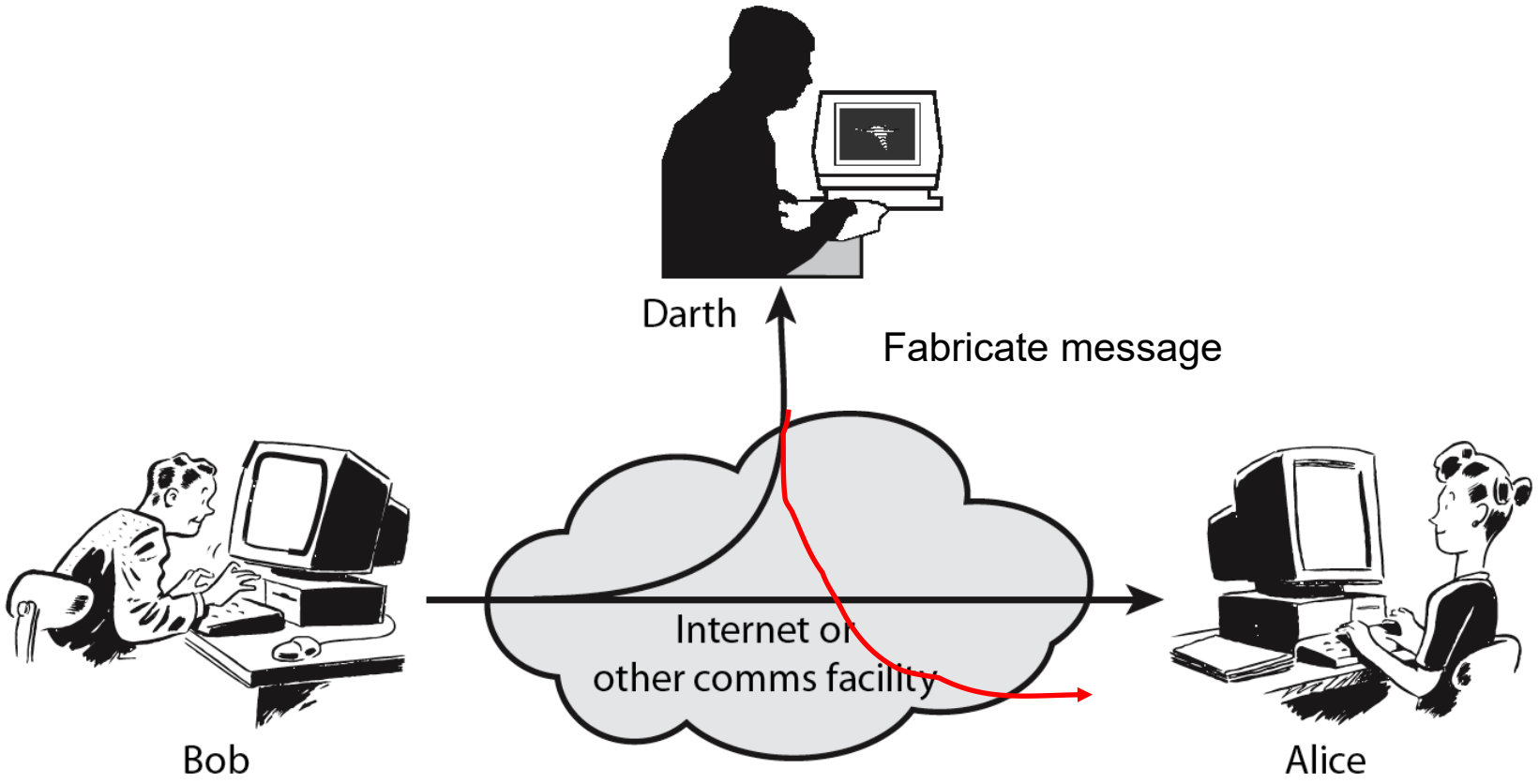
- Contoh: *man-in-the-middle attack*

Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



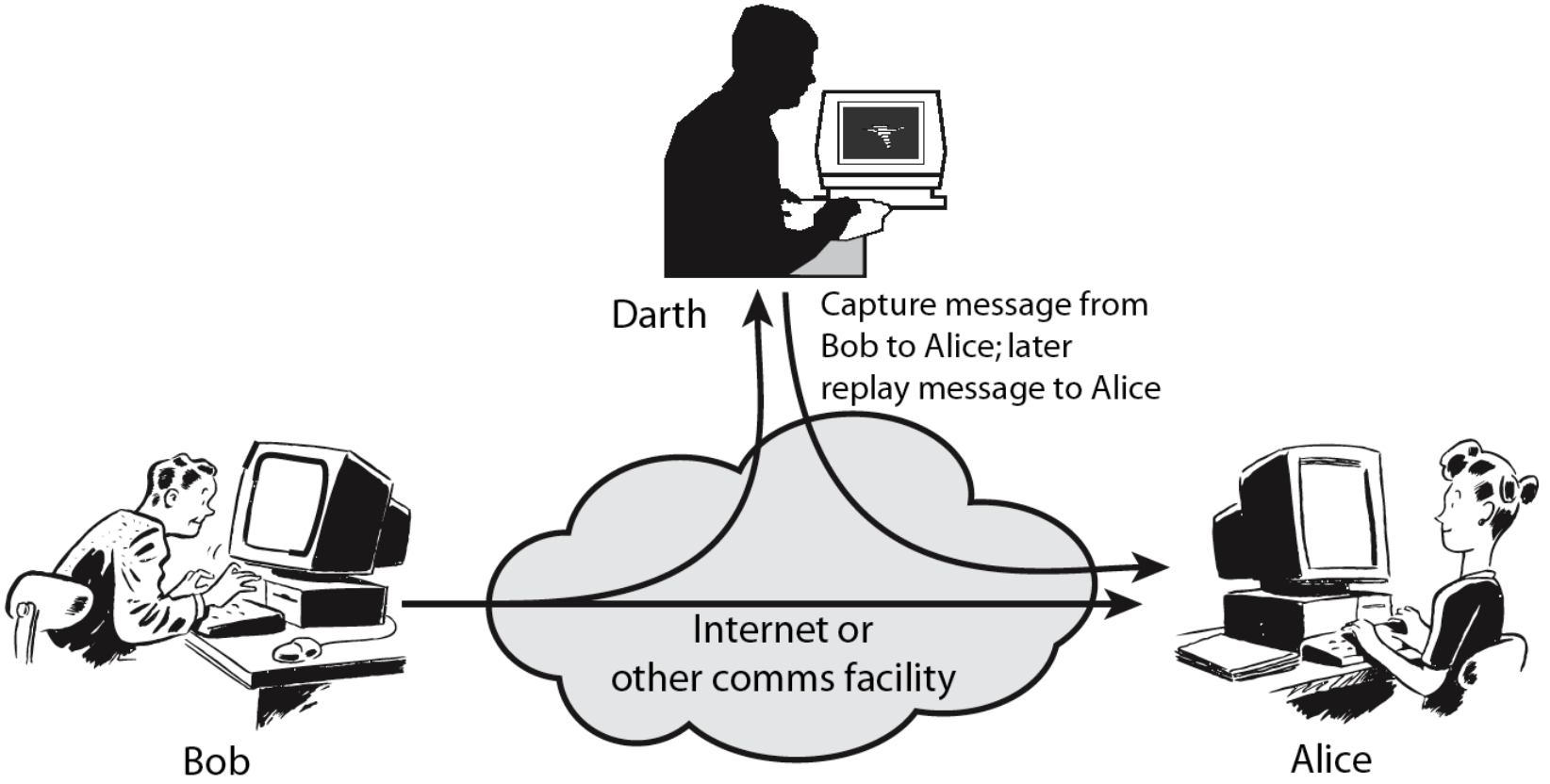
Active Attack: Interruption

Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



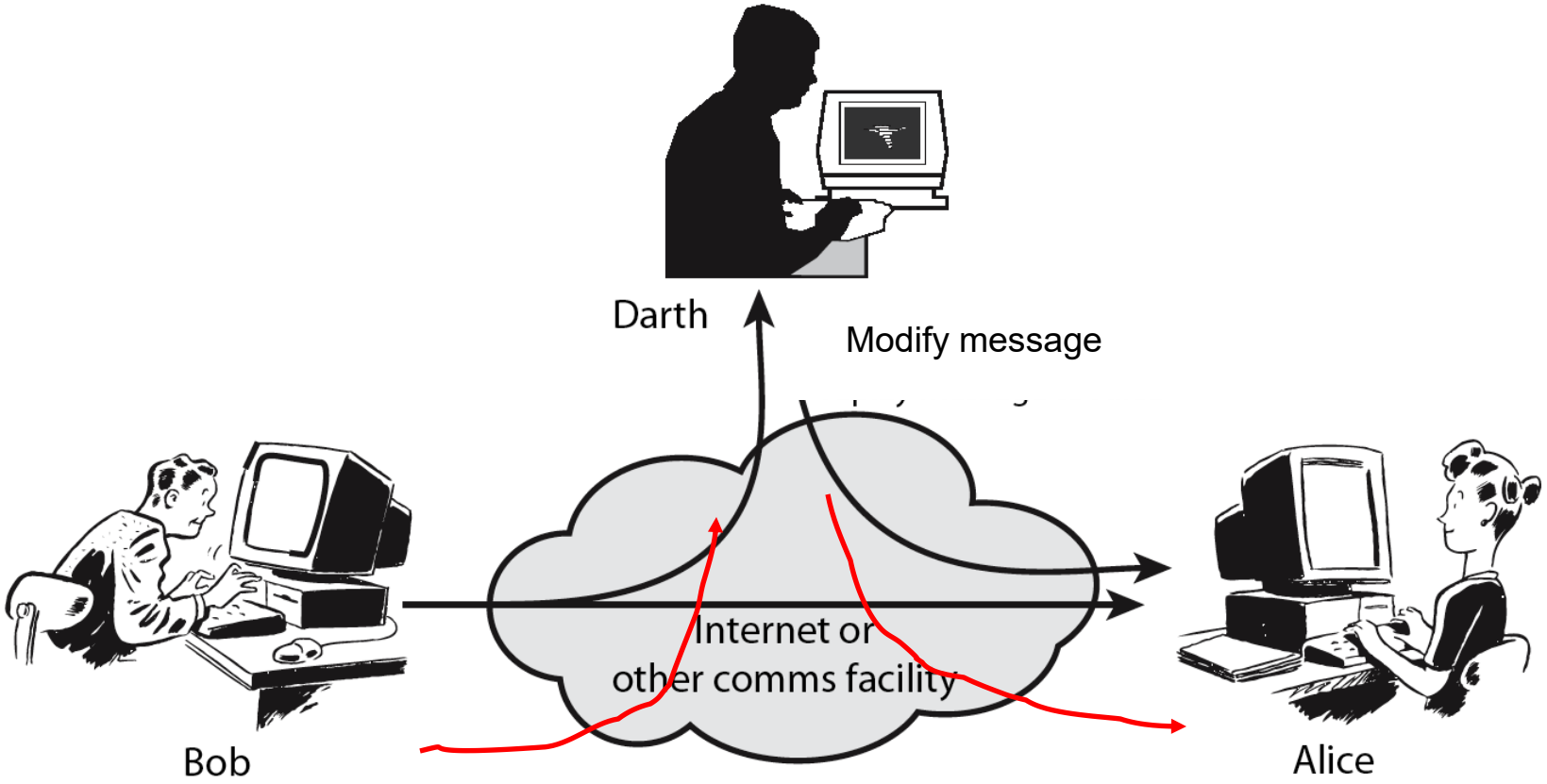
Active Attack: Fabrication

Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



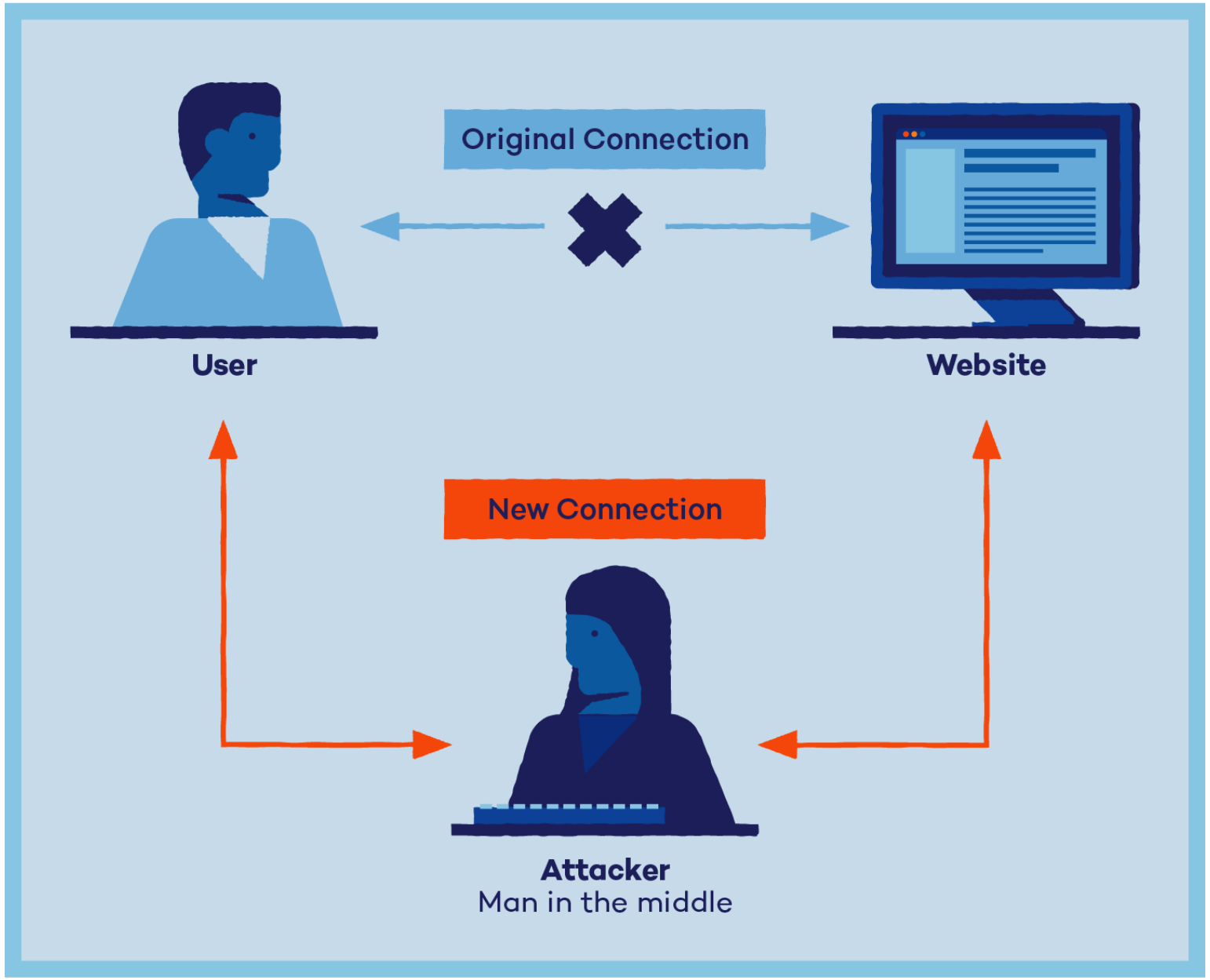
Active Attack: Replay

Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1

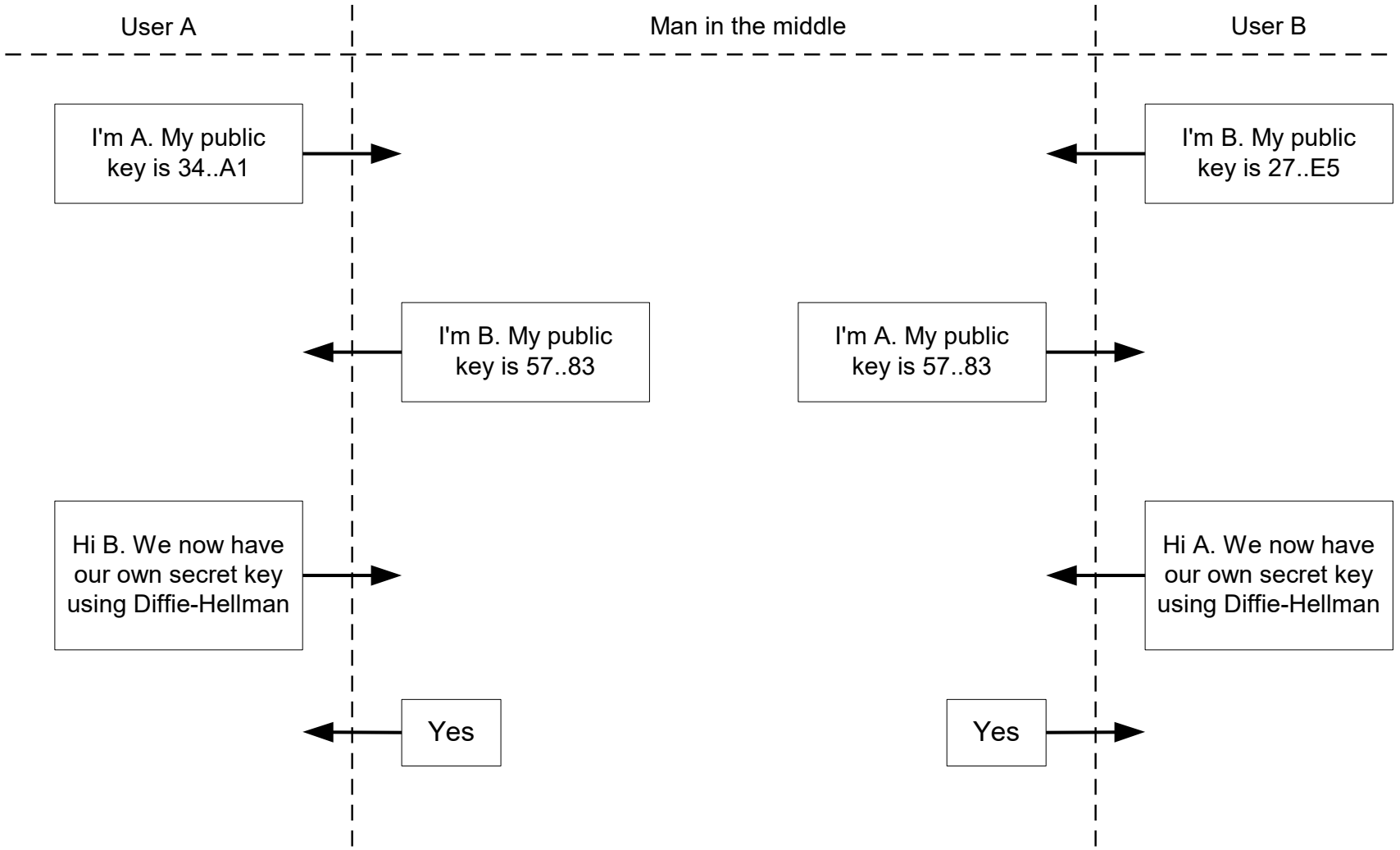


Active Attack: Modification

Man-in-the-middle-attack

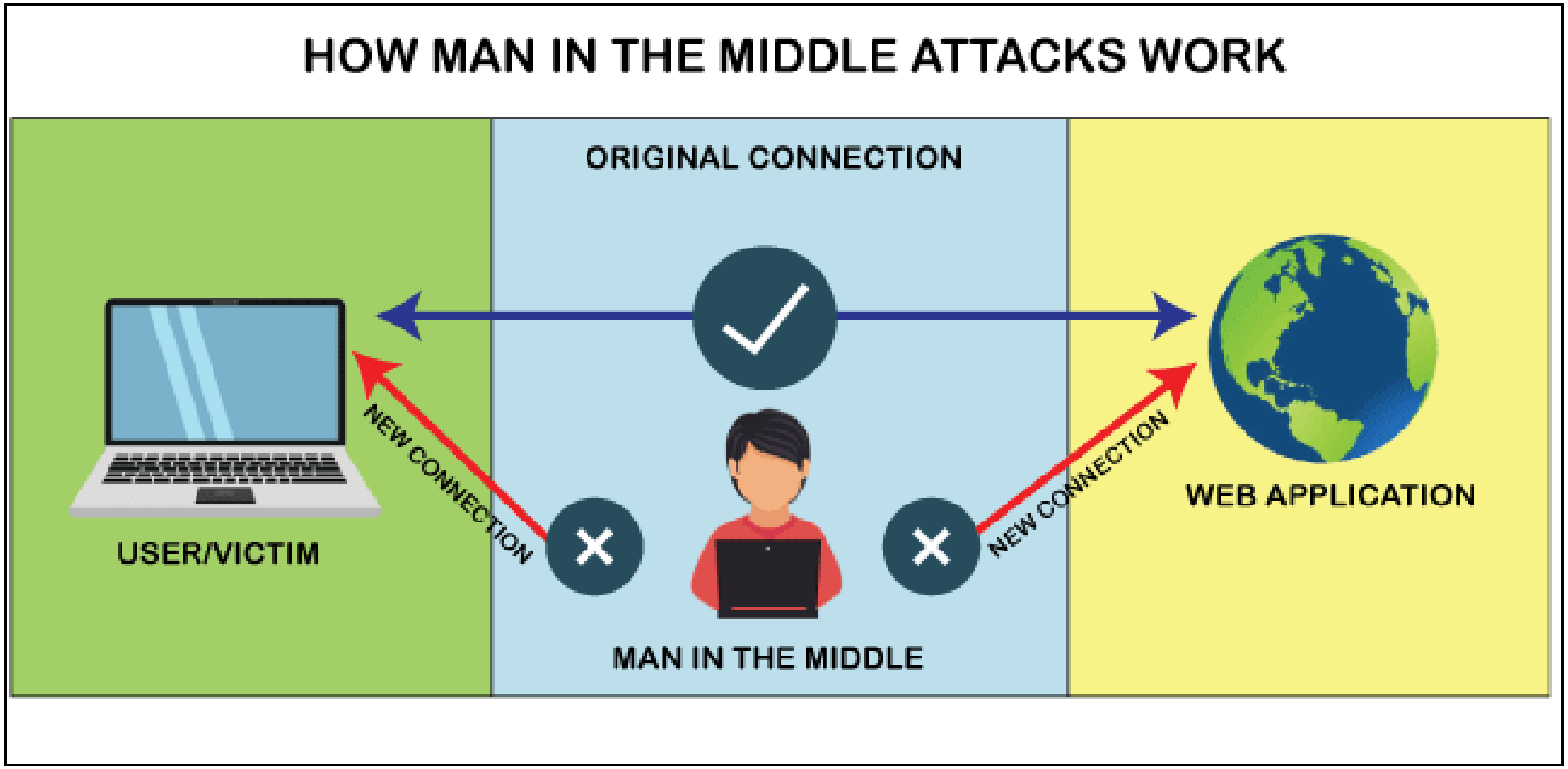


Man-in-the-middle-attack



Man-in-the-middle-attack

Serangan aktif yang berbahaya



Jenis-jenis Serangan

Berdasarkan teknik yang digunakan untuk menemukan kunci:

1. *Exhaustive attack/brute force attack*
2. *Analytical attack*

1. Exhaustive attack /brute force attack

- Mengungkap plainteks dengan mencoba semua kemungkinan kunci untuk dekripsi cipherteks

. Contoh: *dictionary attack*

- Pasti berhasil menemukan kunci jika diberikan waktu yang cukup dan sumberdaya *hardware* dan *software* yang memenuhi.

Tabel 1 Waktu yang diperlukan untuk *exhaustive key search*
(Sumber: William Stallings, *Data and Computer Communication Fourth Edition*)

Ukuran kunci	Jumlah kemungkinan kunci	Lama waktu untuk 10^6 percobaan per detik	Lama waktu untuk 10^{12} percobaan per detik
16 bit	$2^{16} = 65536$	32.7 milidetik	0.0327 mikrodetik
32 bit	$2^{32} = 4.3 \times 10^9$	35.8 menit	2.15 milidetik
56 bit	$2^{56} = 7.2 \times 10^{16}$	1142 tahun	10.01 jam
128 bit	$2^{128} = 4.3 \times 10^{38}$	5.4×10^{24} tahun	5.4×10^{18} tahun

Solusi: Kriptografer harus membuat kunci yang panjang dan tidak mudah ditebak.

2. *Analytical attack*

- Menganalisis kelemahan cipher secara matematik untuk menemukan parameter kunci, atau untuk mengurangi kemungkinan kunci yang tidak mungkin ada.

- Caranya: memecahkan persamaan-persamaan matematika (yang diperoleh dari konsep yang digunakan di dalam ciphernya) yang mengandung peubah-peubah yang merepresentasikan plainteks atau kunci.

- Contoh: Lihat kembali *Affine Cipher* → Enkripsi: $C \equiv mP + b \pmod{n}$
  Dekripsi: $P \equiv m^{-1}(C - b) \pmod{n}$
  Kunci: m dan b

  m bilangan bulat yang relatif prima dengan n
  b adalah jumlah pergeseran

  m dan b dapat ditemukan dengan memecahkan dua buah persamaan linier yang memuat peubah m dan b, asalkan diketahui dua pasang plainteks dan cipherteks yang berkoresponden.

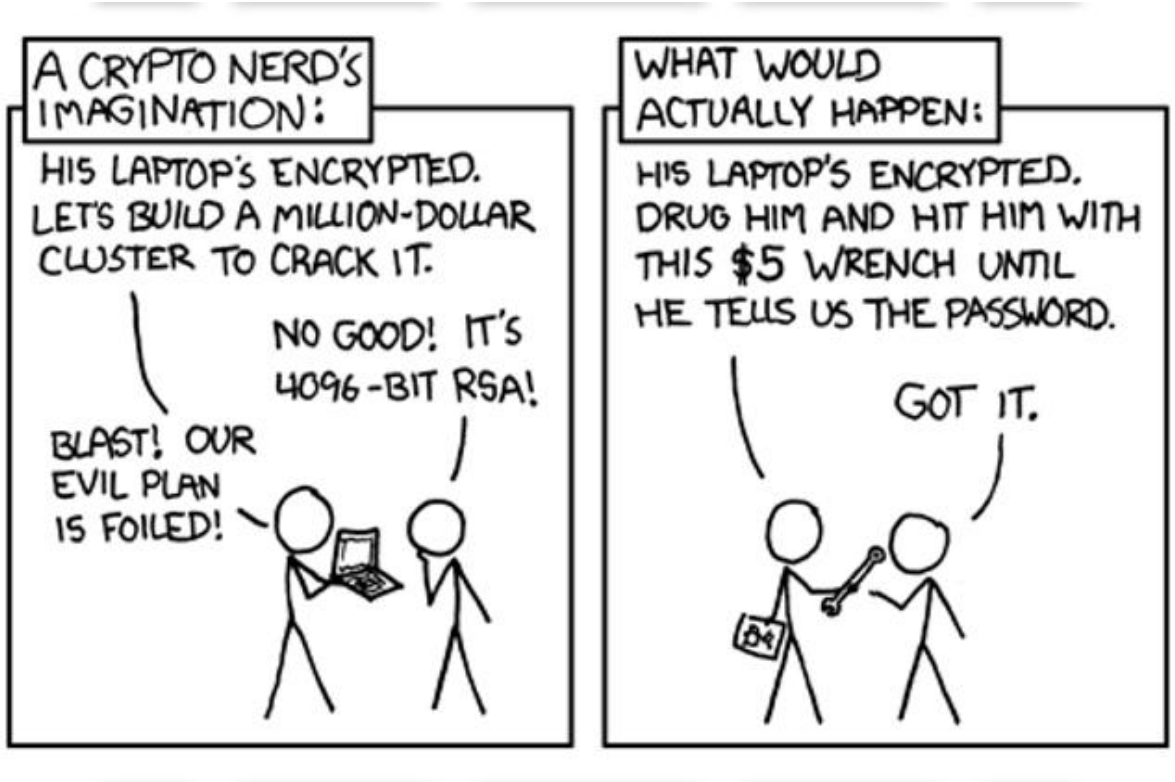
- Metode *analytical attack* biasanya lebih cepat menemukan kunci dibandingkan dengan *exhaustive attack*.

- Solusi: kriptografer harus membuat algoritma kriptografi yang sekompleks mungkin sehingga lebih sukar dianalisis

Jenis-jenis Serangan

- Berdasarkan ketersediaan data yang digunakan untuk menyerang sistem kriptografi:

1. *Chipertext-only attack*
2. *Known-plaintext attack*
3. *Chosen-plaintext* attack
4. *Adaptive-chosen-plaintext attack*
5. *Chosen-chipertext attack*



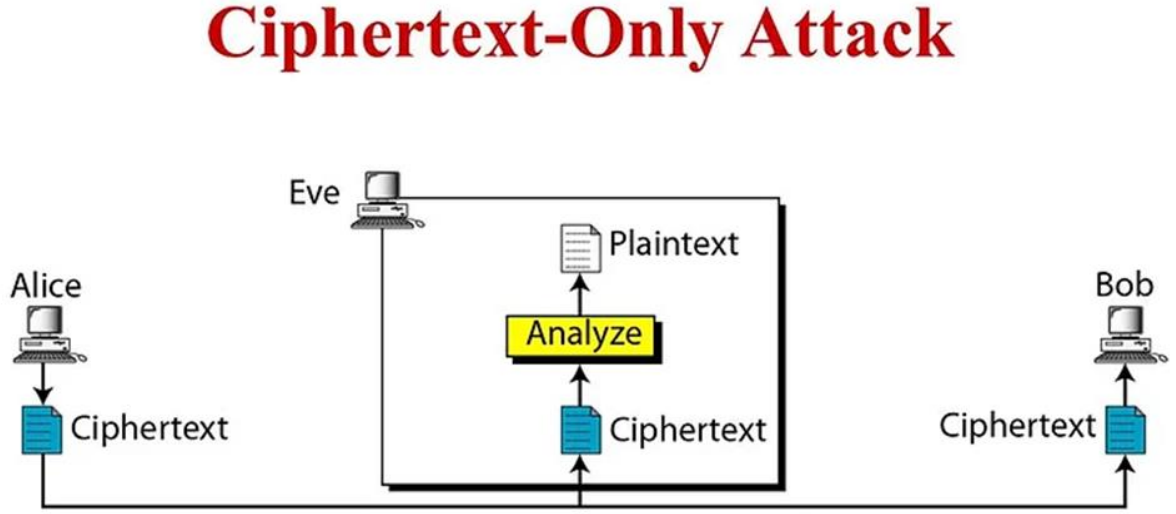
1. *Chipertext-only attack*

Kriptanalis hanya memiliki cipherteks saja. Ini serangan paling sulit karena informasi yang diperoleh hanya sedikit (cipherteks saja).

Teknik yang digunakan: *exhaustive key search,* terkaan, *metode* analisis frekuensi, dsb.

Diberikan: $C_1 = E_k(P_1), C_2 = E_k(P_2), \ldots, C_i = E_k(P_i)$

Deduksi: $P_1, P_2, \ldots, P_i$ atau k untuk mendapatkan P_{i+1} dari $C_{i+1} = E_k(P_{i+1})$.



Sumber: https://www.hackers-arise.com/post/2019/04/30/cryptography-basics-part-2-attack-models-for-cryptanalysis

```
hQIMAw3Jn/nLK/38ARAAsSXLDhCtzUYKMptNxZImJXwhhIRm3QxfuyHjJ93ASylE
e+6ABkuyFLJhiKryxp/JmS/alMPfF7hx2aTgovagaPzTwTV1jo6If2mhdCl6keed
1Iz7C0f6jHIqq9d8g0bWDyvELEipn5LNDTX3Xp2Csx5ojRB2wckrUt1l1Xyj8G0H
4DQUYbINRmJVulJJC/acGvgOze66pHuRgSCxxHDscefjXenh/XejSYTo7aMi+Es7
DCcD49zH6ZLDQN6BlN9q2oFI8QIhQ2y1QJbat1dWi/4yYWlKZcLKRSm8eo/gNCdL
h9MncXBBSfgbvbu67CDZ9GO5geZOn3LzQOpJ8hrZq/6K/uMcUKeZjW3RCo0T754f
E5zYe1wUgtwS/lmQ2w5PQF/89bpshtDSYuL1fZgzrsE6DwophuCri5zwCGbEKlsI
g6REIETFbZ2aCL4N2pZVunCIEuoP0zgEB6+M9egdpyxMsMqEBVg3AH7Sa1AtEguP
T/MCxI0bZHCUhPupEKT8slbSrDNxTWMUXQt3XpL0bGCCrDMKLSoWYfDiNnRkFbWK
iiqw9hx4Q9CJg7xX7JRnVgwOeREiFnMYSbFlvPSxEOu6FdBYhdqSefKin4Wnkmdw
qrSl8fjIW/kZ2v72uz0buEKkY9ubBox76yjlRo9KUQMs3em03kc64959gTDiZ0qF
AgwDrosDPQ2BeYQBD/9H5VKFw0an5j5MX1JpOSBAqNGKWq2bcEFnwJfk0DDlhyHD
owHiG7gDowCS+5y/pf56v36HkzpJZATKqoRyKVxmQOxU9l3YnPc5fw8iFhxlrfcG
ywzkJh/BRDQ/uy5fhGc/PbSm6iLv/SkkWTK8PSUD+g1yZyK0W7WkMh9QYS2OE7lQ
qbwpNiy57reWkUWCoE4QmKqqpe7NXXM0eLT9l2D0hG2lthyvTvspkpxszl8+HMJv
M2LMcY2FmmZWAJSdxsQSq9NQdyvCJX2D8oa89WQyXmp7mPXL7BQfoQNPndmn6Obi
0EQojoeMRNh14XNhMjPjxW7m34rH2gtvdN3Dg8iFrtocoVJqXqU3N+9T2sNe/bS8
```

Cipherteks

- Contoh pada Caesar Cipher, penyerang hanya memiliki cipherteks berikut:

  KHOORZRUOG

  Penyerang tidak tahu kunci (pergeseran berapa huruf), hanya punya ciphertext itu saja.

  Namun karena Caesar cipher hanya punya 25 kemungkinan pergeseran huruf, penyerang bisa mencoba semua pergeseran (brute force).

  Saat dicoba pergeseran 3 huruf ke kiri, diperoleh plaintext:

  HELLOWORLD

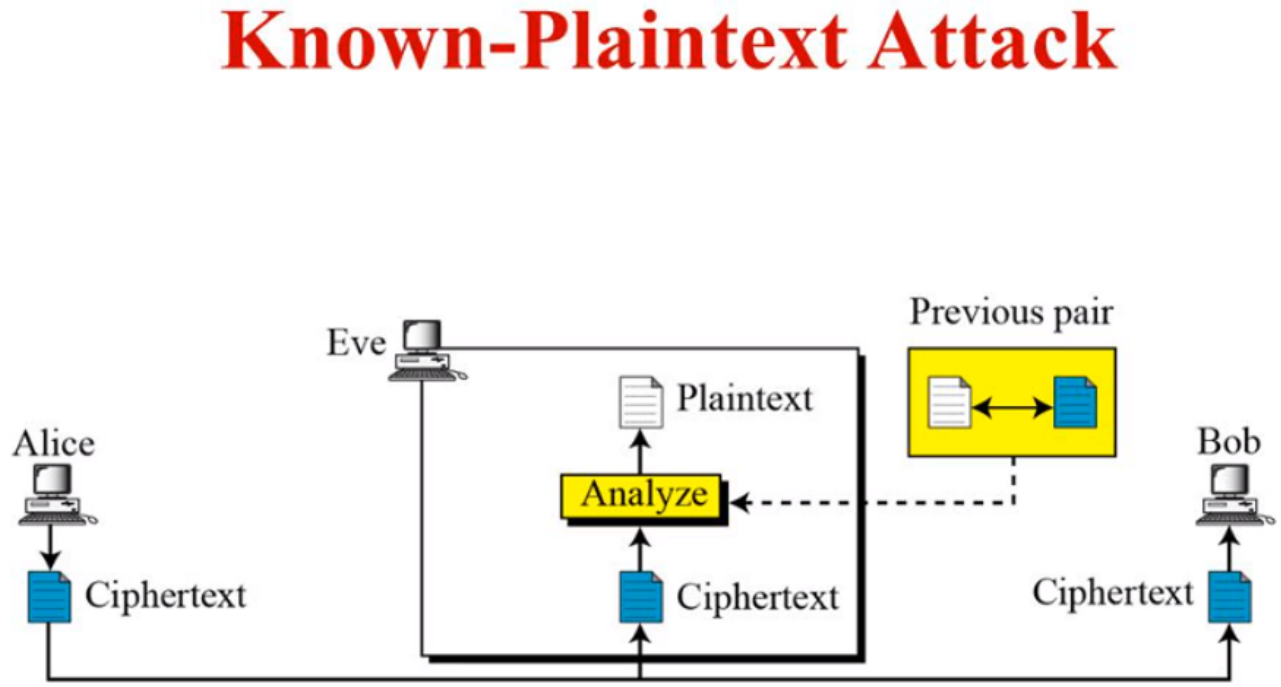
  Serangan berhasil!

2. *Known-plaintext attack*

Diberikan sejumlah pasangan plainteks dan cipherteks yang berkoresponden:

P_1 dan $C_1 = E_k(P_1)$, P_2 dan $C_2 = E_k(P_2)$, … , …, P_i dan $C_i = E_k(P_i)$

Deduksi: k untuk mendapatkan P_{i+1} dari $C_{i+1} = E_k(P_{i+1})$.



Sumber: https://www.hackers-arise.com/post/2019/04/30/cryptography-basics-part-2-attack-models-for-cryptanalysis

- Ini sering terjadi jika sebagian isi pesan bisa ditebak (misalnya header file, protokol standar, salam pembuka email).
- Beberapa pesan yang formatnya terstruktur membuka peluang untuk menerka plainteks dari cipherteks yang bersesuaian.

  Contoh:

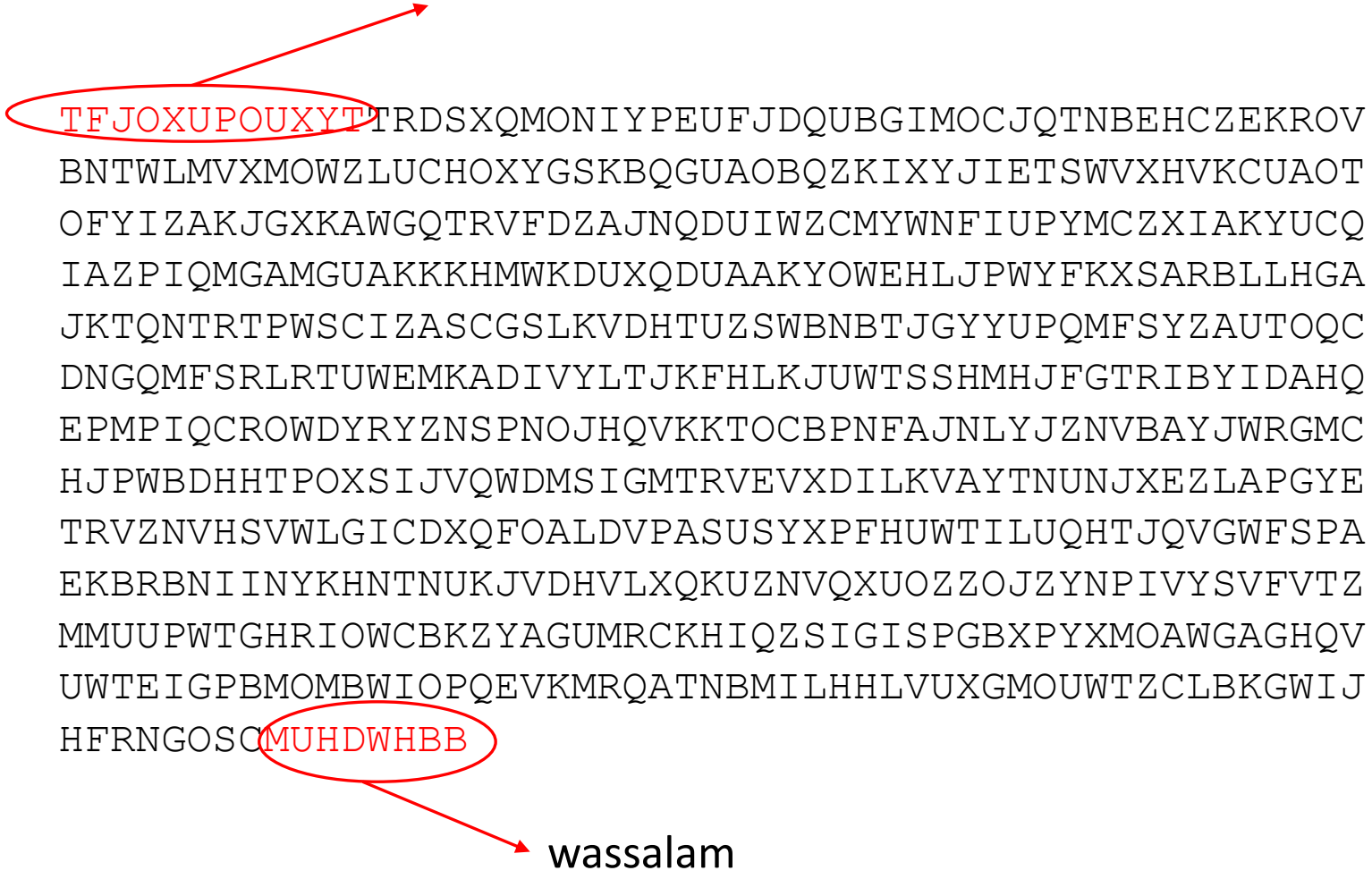
  From dan *To* di dalam *e-mail*,

  ”Dengan hormat”, *wassalam*, pada surat resmi.

  #include, *program*, di dalam *source code*

- Dengan menggunakan pasangan plainteks dan cipherteks, kriptanalis dapat menemukan kunci enkripsi (lihat pembahasan *cipher* klasik *affine cipher* dan *hill cipher*)

Dengan hormat



- Contoh di dalam Caesar Cipher, misalkan penyerang tahu ciphertext:

  KHOOR ZRUOG

  dan juga tahu plaintext-nya seharusnya mengandung kata:

  HELLO

  Dari pasangan H→K, penyerang tahu pergeseran (k) = 3.

  Dengan kunci itu, semua ciphertext lain bisa didekripsi.

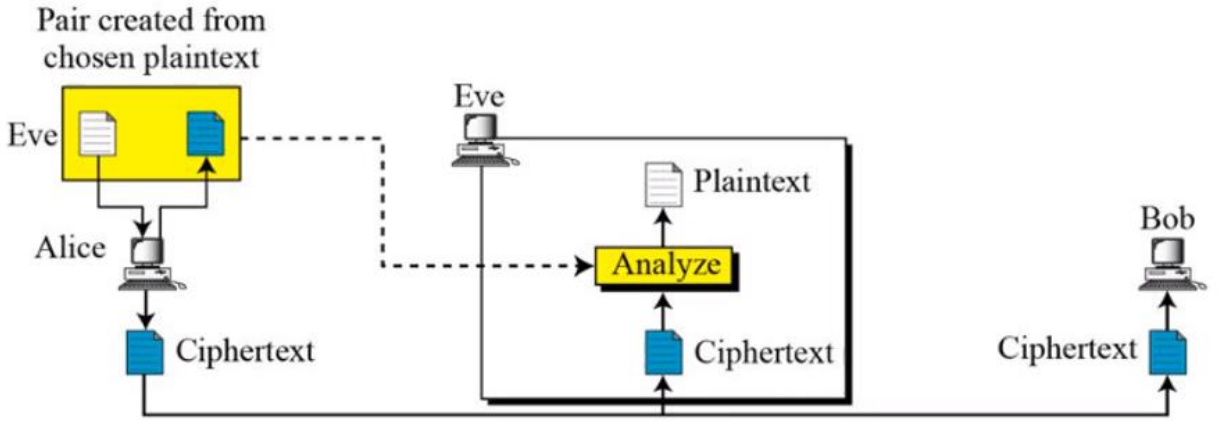
3. *Chosen-plaintext attack*

Kriptanalis dapat memilih plainteks sesuka hati untuk dienkripsikan, yaitu plainteks-plainteks yang lebih mengarahkan penemuan kunci, lalu mengirim plainteks ke sistem enkripsi. Asumsi: Penyerang memiliki akses ke sistem enkripsi

Diberikan: $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \ldots, P_i, \ C_i = E_k(P_i)$
di mana kriptanalis dapat memilih diantara $P_1, P_2, \ldots, P_i$

Deduksi: k untuk mendapatkan P_{i+1} dari $C_{i+1} = E_k(P_{i+1})$.

Chosen-Plaintext Attack



- Contoh pada Caesar Cipher, penyerang memilih plaintext sederhana, misalnya:

  AAAAA

  lalu mengirim plainteks ke sistem enkripsi

  Sistem mengembalikan ciphertext, misalnya:

  FFFFF

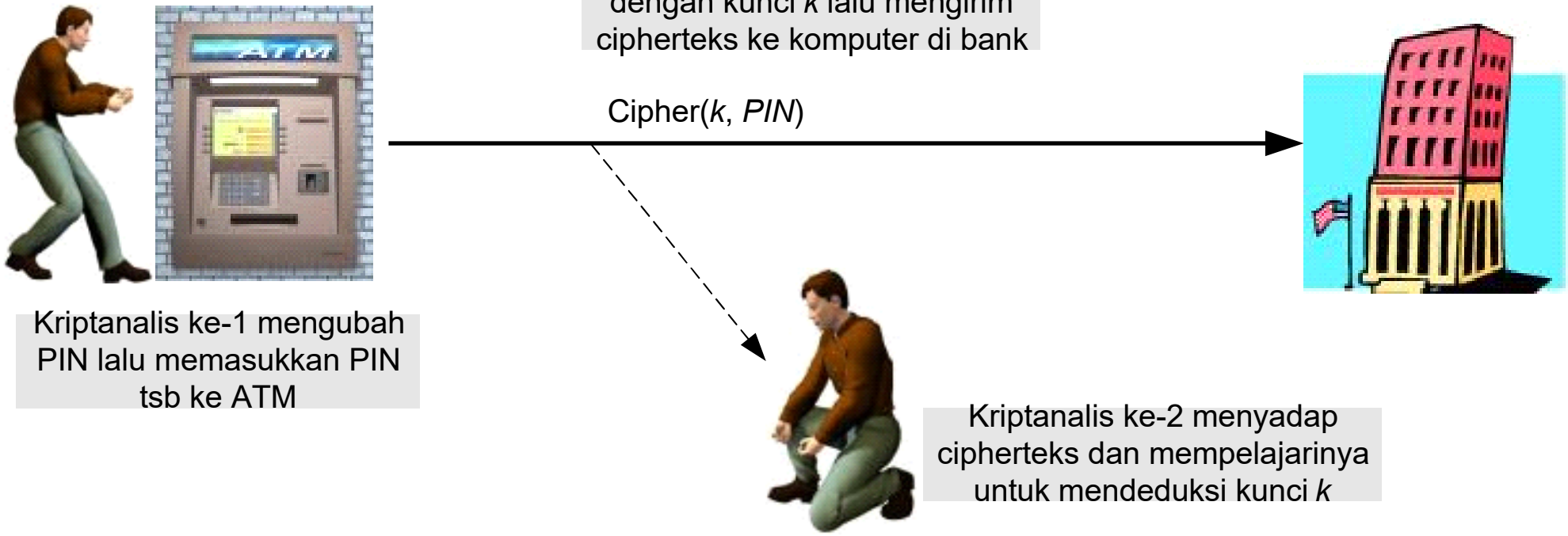
  Dari sini terlihat huruf A → F, berarti pergeseran = 5.

  Dengan kunci itu, penyerang bisa mendekripsi semua ciphertext lain.

- Contoh di dalam Vigenere Cipher, ciphertext setiap huruf plainteks dihitung dengan:

$$C = (P + K)\ mod\ 26$$

- Penyerang memilih plaintext P = AAAAAAA… (= 000000…).
- Maka ciphertext yang keluar, C, pasti sama dengan K (C = K).
- Dengan mengetahui K, ciphertext lain bisa langsung dicoba didekripsi dengan K tersebut.



Chosen-plaintext attack

4. ***Adaptive-chosen-plaintext attack***

- *Adaptive chosen plaintext attack* adalah variasi dari *chosen plaintext attack*, dalam hal ini penyerang **tidak hanya memilih plaintext**, tapi dapat **menyesuaikan plaintext berikutnya** berdasarkan hasil *ciphertext* dari *query* sebelumnya.

- Artinya, penyerang belajar secara bertahap: setelah melihat hasil enkripsi pertama, ia memilih *plaintext* baru yang lebih “cerdas” untuk menggali informasi lebih dalam tentang algoritma/kunci.

- Contoh di dalam Caesar Cipher, penyerang memilih plaintext:

  A

  → Ciphertext keluar: F. Artinya pergeseran mungkin 5.

  Untuk memastikan, penyerang pilih plaintext lain:

  B

  → Ciphertext keluar: G. Polanya konsisten, maka kunci pergeseran dipastikan = 5.

- Contoh di dalam Vigenere Cipher
- Enkripsi setiap huruf plainteks:

$$C = (P + K) \bmod 26$$

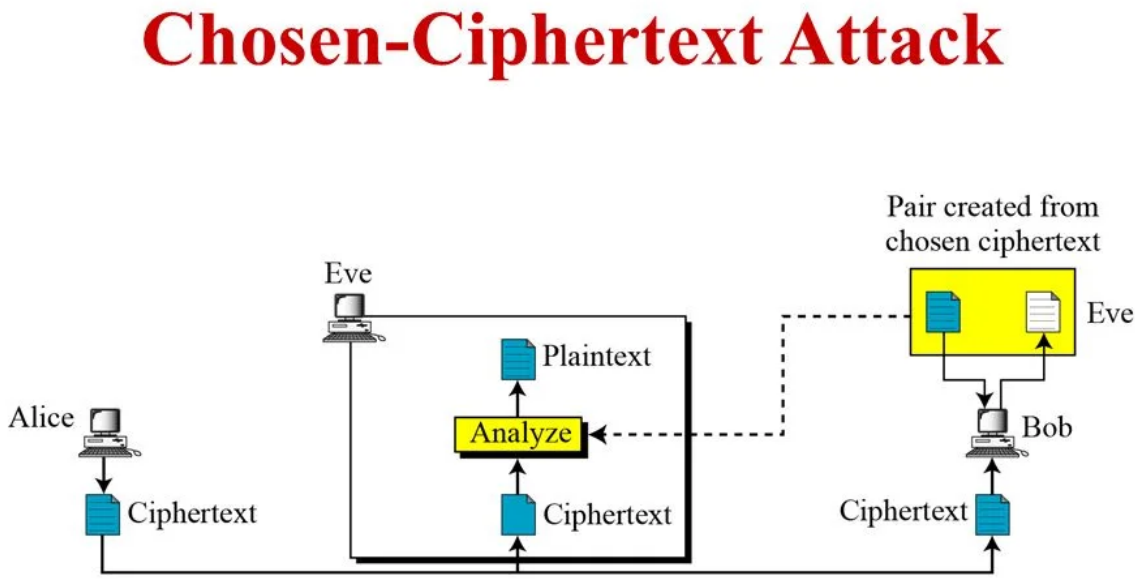
- Penyerang adaptif:
  - Pertama masukkan plaintext = AAAA (=0000), maka ciphertext C = K.
  - Setelah tahu K, ia coba plaintext lain untuk memverifikasi, misalnya BBBB (=1111), hasilnya menegaskan hasil enkripsi dengan K yang sama.
  - Dengan informasi itu, ciphertext lain bisa langsung dipecahkan.

5. *Chosen-ciphertext attack*

Penyerang dapat memilih ciphertext tertentu dan mendekripsinya dengan oracle yang bisa mendekripsi, dan menggunakan hasil dekripsi untuk memecahkan cipher atau menemukan kunci.

$C_1, P_1 = D_k(C_1), C_2, P_2 = D_k(P_2), \ldots, C_i, P_i = D_k(C_i)$

Deduksi: k (yang mungkin diperlukan untuk mendekripsi pesan pada waktu yang akan datang).



- Contoh di dalam Caesar cipher, penyerang memilih ciphertext:

  C = ‘A’

  lalu sistem dekripsi menghasilkan plaintext:

  P = ‘X’

  berarti penyerang mengetahui pergeseran huruf (k) = 3

Sebuah algoritma kriptografi dikatakan aman secara komputasi (*computationally secure*) bila ia memenuhi tiga kriteria berikut:

1. Persamaan matematika yang menggambarkan operasi di dalam algoritma kriptografi sangat kompleks sehingga algoritma tidak mungkin dipecahkan secara analitik.

2. Biaya untuk memecahkan cipherteks melampaui nilai informasi yang terkandung di dalam cipherteks tersebut.

3. Waktu yang diperlukan untuk memecahkan cipherteks melampaui lamanya waktu informasi tersebut harus dijaga kerahasiaannya.