Bahan kuliah IF4020 Kriptografi

07 - Serangan Terhadap Kriptografi

Oleh: Rinaldi M

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

Pendahuluan

- Keseluruhan *point* dari kriptografi adalah menjaga kerahasiaan pesan atau kunci dari pihak ketiga. Pihak ketiga: musuh, lawan, penyadap, kriptanalis (*cryptanalyst*).

- Pihak lawan berusaha memecahkan cipherteks dengan melakukan serangan terhadap sistem kriptografi.

- Tujuan serangan adalah untuk mengungkap plainteks dari cipherteks atau mendapatkan kunci dekripsi, sehingga kunci dekripsi dapat digunakan untuk mendekripsi potongan cipherteks lainnya.

Serangan (*attack*)

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

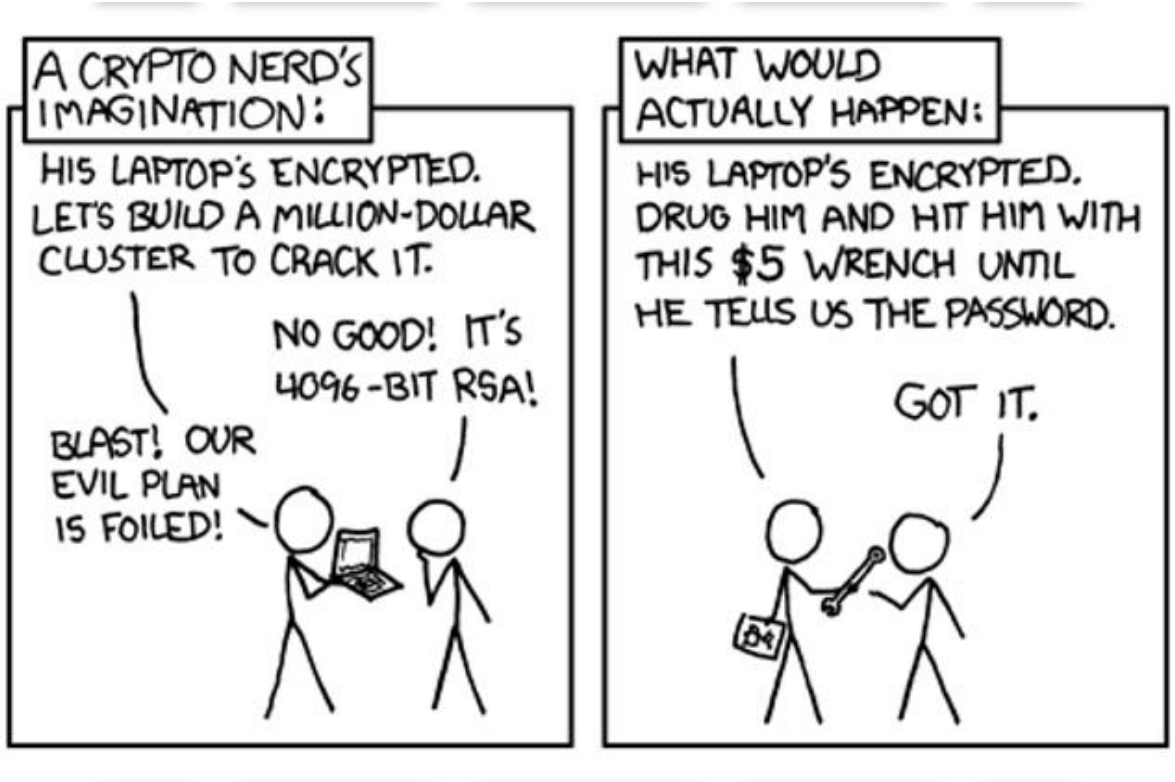
Prinsip Kerckhoff: Semua algoritma kriptografi harus publik; hanya kunci yang rahasia.

Jadi,keamanan sistem kriptografi terletak pada kunci!

Jenis-jenis Serangan (1)

- Berdasarkan ketersediaan informasi 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*

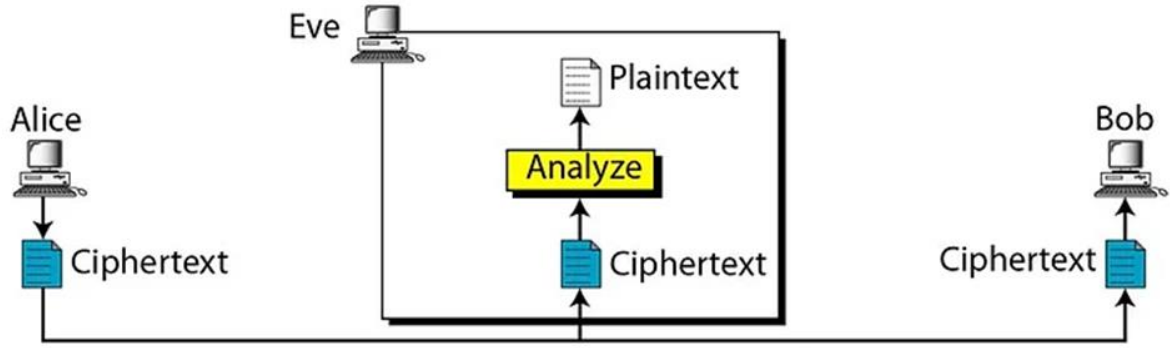
Kriptanalis hanya memiliki cipherteks saja. Ini serangan paling sulit karena informasi yang diperoleh hanya sedikit (cipherteks saja).

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 $C_{i+1} = E_k(P_{i+1})$.

Ciphertext-Only Attack



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 cipherteks 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 plainteks:

  HELLOWORLD

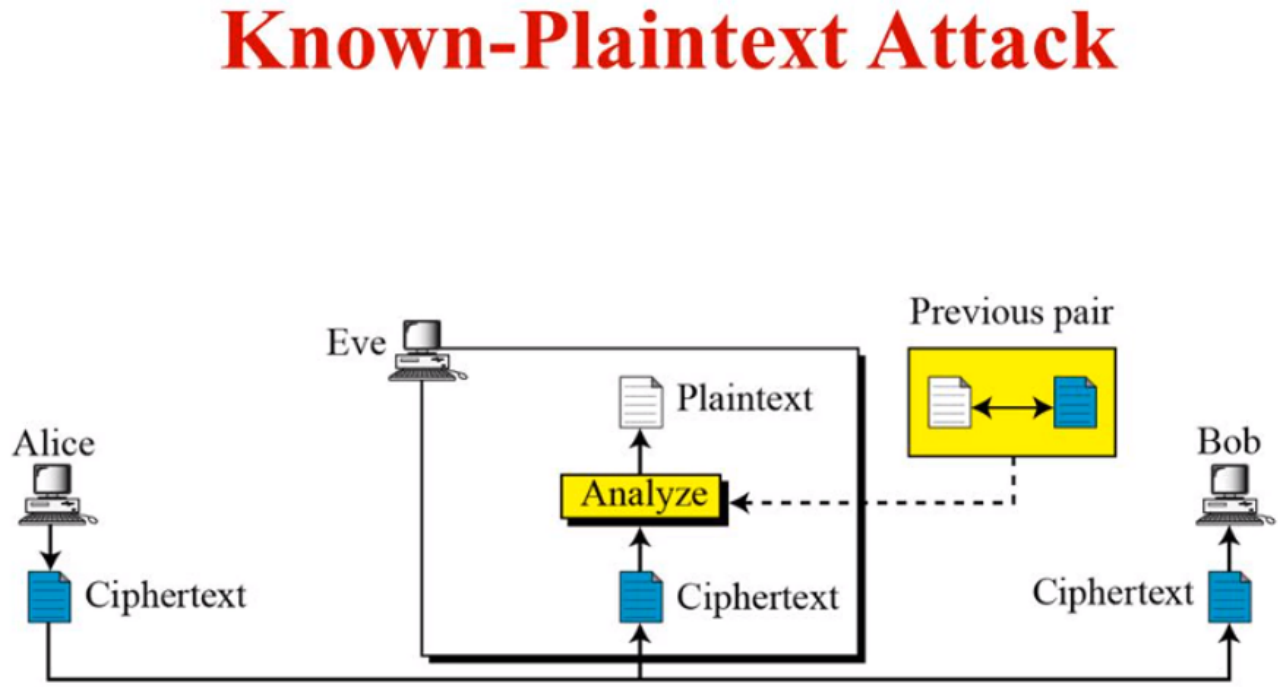
  Serangan berhasil menemukan kunci dan memecahkan cipherteks!

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:

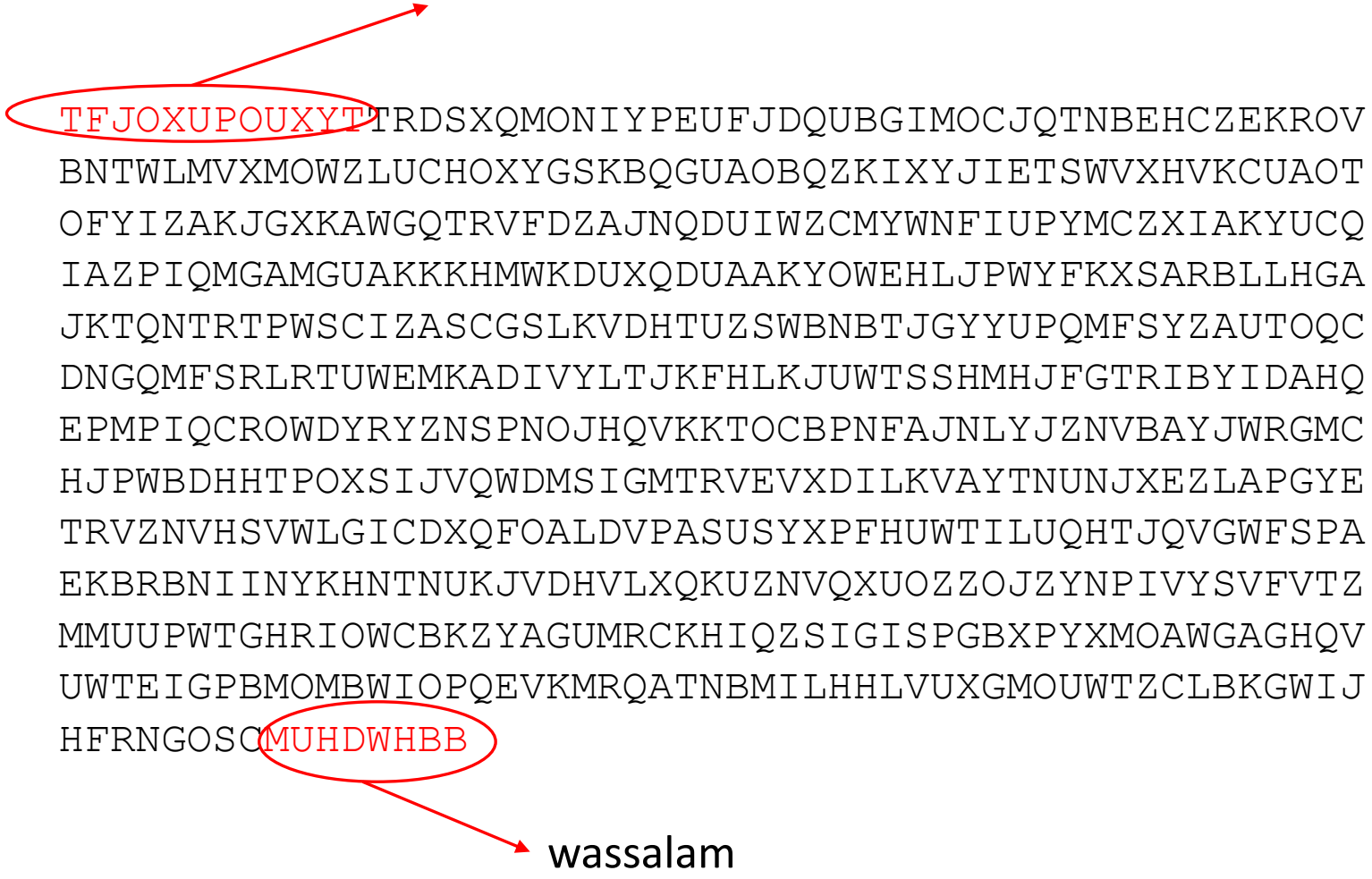
  From dan *To* di dalam *e-mail*,

  ”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 cipherteks:

  KHOOR ZRUOG

  dan juga tahu plainteks-nya seharusnya mengandung kata:

  HELLO

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

  Dengan kunci k = 3, semua cipherteks lain bisa didekripsi.

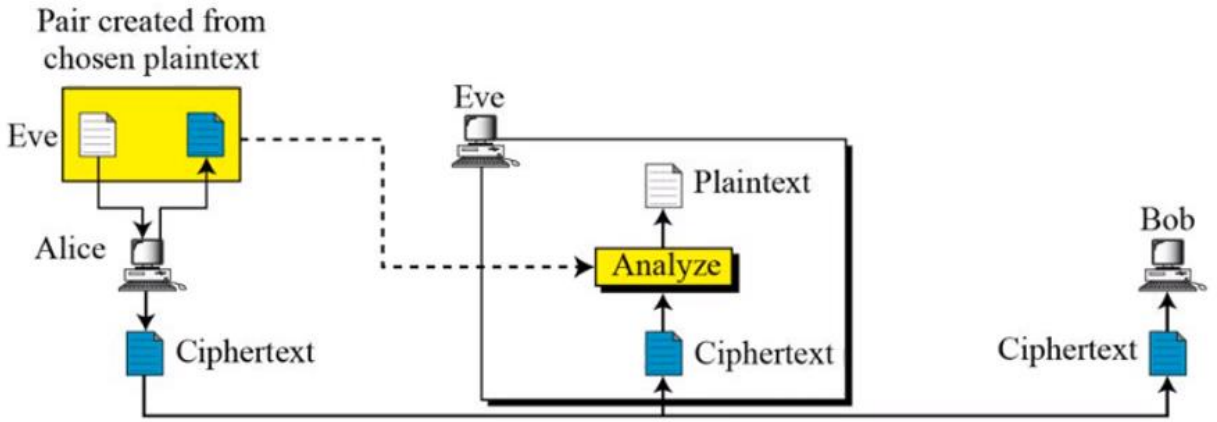
3. *Chosen-plaintext attack*

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 plainteks sederhana, misalnya:

  AAAAA

  lalu mengirim plainteks ke sistem enkripsi

  Sistem mengembalikan cipherteks, misalnya:

  FFFFF

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

  Dengan kunci itu, penyerang bisa mendekripsi semua cipherteks lain.

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

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

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

Kriptanalis ke-1 mengubah PIN lalu memasukkan PIN tsb ke ATM

ATM mengenkripsi PIN dengan kunci *k* lalu mengirim cipherteks ke komputer di bank

Cipher(*k*, *PIN*)

Kriptanalis ke-2 menyadap cipherteks dan mempelajarinya untuk mendeduksi kunci *k*

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 plainteks**, tapi dapat **menyesuaikan plainteks berikutnya** berdasarkan hasil *ciphertext* dari *query* sebelumnya.

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

- Contoh di dalam Caesar Cipher, penyerang memilih plainteks:

  A

  → Cipherteks keluar: F. Artinya pergeseran mungkin 5.

  Untuk memastikan, penyerang pilih plainteks lain:

  B

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

Contoh di dalam Vigenere Cipher:

- Enkripsi setiap huruf plainteks:

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

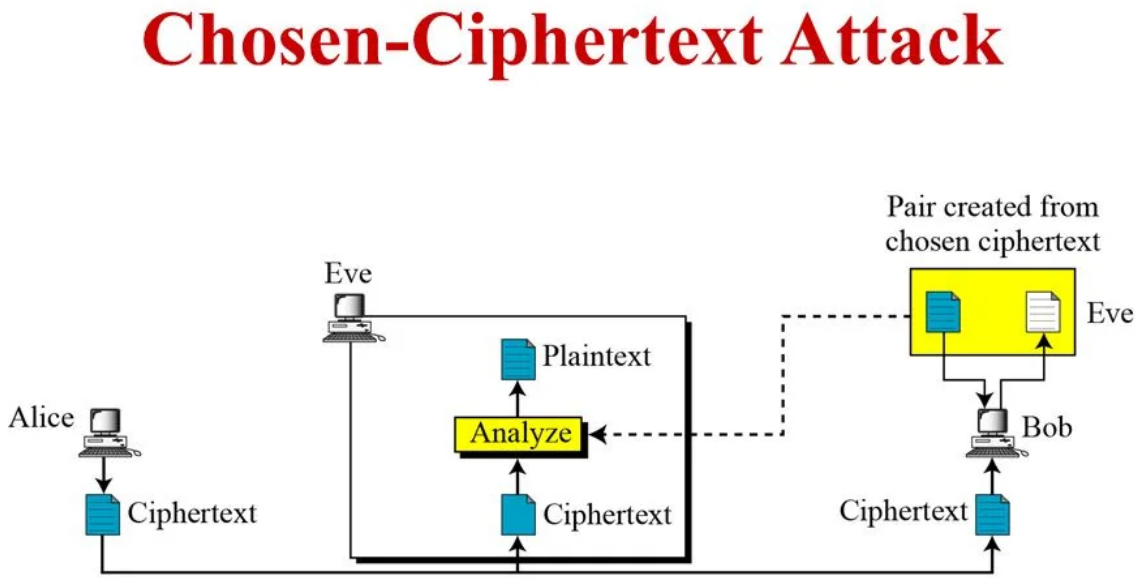
- Penyerang adaptif:
  - Pertama masukkan plainteks = AAAA (=0000), maka ciphertext C = K.
  - Setelah tahu K, ia coba plainteks lain untuk memverifikasi, misalnya BBBB (=1111), hasilnya menegaskan hasil enkripsi dengan K yang sama.
  - Dengan informasi itu, cipherteks lain bisa langsung dipecahkan.

5. *Chosen-ciphertext attack*

Penyerang dapat memilih cipherteks 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 cipherteks:

  C = ‘A’

  lalu sistem dekripsi menghasilkan plainteks:

  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.

Jenis-jenis Serangan (2)

- Berdasarkan berdasarkan tujuan atau hasil yang ingin dicapai penyerang:

1. *Key-Recovery Attack*
2. *Plaintext-Recovery Attack*
3. *Distinguishing Attack*
4. *Forgery Attack*
5. *Replay Attack*

1. Key-Recovery Attack

- Tujuan: memperoleh kunci kriptografi yang digunakan untuk mengenkripsi/dekripsi.
- Misalkan sebuah sistem menggunakan

  $C = E_K(P)$, P = plainteks, K = kunci, C = cipherteks
- Dalam *key-recovery attack*, penyerang berusaha menemukan K.
- Contoh: Misalkan sebuah cipher menggunakan kunci 16 karakter. Penyerang melakukan pencarian kunci secara *brute force*:

  $K_1 \rightarrow$ dekripsi C $\rightarrow P_1$
  $K_2 \rightarrow$ dekripsi C $\rightarrow P_2$
  $K_3 \rightarrow$ dekripsi C $\rightarrow P_3$
  ⋮

  sampai ditemukan kunci yang menghasilkan plainteks yang masuk akal.

2. Plaintext-Recovery Attack

- Tujuan: memperoleh plainteks, tetapi tidak harus memperoleh kunci.
- Contoh: Misalkan sebuah cipherteks dikirim:

  C = XQZ...

  Penyerang menganalisis struktur cipherteks dan mengetahui bahwa pesan tersebut kemungkinan berbunyi:

  MEET AT THE STATION

  tanpa mengetahui kunci sebenarnya.

  Dalam beberapa situasi, mendapatkan plainteks tertentu sudah cukup bagi penyerang. Misalnya, jika pesan berisi informasi penting, penyerang mungkin tidak perlu mengetahui kunci untuk mencapai tujuannya.

3. Distinguishing Attack

- Tujuan: menentukan apakah suatu data merupakan keluaran algoritma kriptografi atau data yang benar-benar acak (random).
- Jika secara statistik penyerang dapat membedakannya dengan probabilitas yang secara signifikan lebih baik daripada tebakan acak, berarti terdapat *distinguishing attack*.

- Contoh: misalkan sebuah cipher menghasilkan:

  10101010101010101010101010...

  Pola tersebut jelas tidak menyerupai random.

  Penyerang dapat membuat distinguisher sederhana:

  Jika jumlah 0 dan 1 sangat tidak seimbang atau terdapat pola tertentu, kemungkinan data tersebut berasal dari cipher.

- *Cipher* modern dirancang agar kelemahan seperti ini tidak mudah ditemukan.

4. Forgery Attack

- Tujuan: membuat pesan palsu yang diterima sebagai pesan sah oleh sistem.
- Serangan ini sangat penting pada sistem yang menggunakan MAC (*message authentication code*) atau *digital signature* (akan dijelaskan pada meteri kuliah selanjutnya).
- Misalkan Alice mengirim Pesan + MAC kepada Bob.
- Bob memeriksa MAC untuk memastikan pesan benar-benar berasal dari pihak yang memiliki kunci.
- Penyerang, Eve, ingin membuat:

  Pesan palsu + MAC yang valid

  sehingga Bob menganggap pesan tersebut autentik.

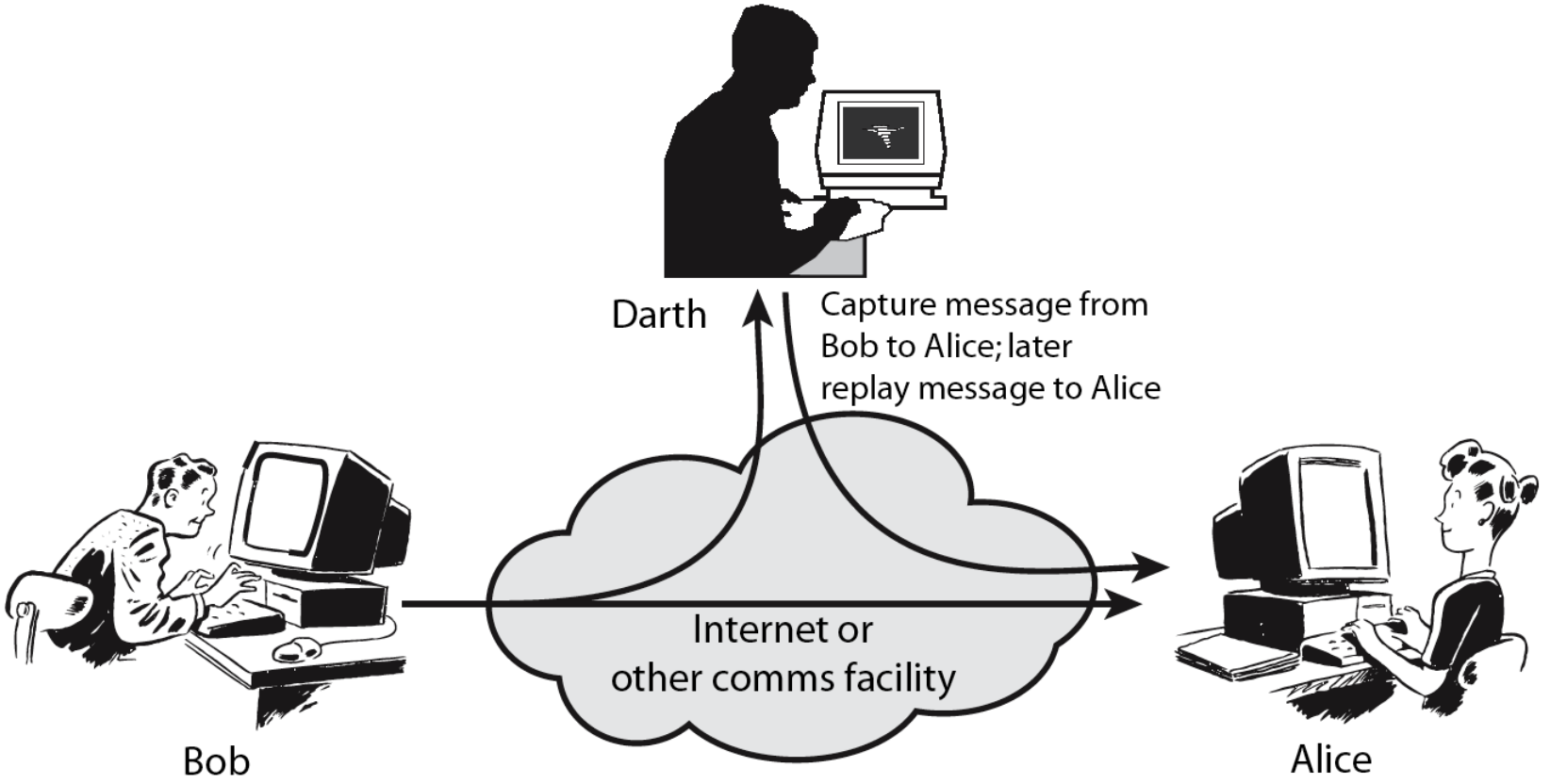
5. Replay Attack

- Tujuan: menggunakan kembali pesan sah yang pernah dikirim sebelumnya untuk menyebabkan suatu tindakan terjadi lagi.
- Yang menarik, pada *replay attack* penyerang tidak perlu memecahkan enkripsi.
- Untuk mencegah *replay attack*, biasanya digunakan mekanisme yang membuat setiap pesan mempunyai keunikan atau urutan, misalnya: *Nonce*, *Timestamp*, atau *Sequence number*
- Misalnya pesan diberi *sequence number*:

  Message 101
  Message 102
  Message 103

  Jika penyerang mengirim kembali Message 101 setelah sistem sudah menerima Message 103, sistem dapat menolaknya karena nomor tersebut sudah kedaluwarsa.

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



Replay Attack

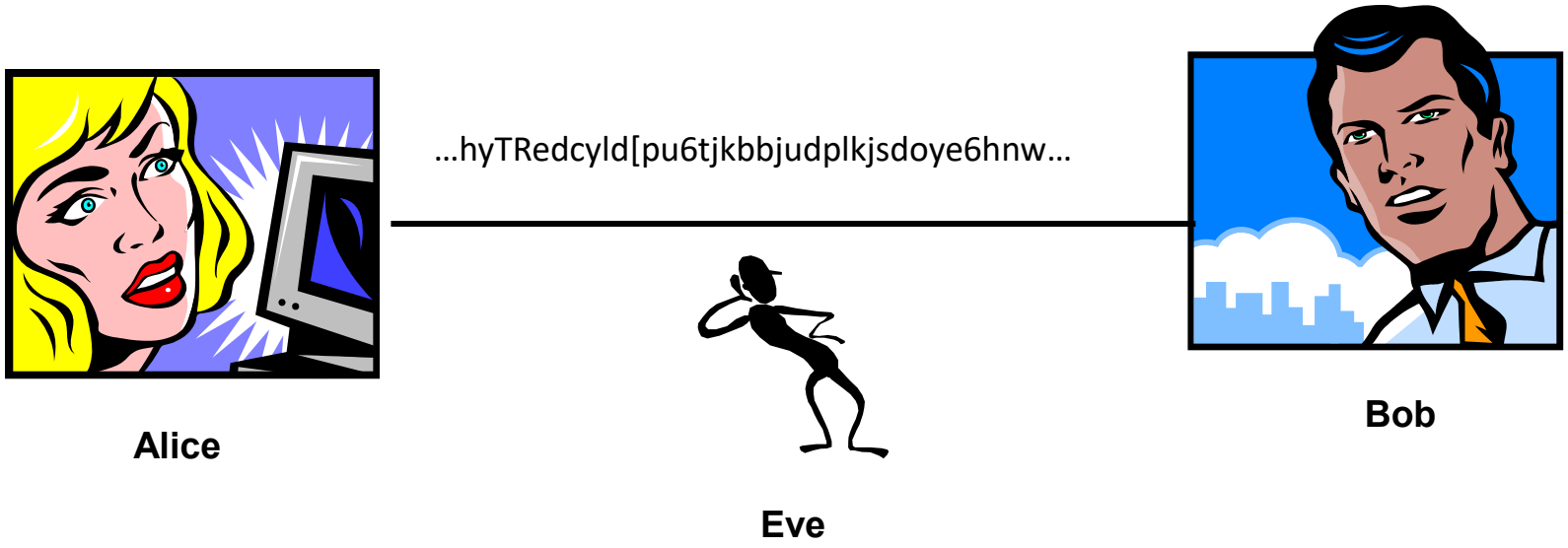
Jenis-jenis Serangan (3)

Berdasarkan keterlibatan penyerang dalam komunikasi pesan**:**

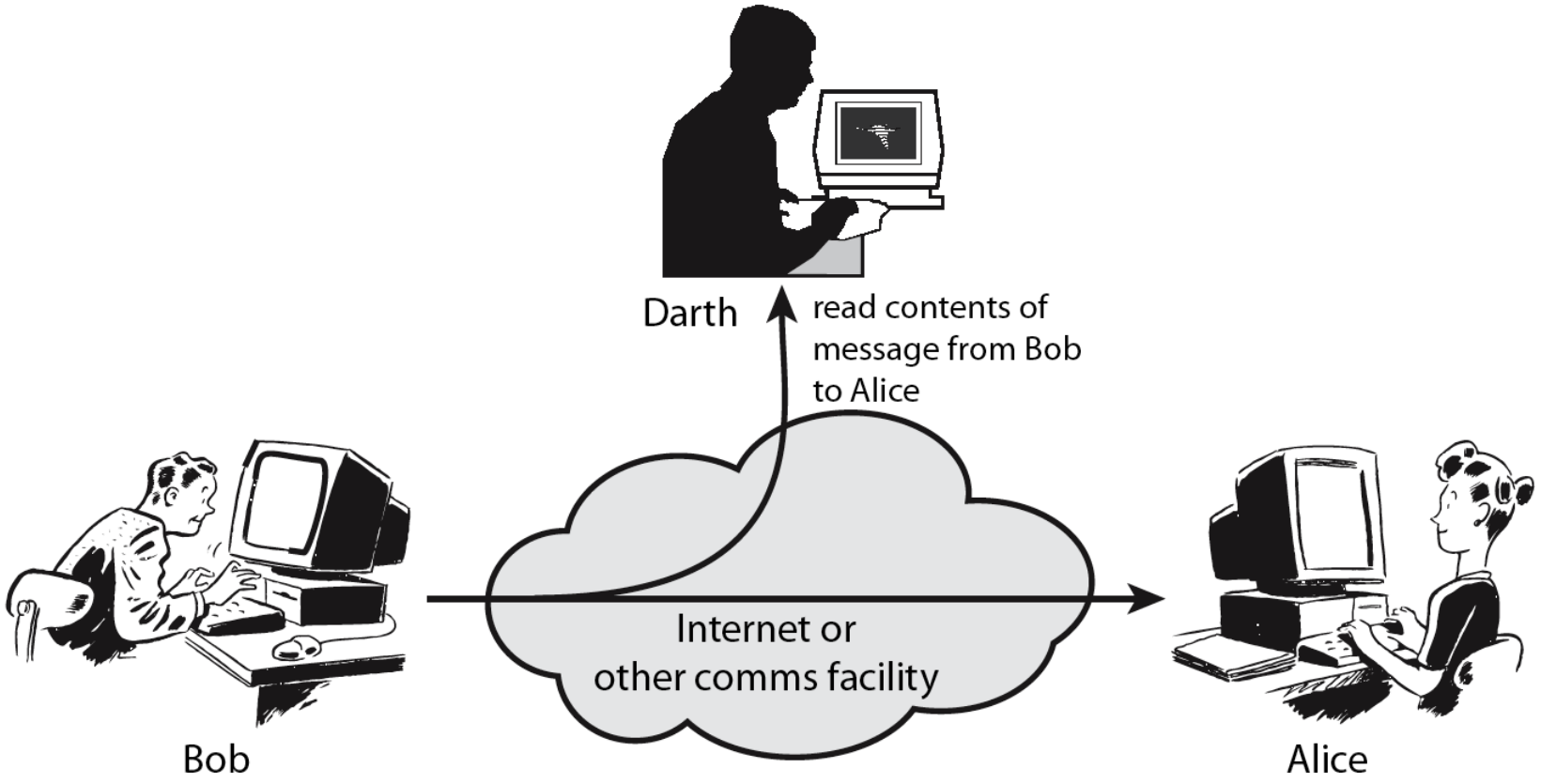
1. **Serangan pasif**
2. **Serangan aktif**

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

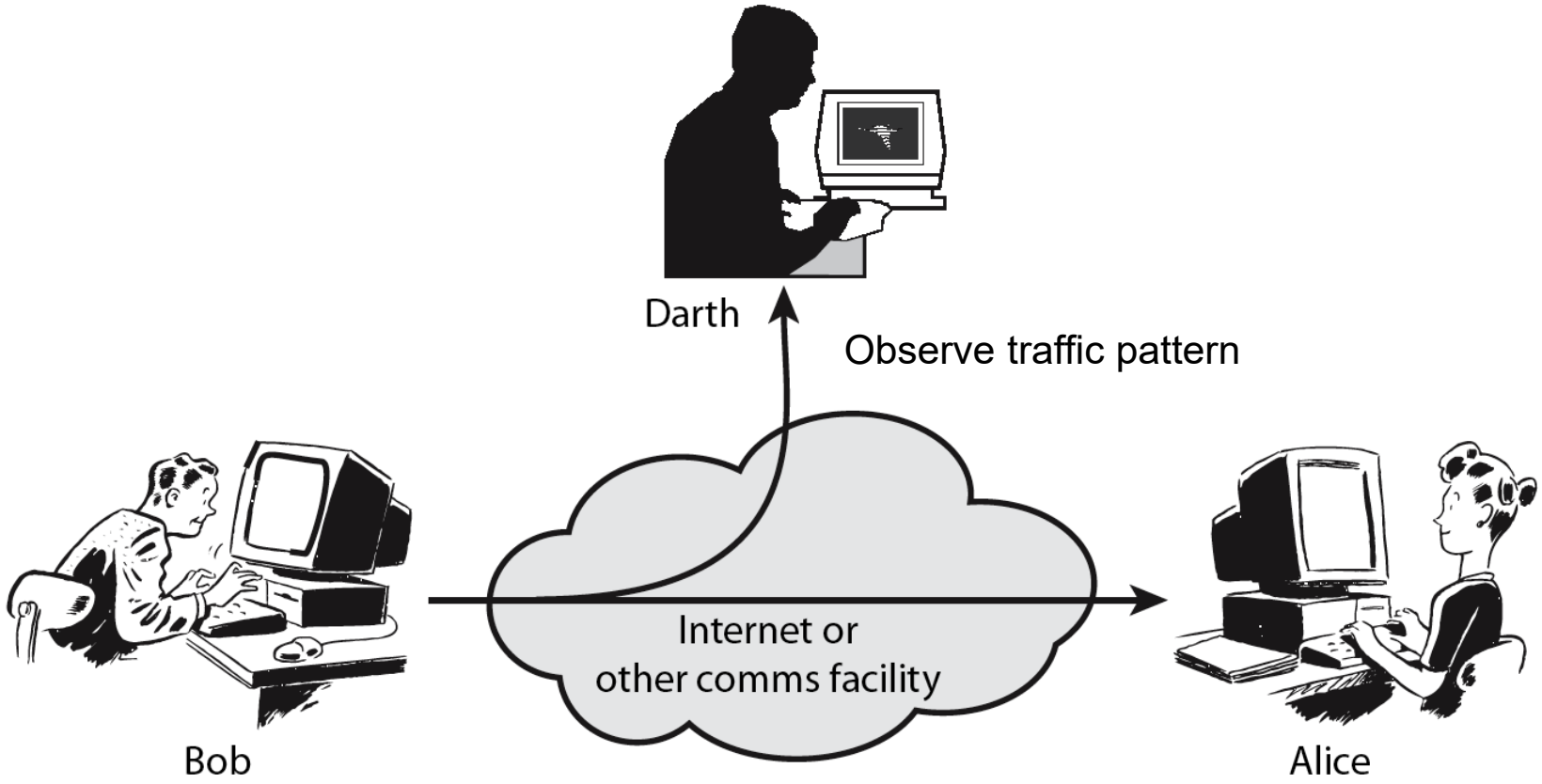


Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



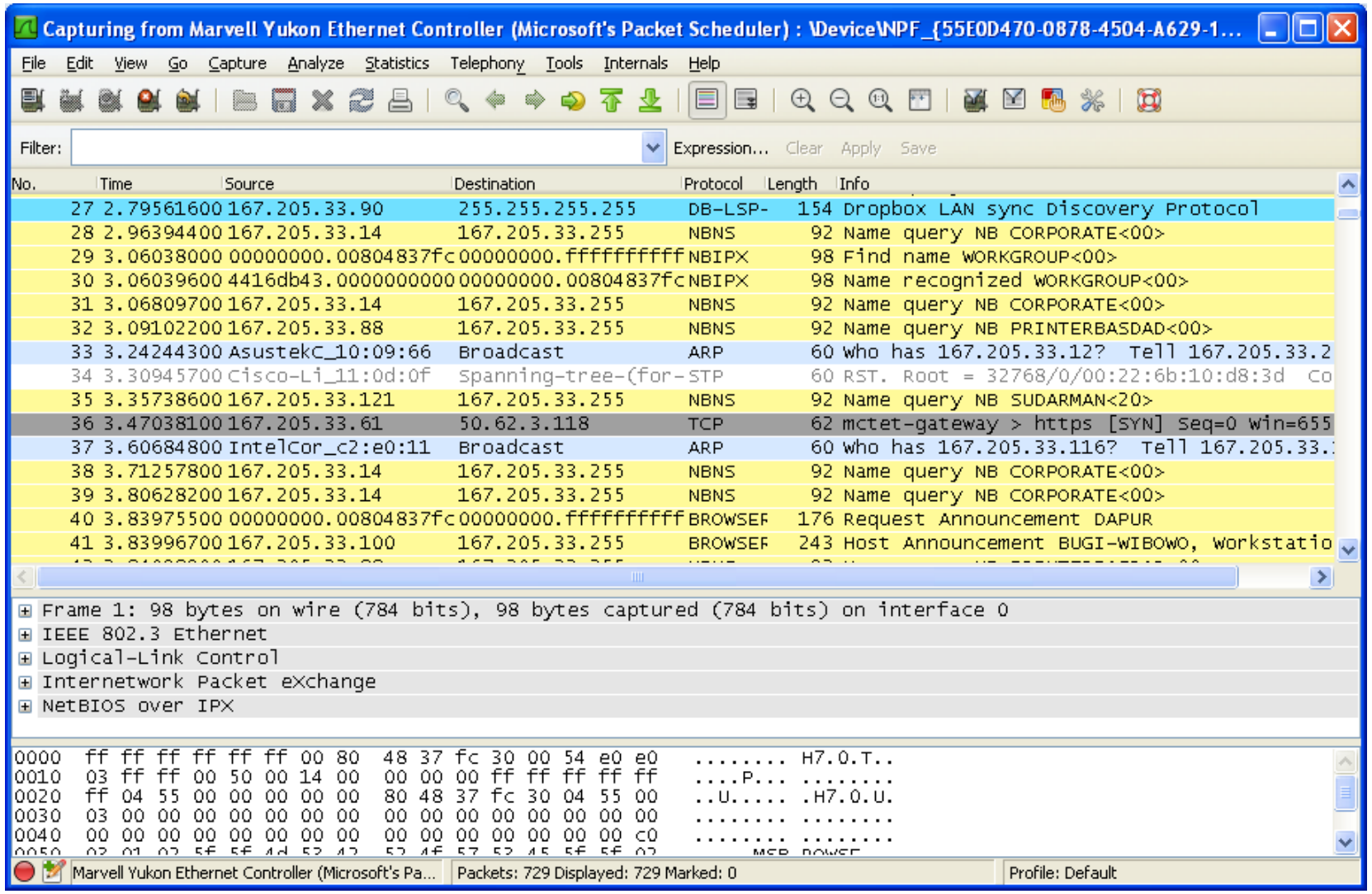
Passive Attack : Interception

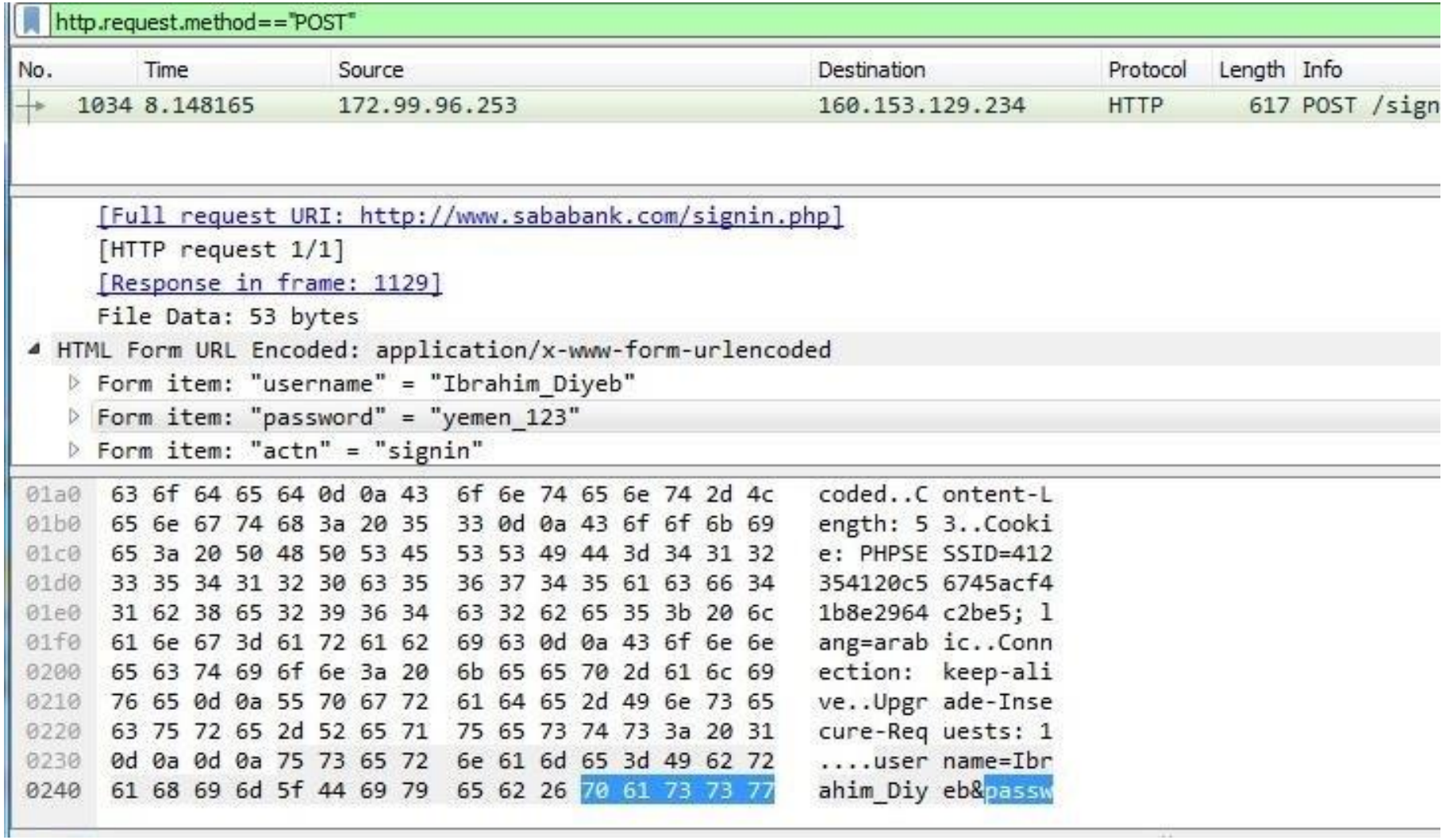
Sumber: William Stalling, Cryptography and Network Security
Overview & Chapter 1



Passive Attack : Traffic Analysis

Screenshot Wireshark (memantau *network traffic*)

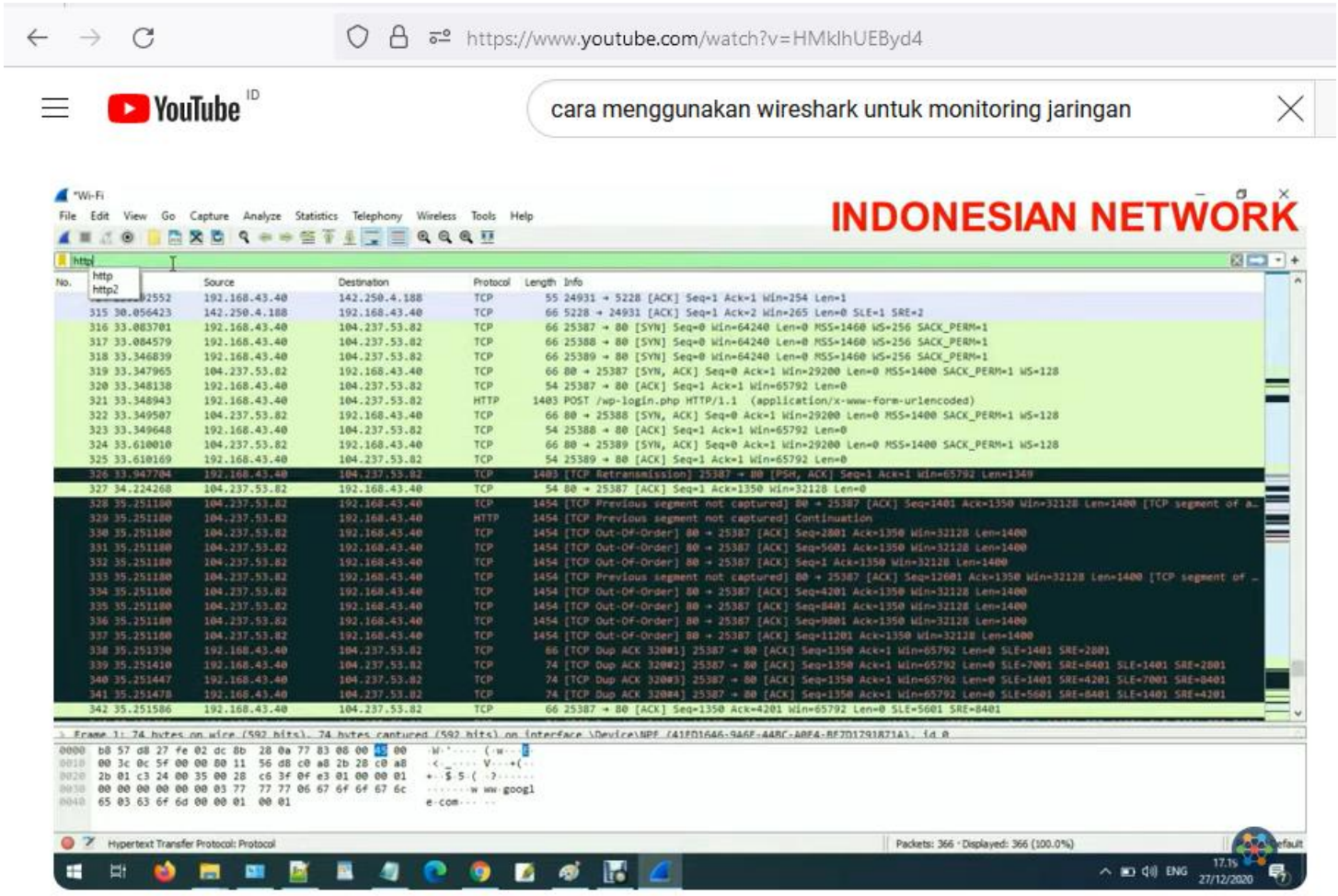




Filtering dengan Wireshark dapat menampilkan plainteks berupa *username* dan *password*

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

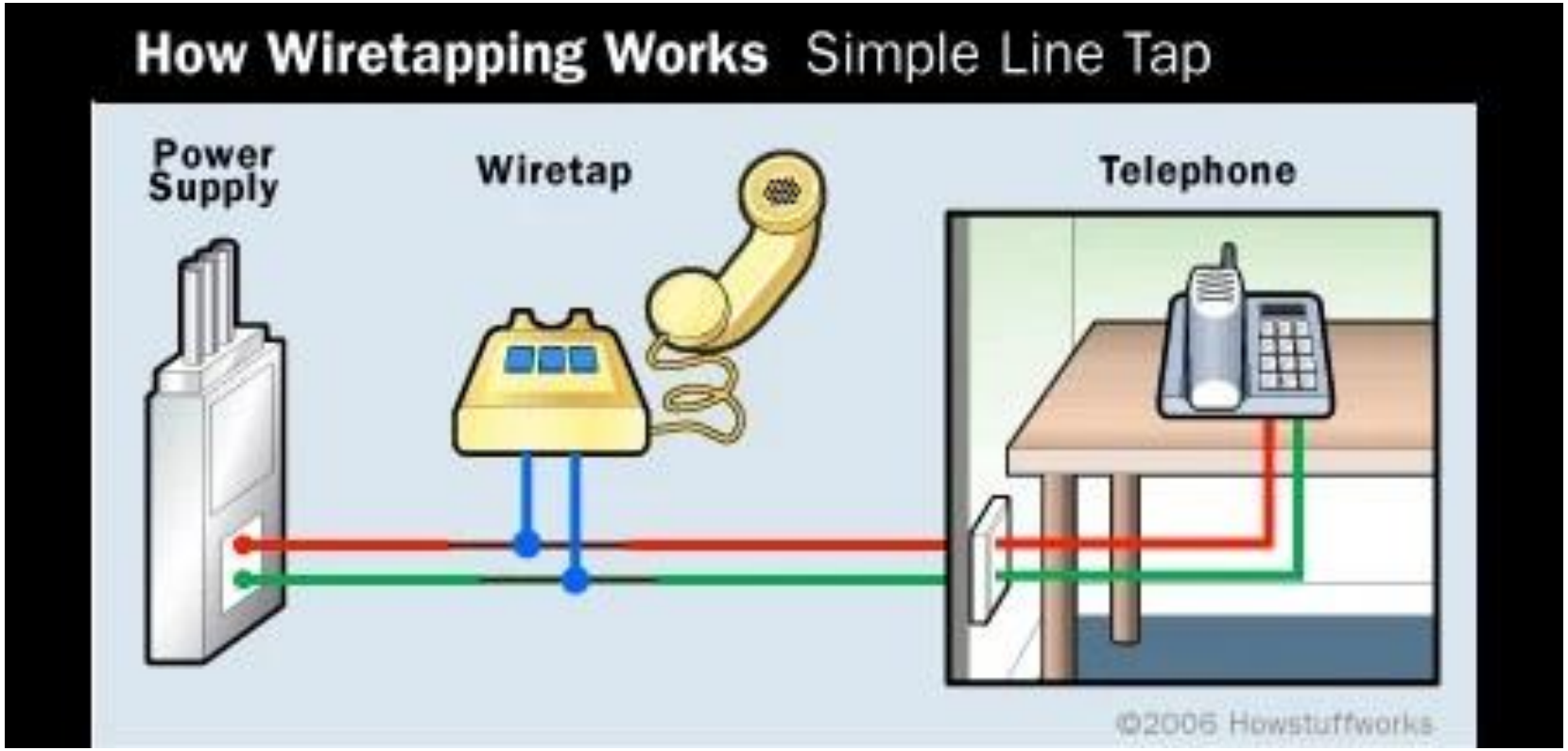


TUTORIAL CARA MENGGUNAKAN WIRESHARK | MENEMUKAN U53RN4M3 P455W0RD PADA LALU LINTAS PACKET JARINGAN

Metode penyadapan:

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

- *Wiretapping*

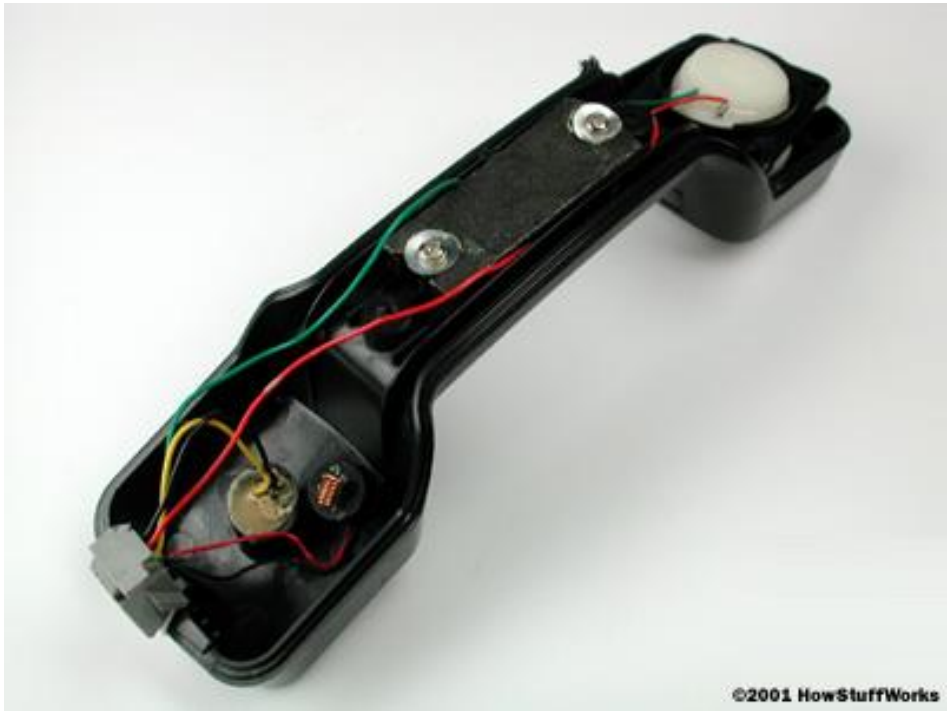


Baca di sini:

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

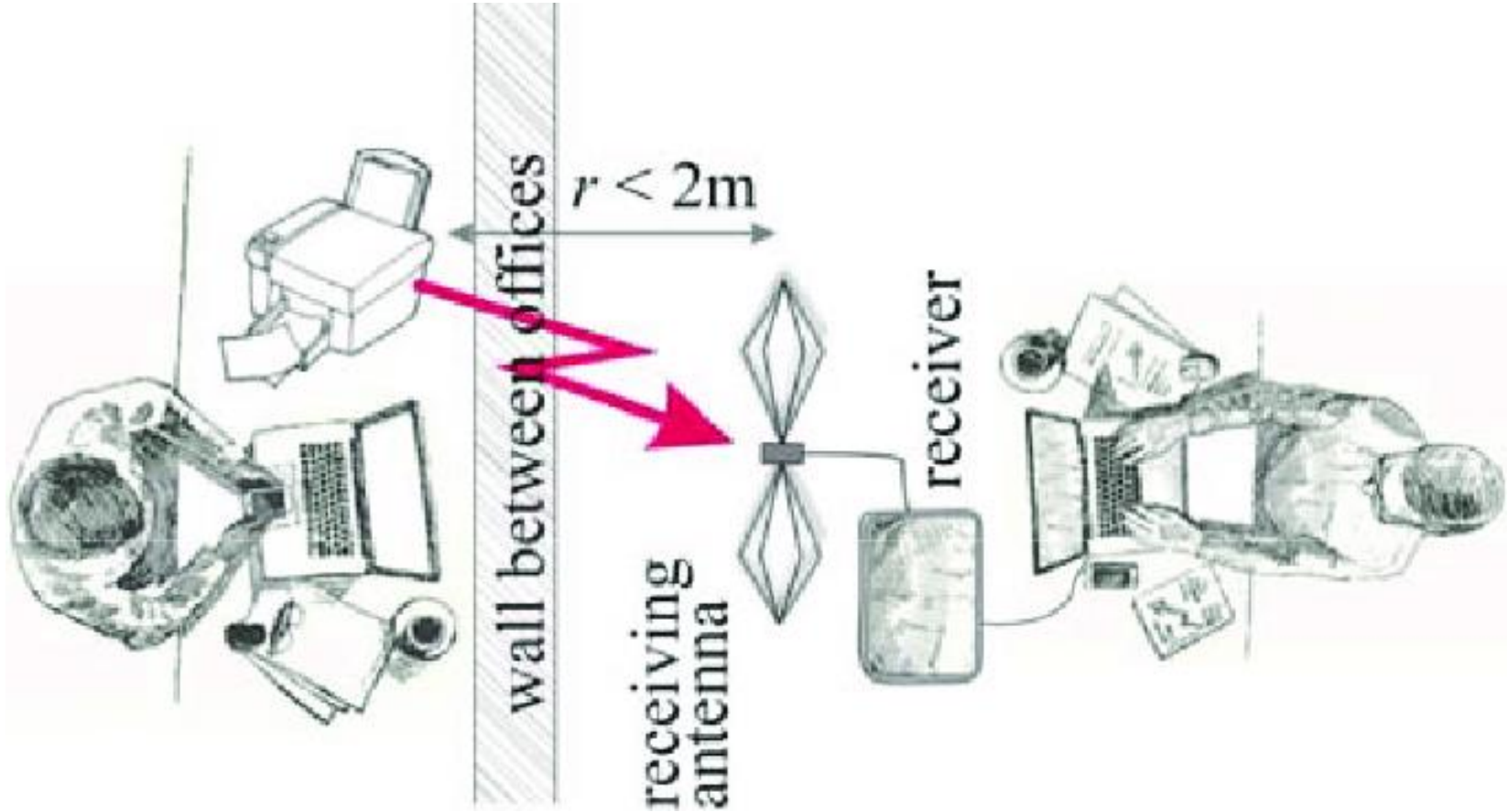
(sumber: http://www.howstuffworks.com/wiretapping.htm)



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.

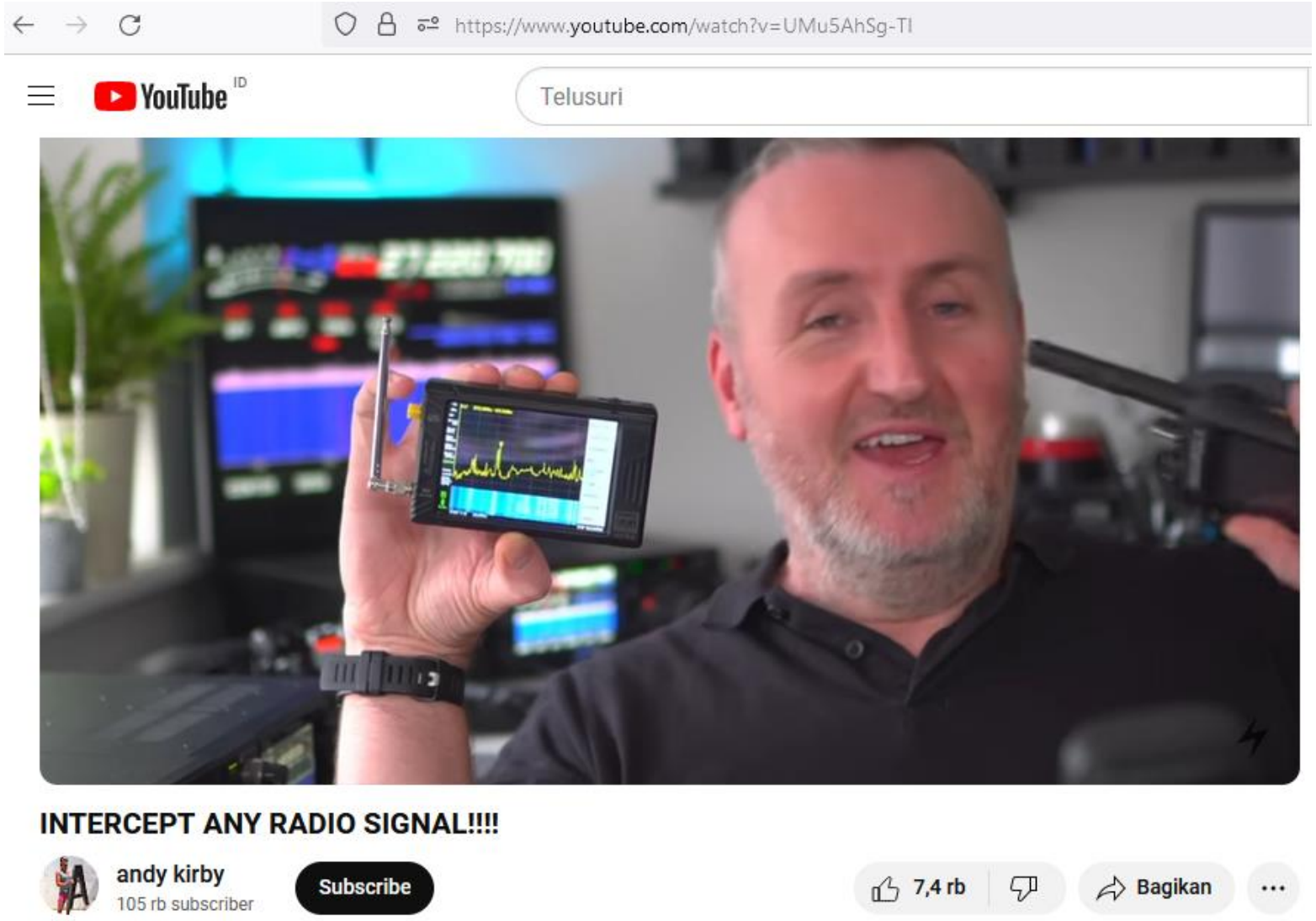
Inside a standard phone cord, you'll find a red wire and a green wire. These wires form a circuit like the one you might find in a flashlight. Just as in a flashlight, there is a negatively-charged end and a positively-charged end to the circuit. In a telephone cord, the green wire connects to the positive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

Electromagnetic eavesdropping

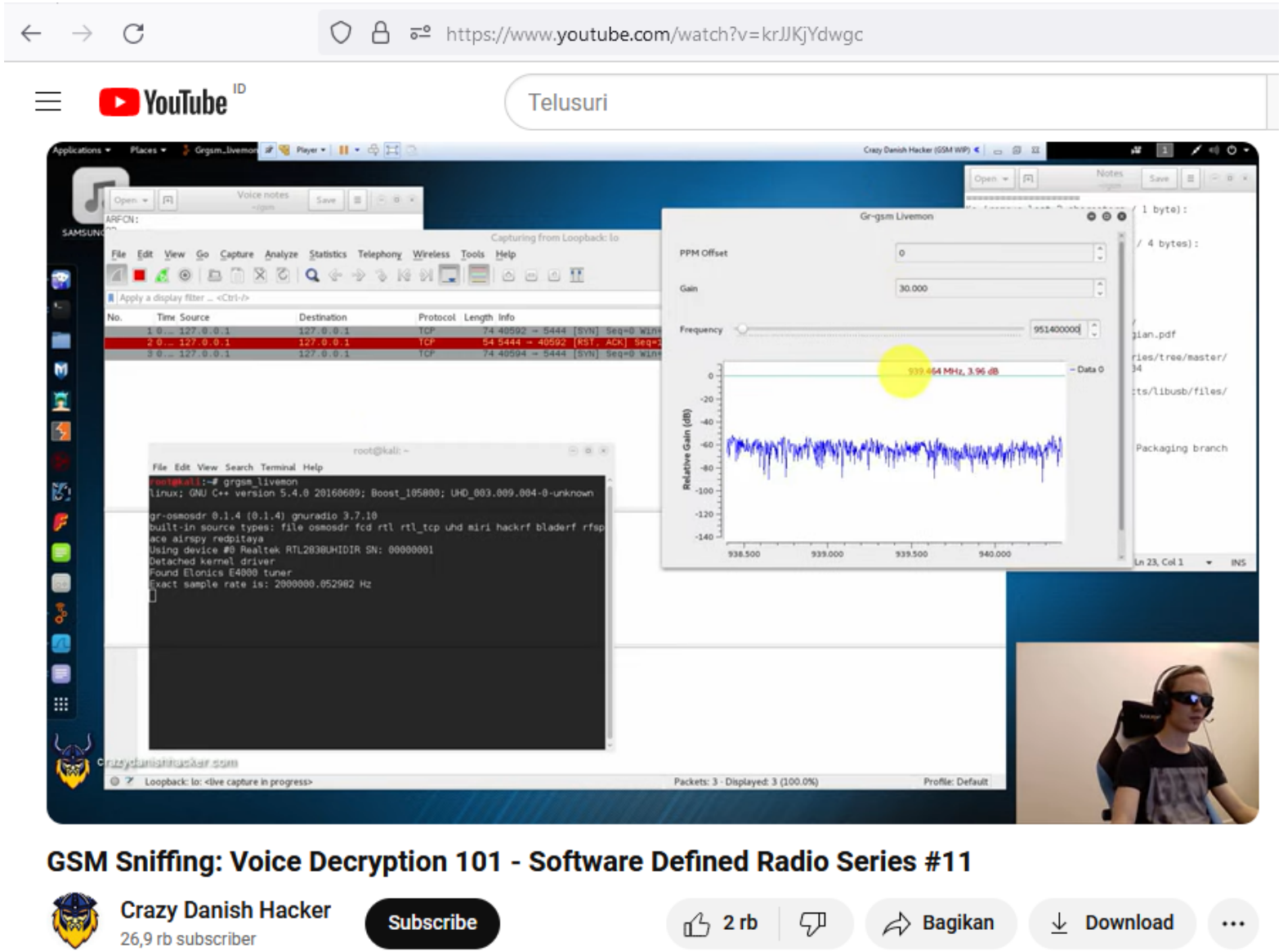


Sumber: https://www.researchgate.net/figure/An-anechoic-chamber-Figure-4-Example-for-an-electromagnetic-eavesdropping-process_fig3_324680618

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



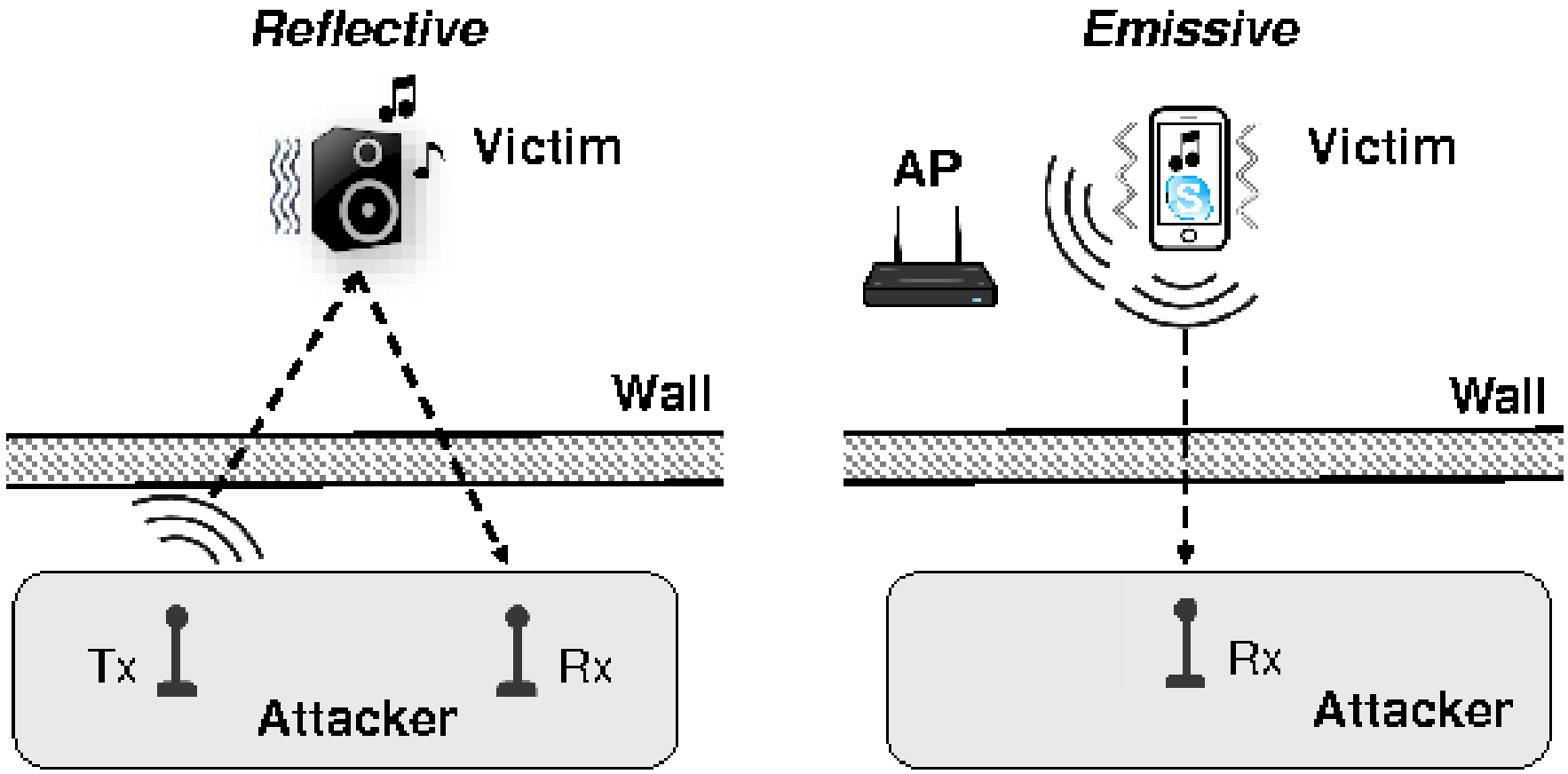
GSM Sniffing: Voice Decryption 101 - Software Defined Radio Series #11

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

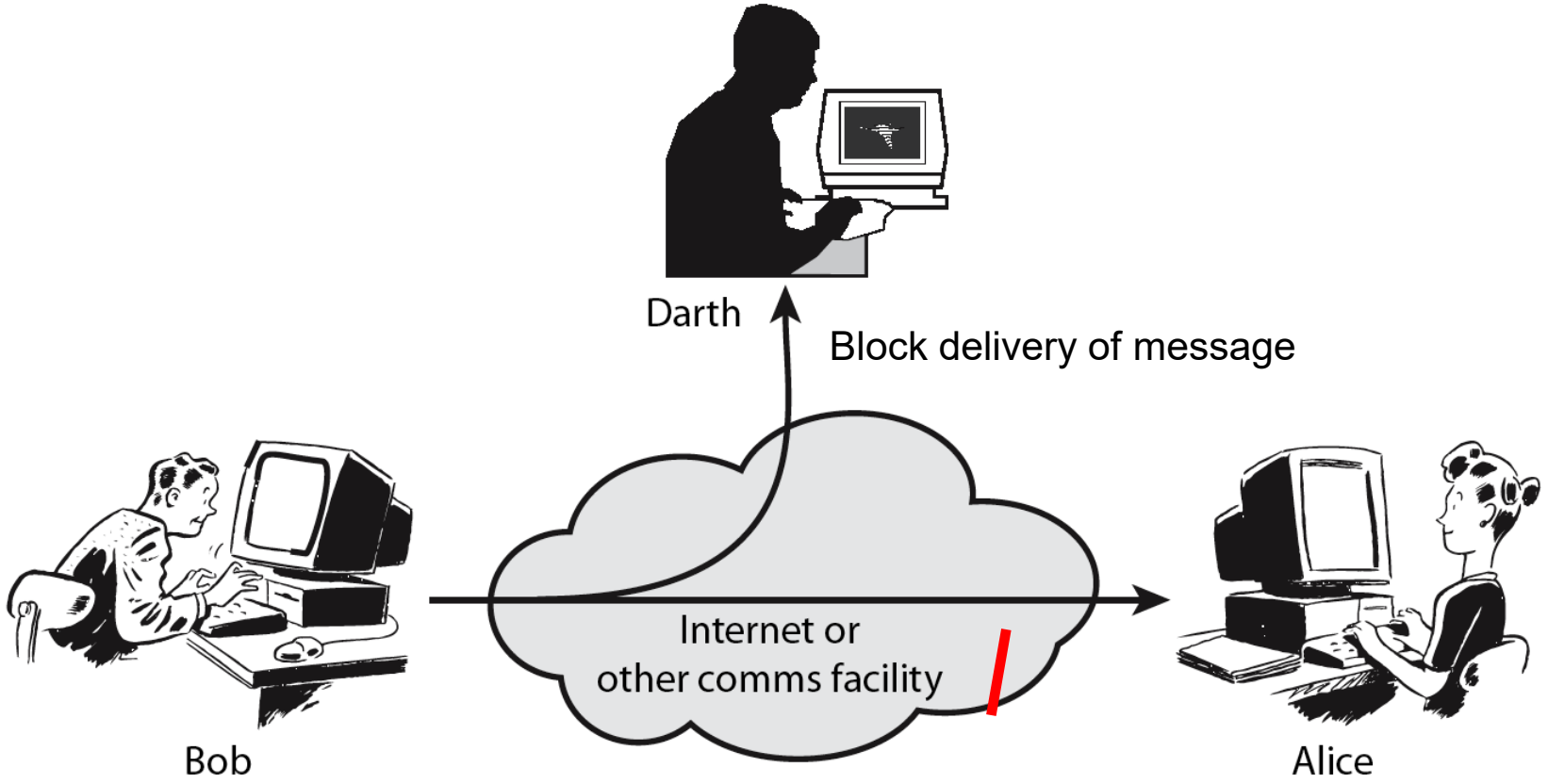
2. Serangan aktif (*active attack*)

- 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

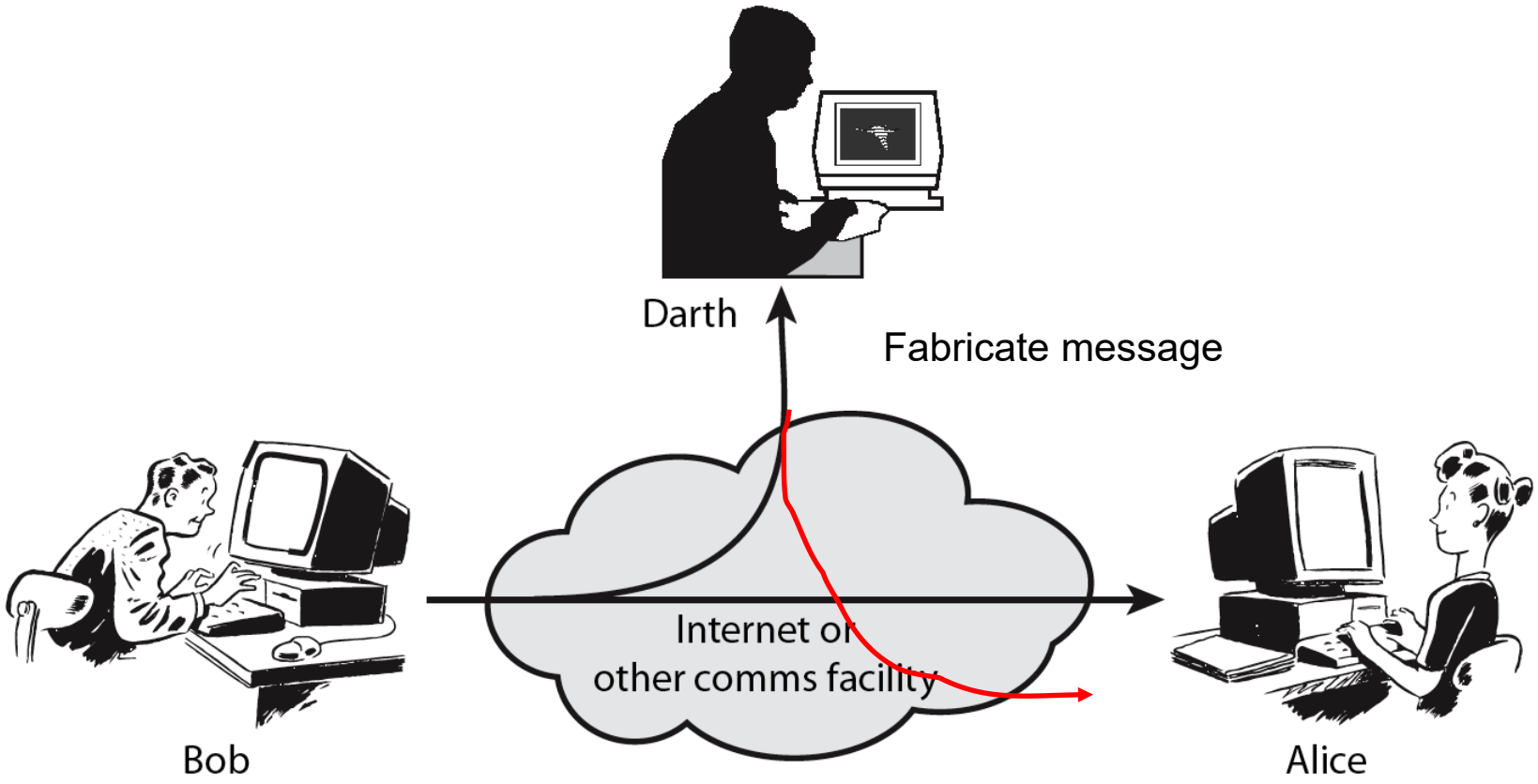
- Contoh: *man-in-the-middle attack*

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



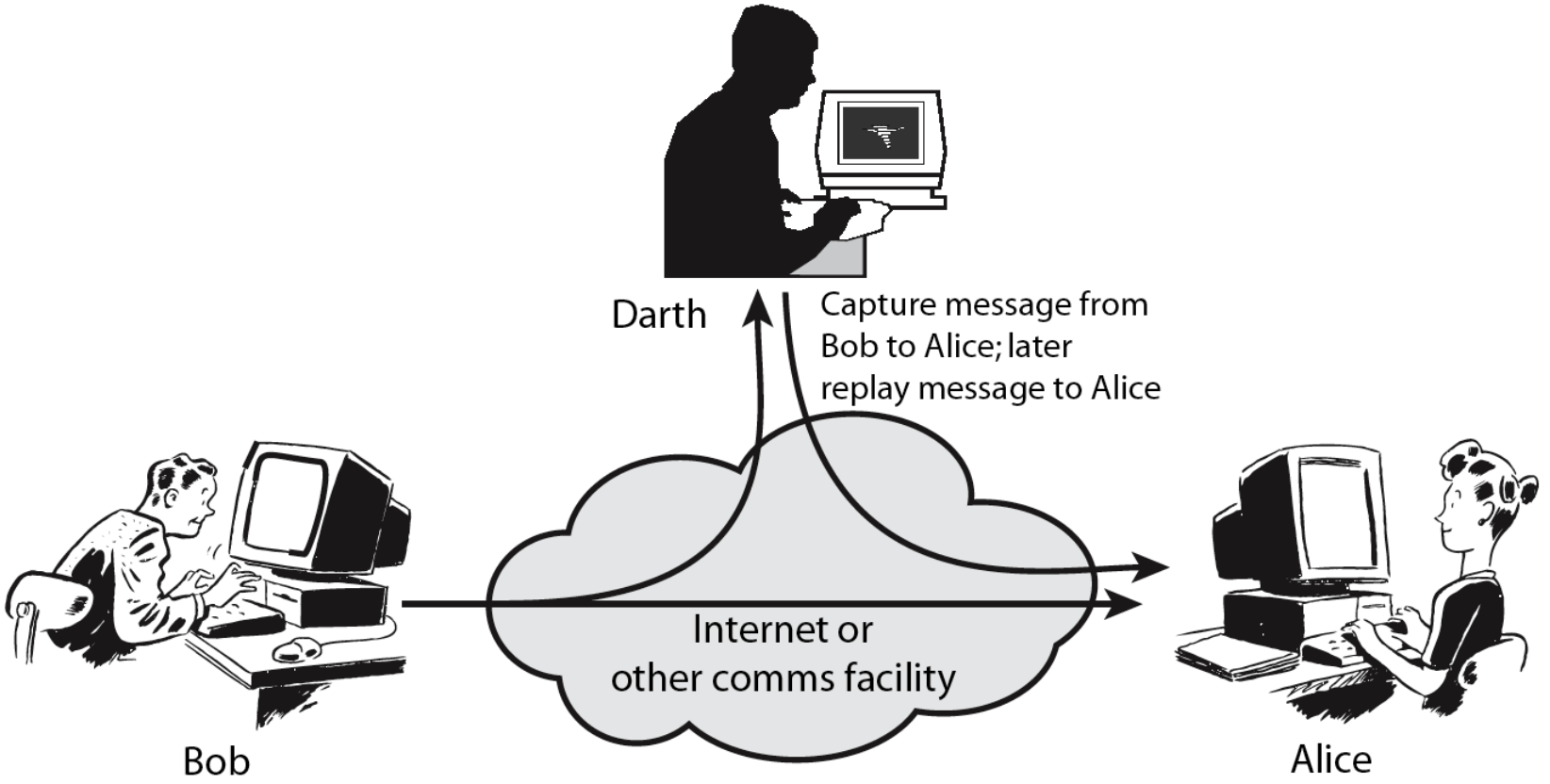
Active Attack: Interruption

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



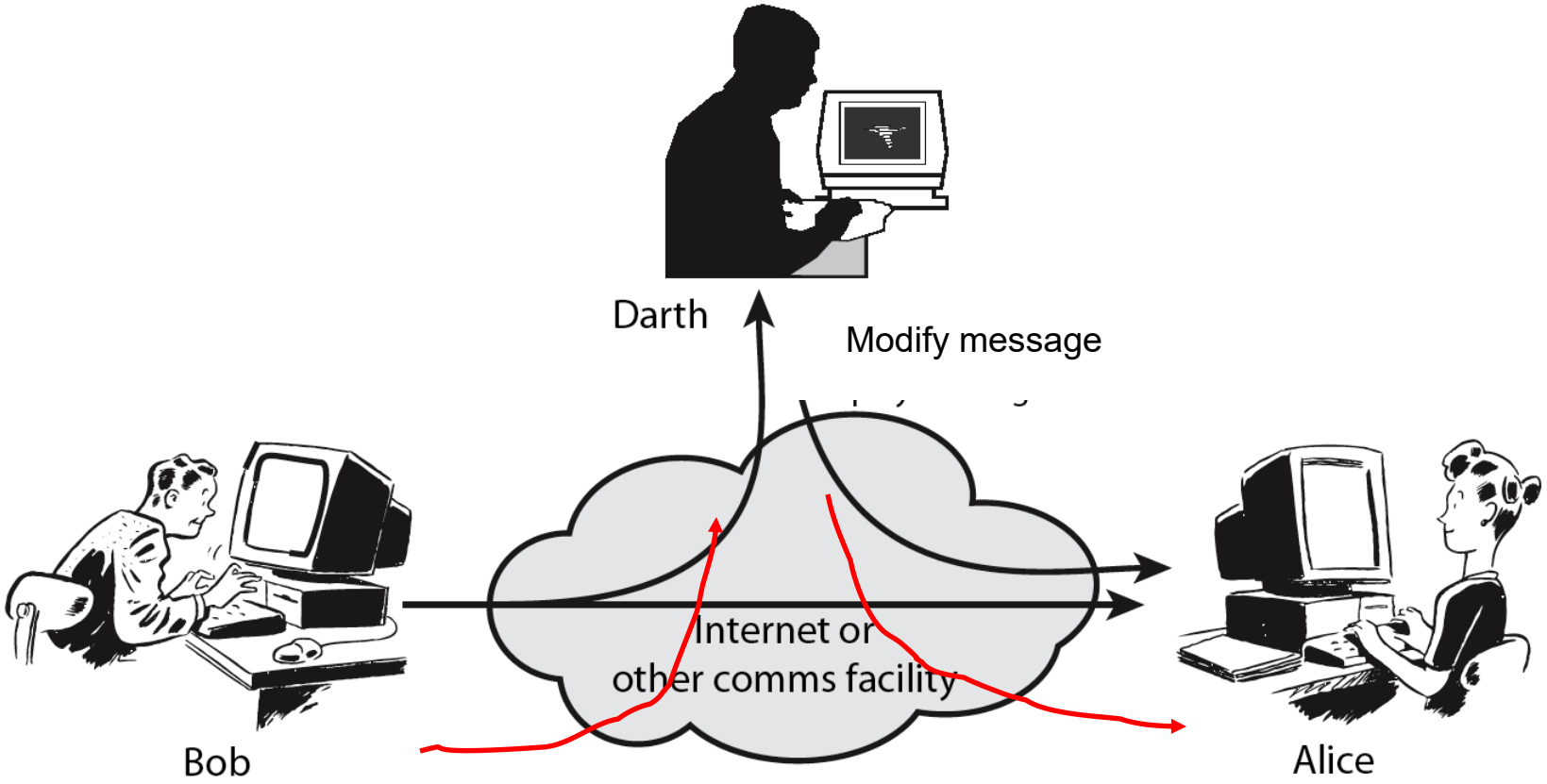
Active Attack: Fabrication

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



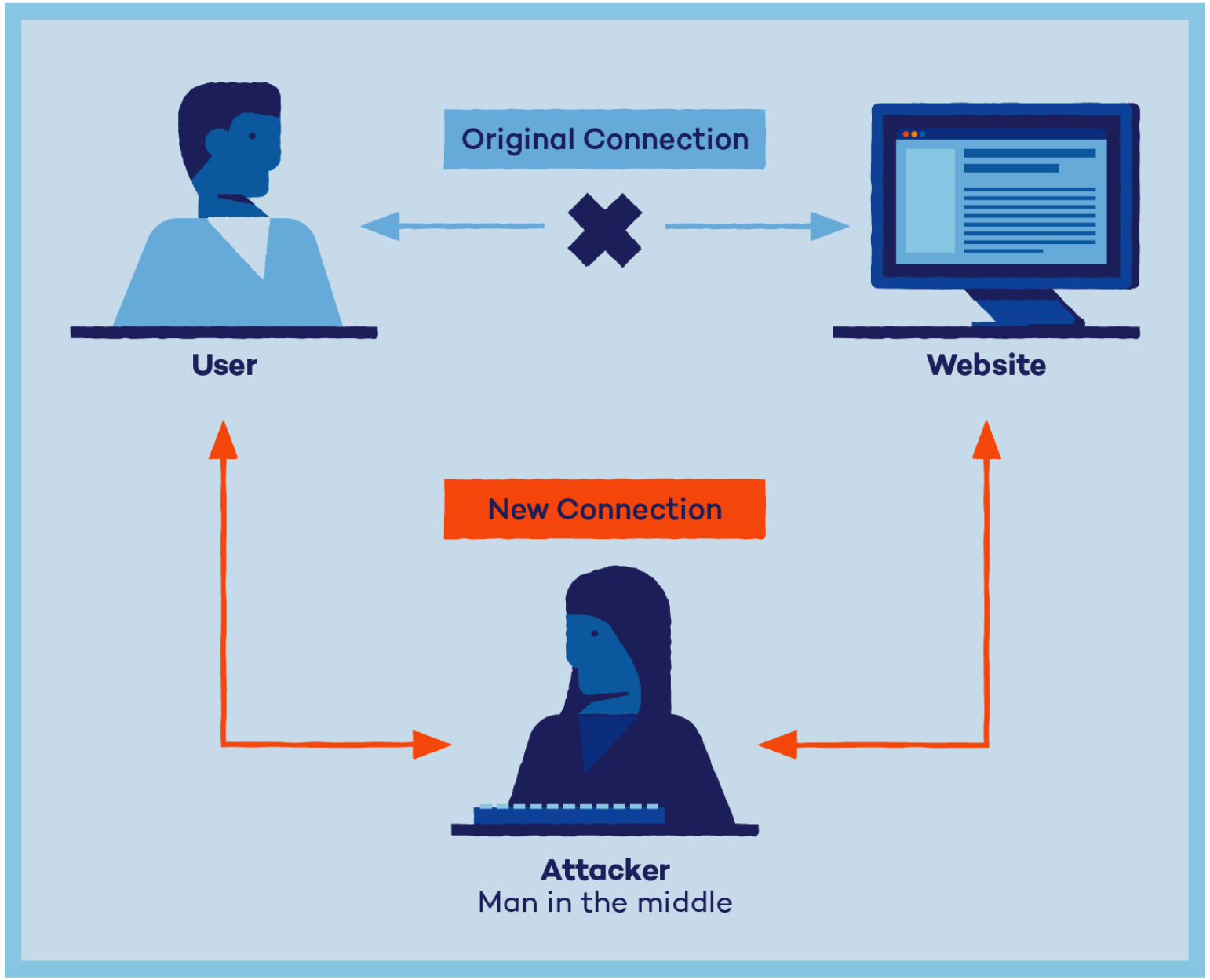
Active Attack: Replay

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

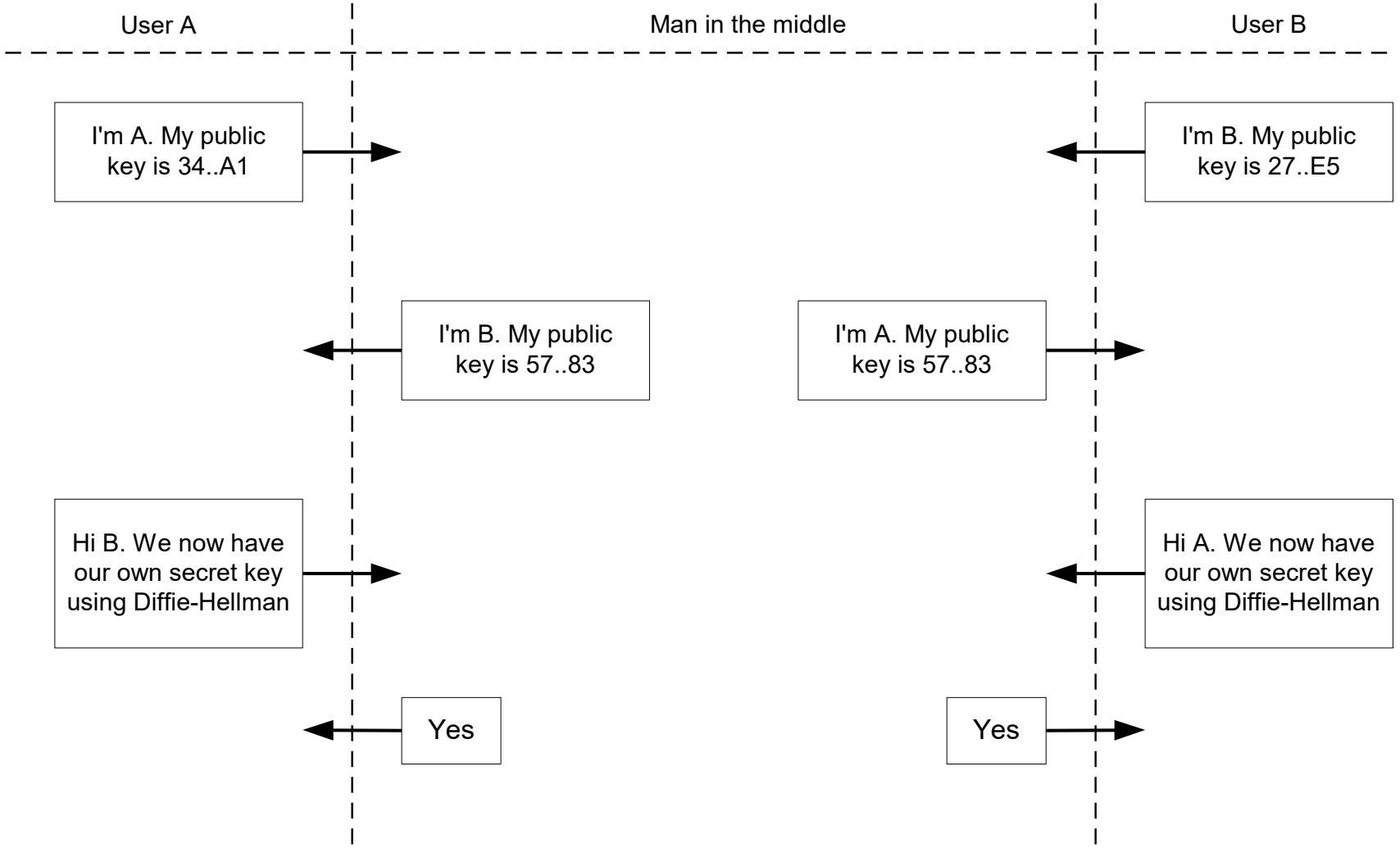


Active Attack: Modification

Man-in-the-middle-attack

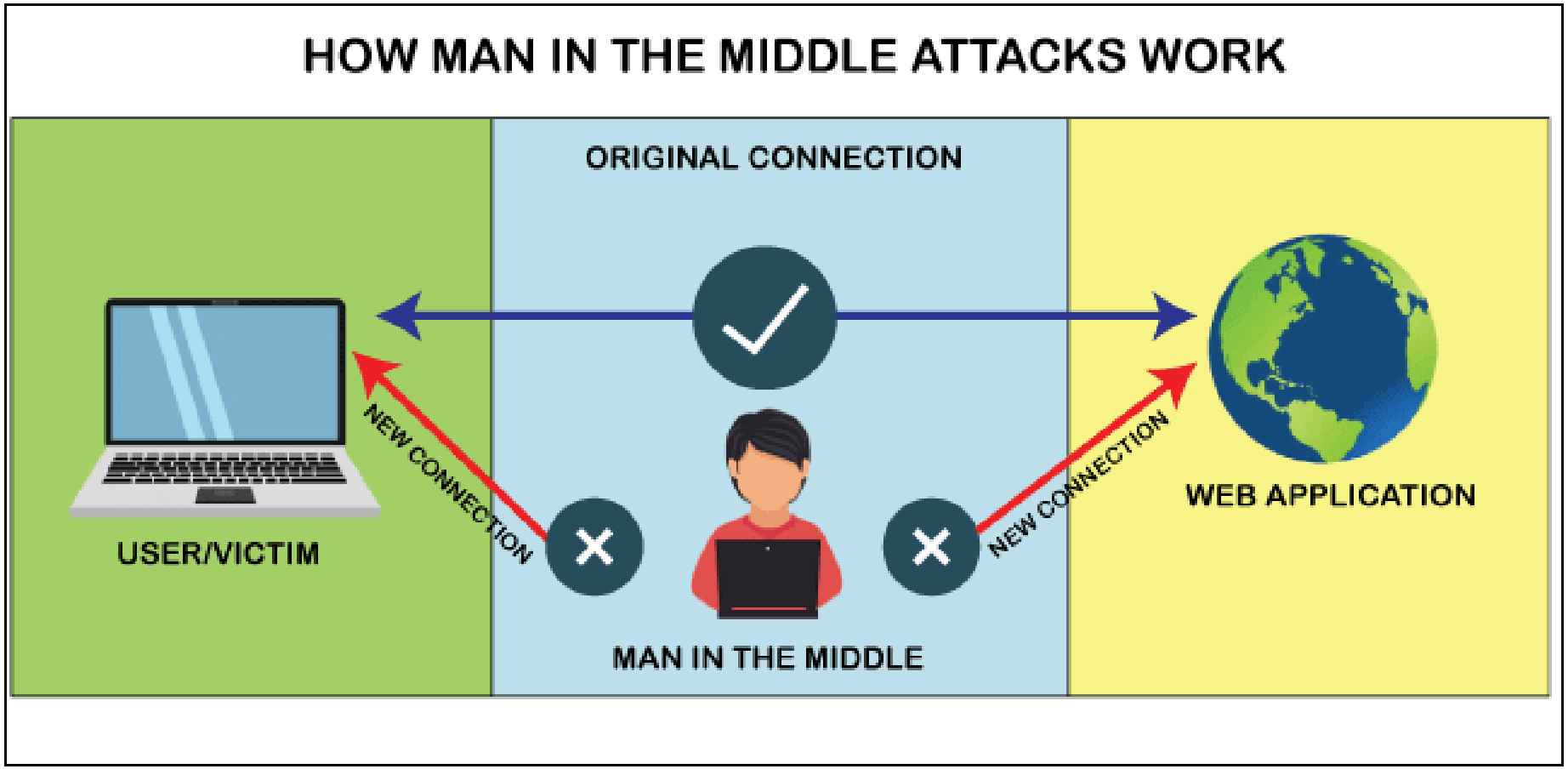


Man-in-the-middle-attack



Man-in-the-middle-attack

Serangan aktif yang berbahaya



Serangan lainnya: *Side-Channel Attack*

- Serangan ini tidak terutama menyerang algoritma matematisnya, tetapi memanfaatkan informasi yang bocor ketika algoritma dijalankan pada perangkat fisik.
- Ada beberapa macam *side-channel attack*:
  1. *Timing Attack*
     Memanfaatkan perbedaan waktu eksekusi untuk memperoleh informasi tentang kunci.
  2. *Power Analysis Attack*
     Menganalisis konsumsi daya perangkat ketika melakukan operasi kriptografi. Contohnya:
     - *Simple Power Analysis (SPA)*
     - *Differential Power Analysis (DPA)*
  3. *Electromagnetic (EM) Attack*
     Menganalisis radiasi elektromagnetik yang dihasilkan perangkat ketika melakukan operasi kriptografi.
  4. *Fault Injection Attack*
     Penyerang sengaja menyebabkan kesalahan pada perangkat atau proses komputasi, kemudian menganalisis *output* yang salah tersebut. Salah satu contohnya adalah *Differential Fault Analysis* (DFA).