

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

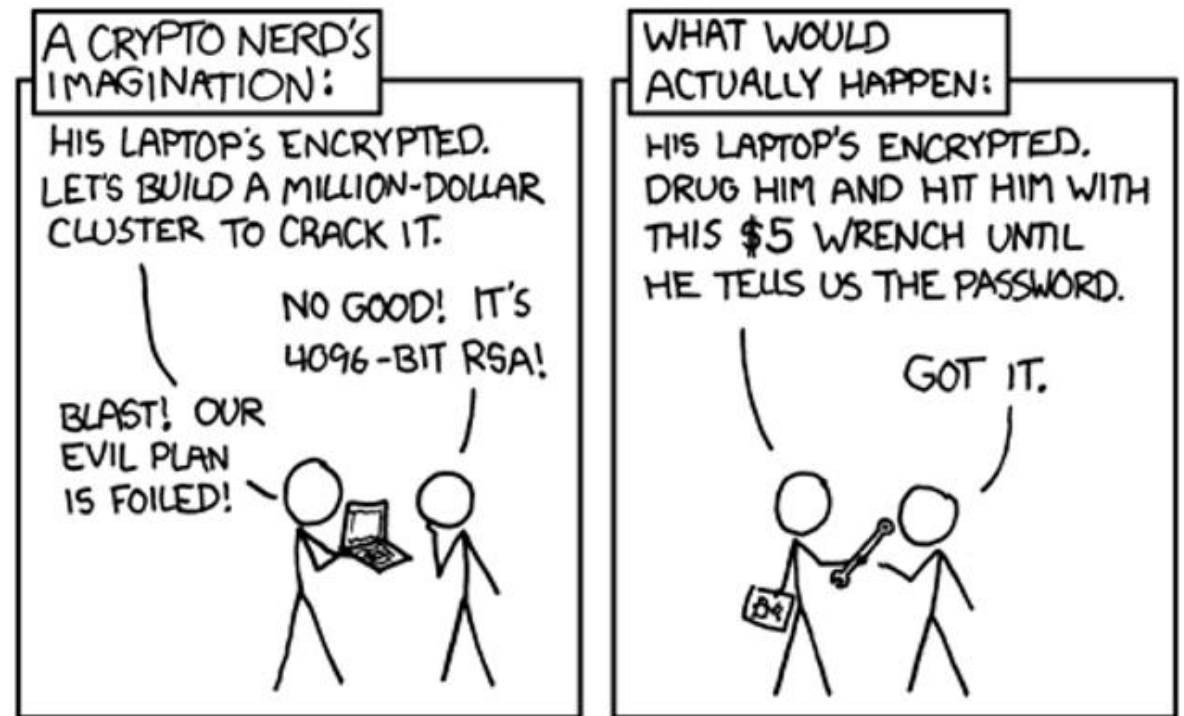
Prinsip Kerckhoff: Semua algoritma kriptografi harus publik; hanya kunci yang rahasia.

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. *Ciphertext-only attack*

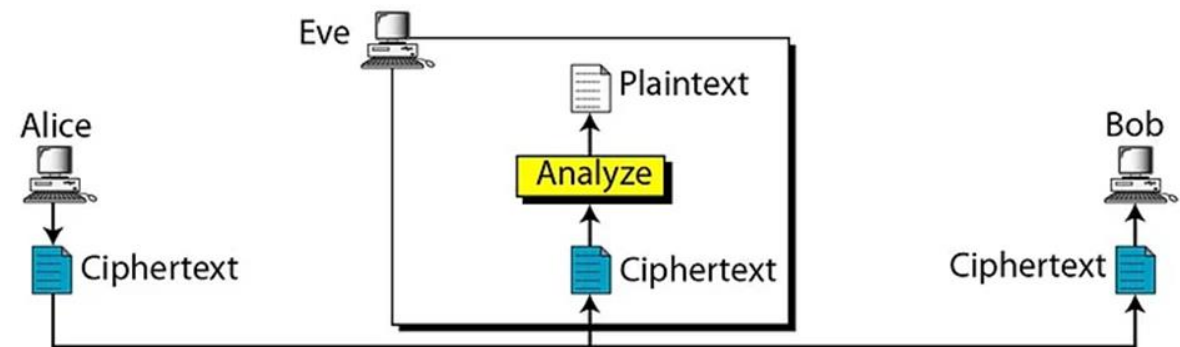
Kriptanalis hanya memiliki cipherteks saja. Ini serangan paling sulit karena informasi yang diperoleh hanya sedikit (cipherteks saja).

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

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

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

Ciphertext-Only Attack



hQIMAw3Jn/nLK/38ARAAAsSXLdhCtzUYKMptNxZImJXwhhIRm3QxfuyHjJ93ASylE
e+6ABkuyFLJhiKryxp/JmS/alMPfF7hx2aTgovagaPzTwTV1jo6If2mhdCl6keed
1Iz7C0f6jHIqq9d8g0bWDyvELEipn5LNDTX3Xp2Csx5ojRB2wckrUt1l1Xyj8G0H
4DQUYbINRmJVulJJc/acGvgOze66pHuRgSCxxHDscefjXenh/XejSYTo7aMi+Es7
DCcD49zH6ZLDQN6BlN9q2oFI8QIhQ2y1QJbatldWi/4yYwLkZcLKRSm8eo/gNCdL
h9MncXBBSfgbvbu67CDZ9G05geZOn3LzQOpJ8hrZq/6K/uMcUKEZjW3RCO0T754f
E5zYe1wUgtwS/lmQ2w5PQF/89bpshtDSYuL1fZgzrsE6DwophuCri5zwCGbEKlsI
g6REIETFbZ2aCL4N2pZVunCIEuoP0zgEB6+M9egdpyxMsMqEBVg3AH7SalAtEguP
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!

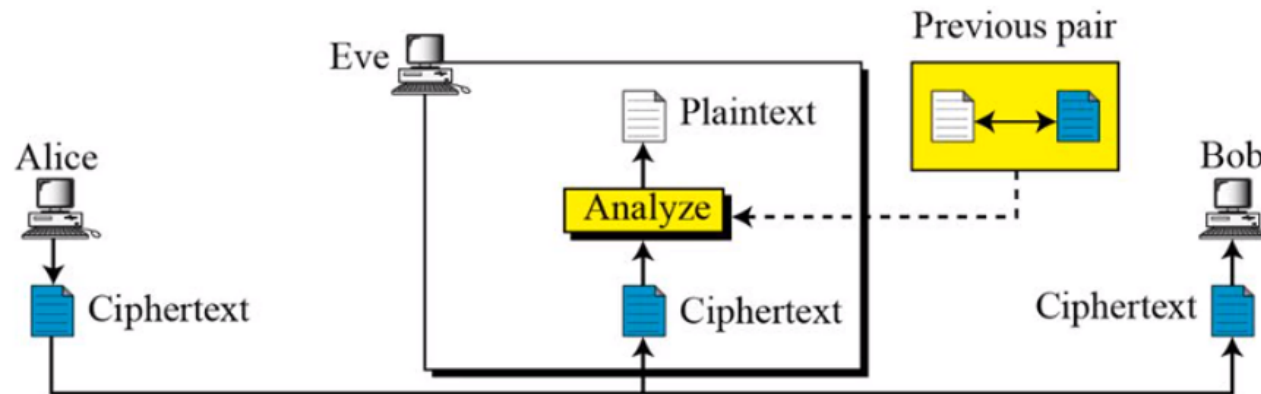
2. *Known-plaintext attack*

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

Known-Plaintext Attack



- 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, kriptanalisis dapat menemukan kunci enkripsi (lihat pembahasan *cipher* klasik *affine cipher* dan *hill cipher*)

Dengan hormat

TFJOXUPOUXYTTRDSXQMONIYPEUFJDQUBGIMOCJQTNBEHCZEKROV
BNTWLMVXMOWZLUCHOXYGSKBQGUAOBQZKIXYJIETSWVXHVKCUAOT
OFYIZAKJGXAAGQTRVFDZAJNQDUIWZCMYWNFIUPYMCZXIAKYUCQ
IAZPIQMGAMGUAKKKHMWKDUXQDUAAKYOWEHLJPWYFKXSARBL LHGA
JKTQNTRTPWSCIZASCGSLKVDHTUZSWBNBTJGYYUPQMFSYZAUTOQC
DNGQMF SRLRTUWEMKADIVYLTJKFHLKJUWTSSHMHJFGTRIBYIDAHQ
EPMPIQCROWDYRYZNSPNOJHQVKKTOCBPNFAJNLYJZNVBAYJWRGMC
HJPWBDHHTPOXSIJVQWDMSIGMTRVEVXDILKVAYTNUNJXEZLAPGYE
TRVZNVH SVWL GICDXQFOALDVPASUSYXPFHUWTILUQHTJQVGWFS PA
EKBRBNI INYKHNTNUKJVDHVLXQKUZNVQXUOZZOJZYNPIVYSVFVTZ
MMUUPWTGHRIOWCBKZYAGUMRCKHIQZSIGISPGBXPYXMOAWGAGHQV
UWTEIGPBMOMBWIO PQEVKMRQATNB MILHHLVUXGMOUWTZCLBKGWIJ
HFRNGOSCMUHDWHBB

wassalam

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

KHOOR ZRUOG

dan juga tahu plainteks-nya seharusnya mengandung kata:

HELLO

Dari pasangan $H \rightarrow K$, penyerang tahu pergeseran $(k) = 3$.

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

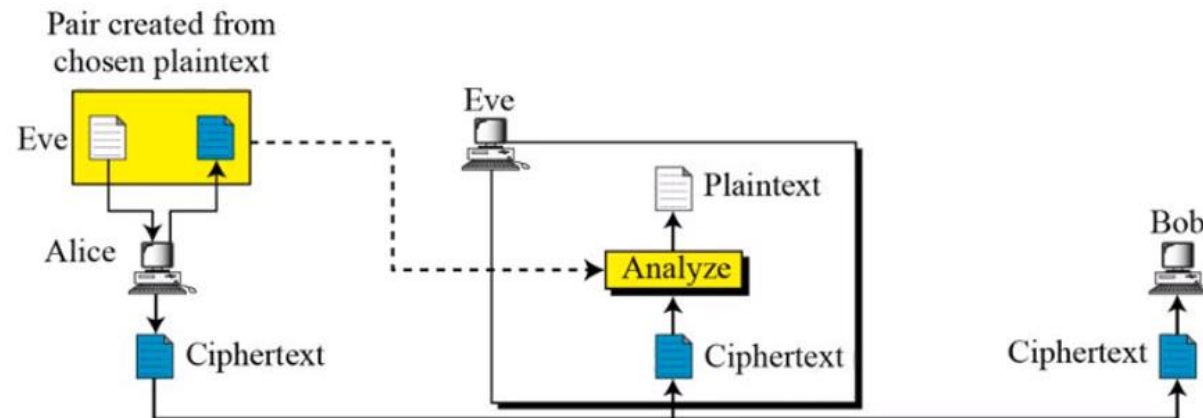
3. *Chosen-plaintext attack*

Kriptanalisis 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: $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \dots, P_i, C_i = E_k(P_i)$
di mana kriptanalisis dapat memilih diantara $P_1, P_2, \dots, P_i$

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

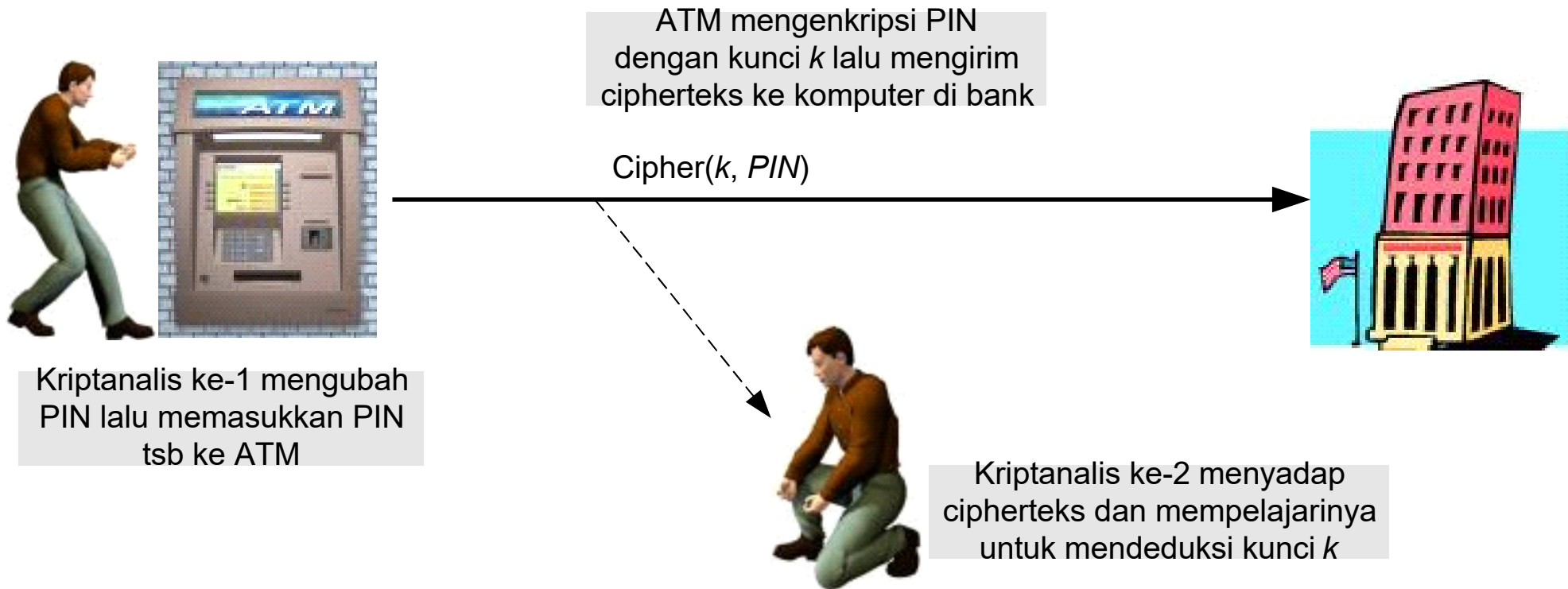
Dari sini terlihat huruf A $\rightarrow$ F, berarti pergeseran = 5.

Dengan kunci itu, penyerang bisa mendekripsi semua cipherteks lain.

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

$$C = (P + K) \text{ mod } 26$$

- Penyerang memilih plaintext $P = \text{AAAAAAAA...}$ ($= 000000...$).
- Maka cipherteks yang keluar, C , pasti sama dengan K ($C = K$).
- Dengan mengetahui K , cipherteks 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 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, ciphertexts lain bisa langsung dipecahkan.

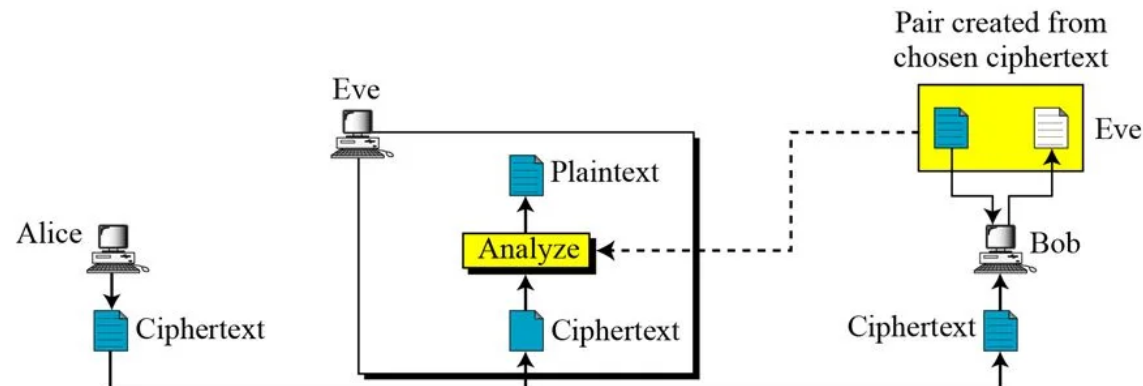
5. Chosen-ciphertext attack

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

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

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

Chosen-Ciphertext Attack



- 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

$$C = E_K(P) \text{ , } P = \text{plainteks, } K = \text{kunci, } C = \text{cipherteks}$$

- Dalam *key-recovery attack*, penyerang berusaha menemukan K.
- Contoh: Misalkan sebuah cipher menggunakan kunci 16 karakter. Penyerang melakukan pencarian kunci secara *brute force*:

$K_1 \rightarrow \text{dekripsi } C \rightarrow P_1$

$K_2 \rightarrow \text{dekripsi } C \rightarrow P_2$

$K_3 \rightarrow \text{dekripsi } C \rightarrow P_3$

$\vdots$

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.

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 materi 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

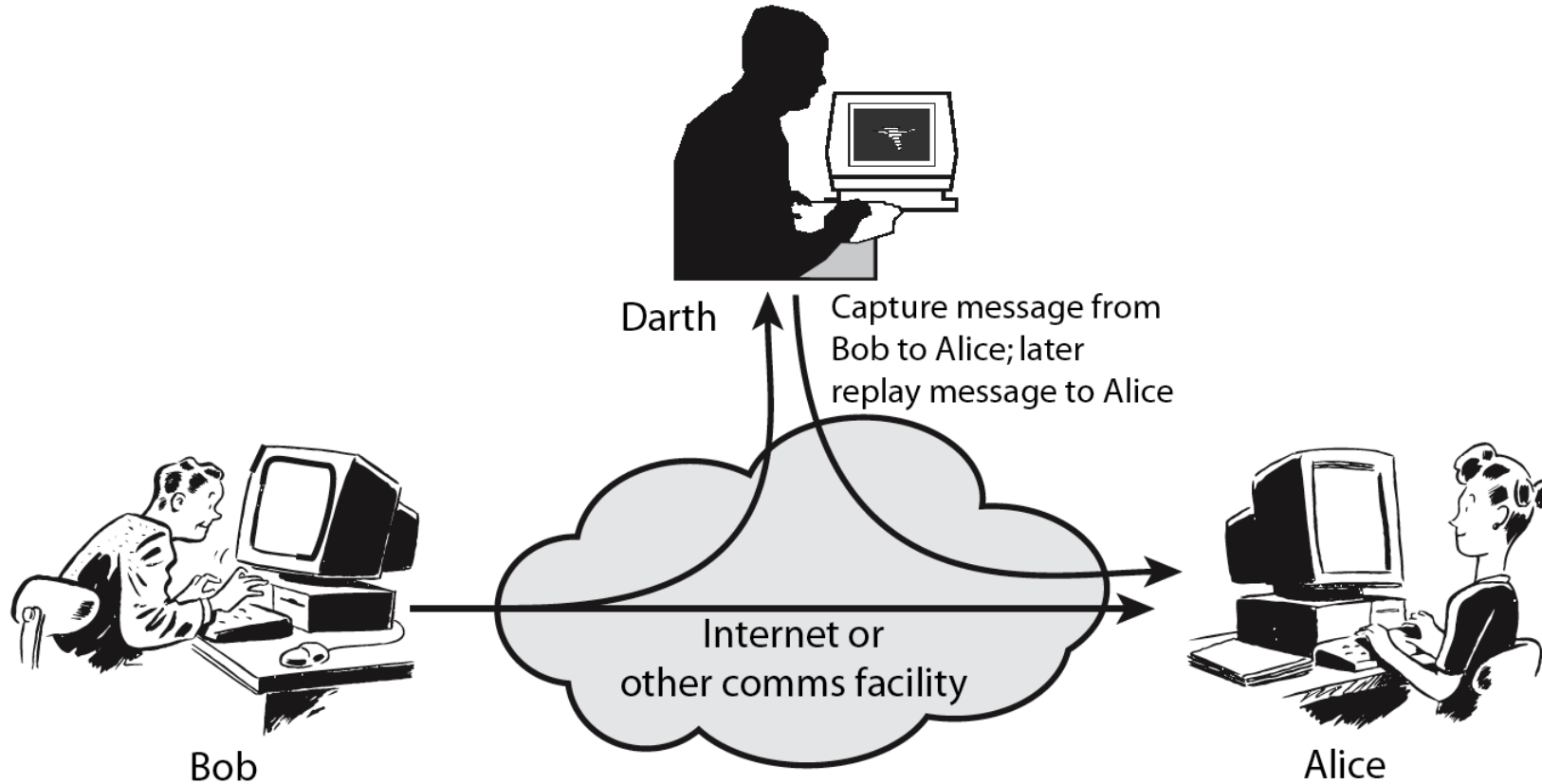
- Tujuan: menggunakan kembali pesan sah yang pernah dikirim sebelumnya untuk menyebabkan suatu tindakan terjadi lagi.
- Yang menarik, pada *replay attack* penyerang tidak perlu memecahkan enkripsi.
- Untuk mencegah *replay attack*, biasanya digunakan mekanisme yang membuat setiap pesan mempunyai keunikan 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.



Replay Attack

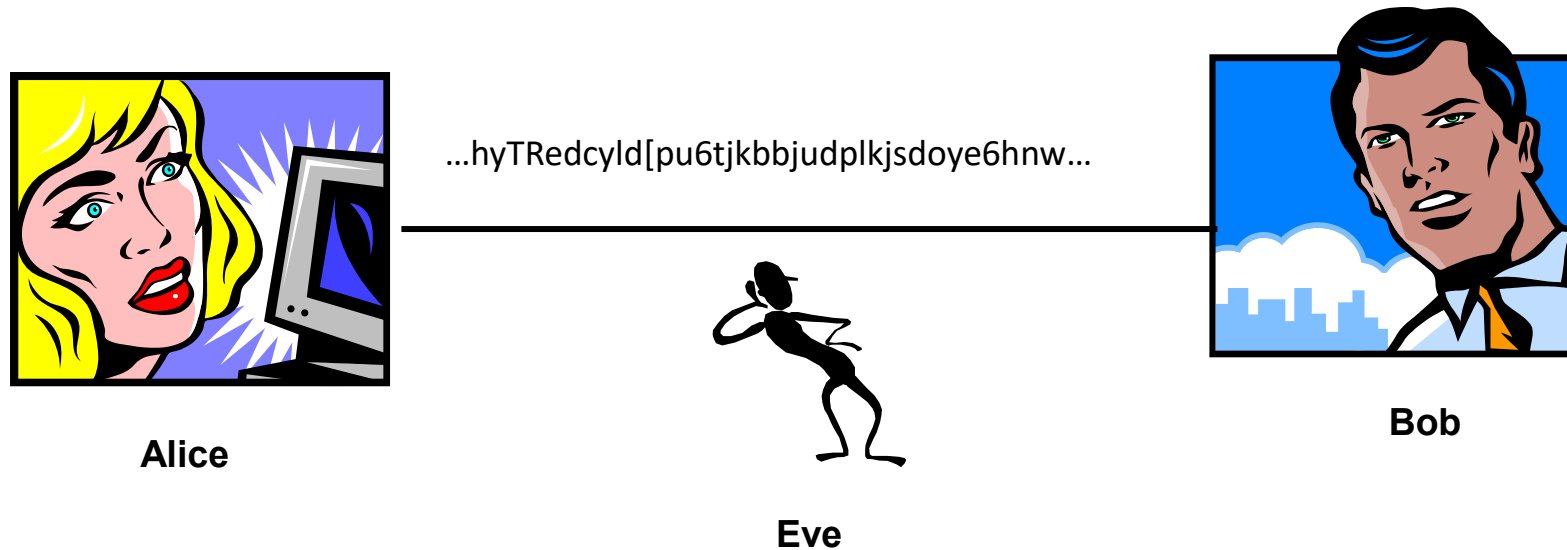
Jenis-jenis Serangan (3)

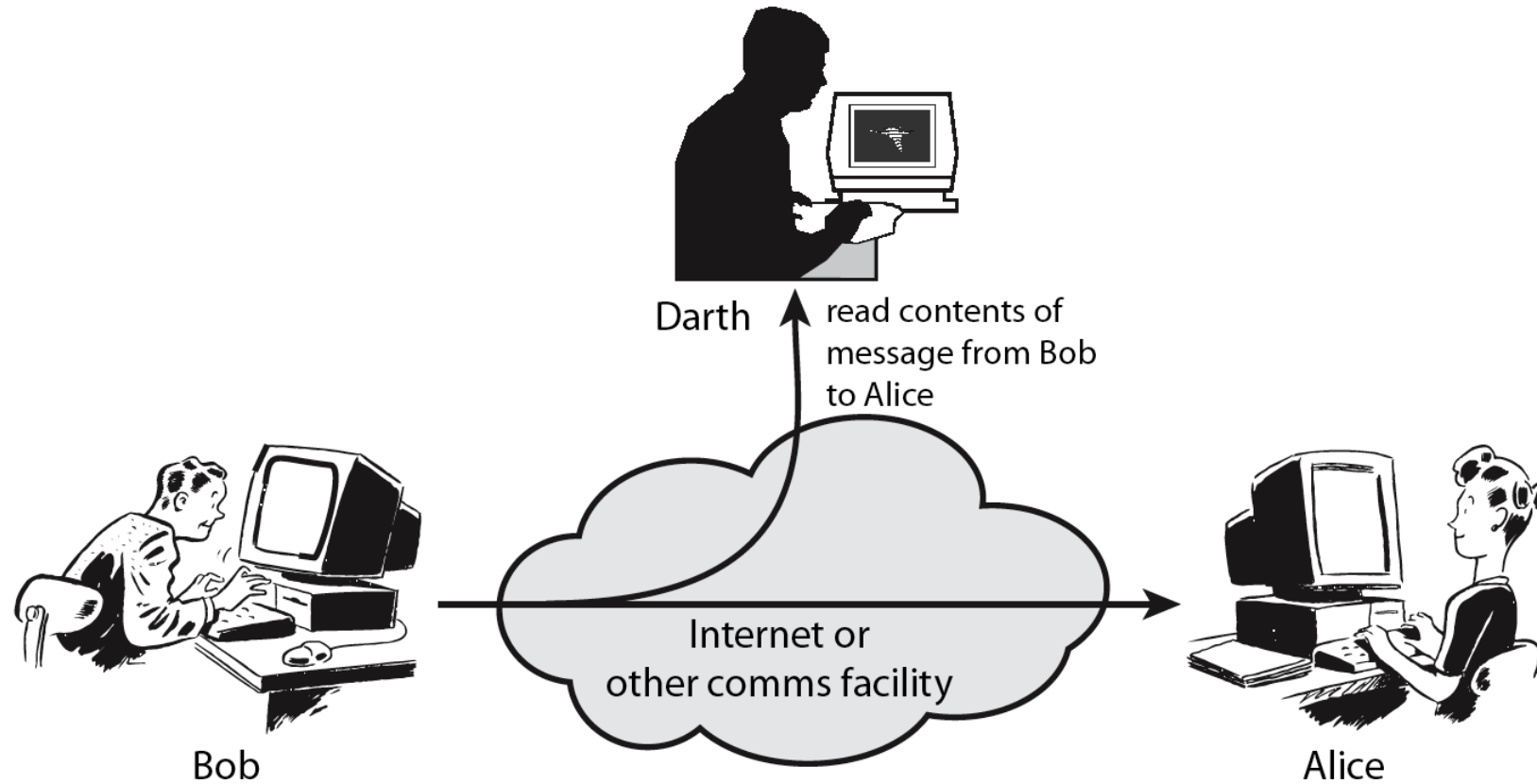
Berdasarkan keterlibatan penyerang dalam komunikasi pesan:

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

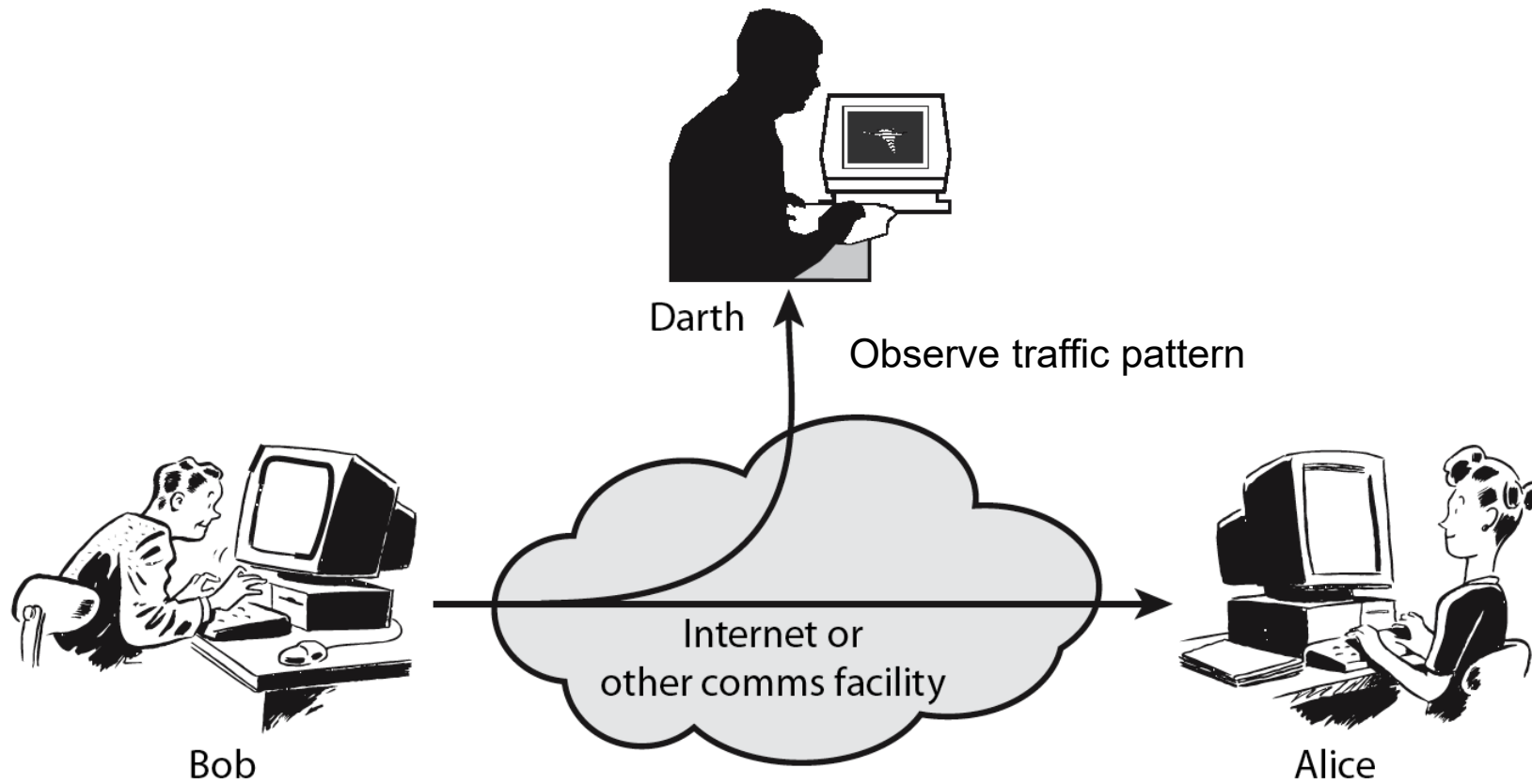
1. Serangan pasif (*passive attack*)

- 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





Passive Attack : Interception



Passive Attack : Traffic Analysis

Screenshot Wireshark (memantau network traffic)

The screenshot shows the Wireshark interface with a packet capture from a Marvell Yukon Ethernet Controller. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 1) is a NetBIOS over IPX packet, showing details for IEEE 802.3 Ethernet, Logical-Link Control, Internetwork Packet exchange, and NetBIOS over IPX. The packet data is displayed in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Length	Info
27	2.79561600	167.205.33.90	255.255.255.255	DB-LSP-	154	Dropbox LAN sync Discovery Protocol
28	2.96394400	167.205.33.14	167.205.33.255	NBNS	92	Name query NB CORPORATE<00>
29	3.06038000	00000000.00804837fc	00000000.ffffffff	NBIPX	98	Find name WORKGROUP<00>
30	3.06039600	4416db43.0000000000	00000000.00804837fc	NBIPX	98	Name recognized WORKGROUP<00>
31	3.06809700	167.205.33.14	167.205.33.255	NBNS	92	Name query NB CORPORATE<00>
32	3.09102200	167.205.33.88	167.205.33.255	NBNS	92	Name query NB PRINTERBASDAD<00>
33	3.24244300	AsustekC_10:09:66	Broadcast	ARP	60	who has 167.205.33.12? Tell 167.205.33.2
34	3.30945700	Cisco-Li_11:0d:0f	Spanning-tree-(for- STP	STP	60	RST. Root = 32768/0/00:22:6b:10:d8:3d Co
35	3.35738600	167.205.33.121	167.205.33.255	NBNS	92	Name query NB SUDARMAN<20>
36	3.47038100	167.205.33.61	50.62.3.118	TCP	62	mctet-gateway > https [SYN] Seq=0 win=655
37	3.60684800	IntelCor_c2:e0:11	Broadcast	ARP	60	who has 167.205.33.116? Tell 167.205.33.1
38	3.71257800	167.205.33.14	167.205.33.255	NBNS	92	Name query NB CORPORATE<00>
39	3.80628200	167.205.33.14	167.205.33.255	NBNS	92	Name query NB CORPORATE<00>
40	3.83975500	00000000.00804837fc	00000000.ffffffff	BROWSEF	176	Request Announcement DAPUR
41	3.83996700	167.205.33.100	167.205.33.255	BROWSEF	243	Host Announcement BUGI-WIBOWO, workstatio

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

- IEEE 802.3 Ethernet
- Logical-Link Control
- Internetwork Packet exchange
- NetBIOS over IPX

0000 ff ff ff ff ff ff 00 80 48 37 fc 30 00 54 e0 e0 H7.0.T..
0010 03 ff ff 00 50 00 14 00 00 00 00 ff ff ff ffP... ..
0020 ff 04 55 00 00 00 00 00 80 48 37 fc 30 04 55 00 ..U..... .H7.0.U..
0030 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 c0
0050 02 01 02 5f 5f 4d 52 42 52 4f 52 45 5f 5f 02 MSB POWER

Marvell Yukon Ethernet Controller (Microsoft's Pa... Packets: 729 Displayed: 729 Marked: 0 Profile: Default

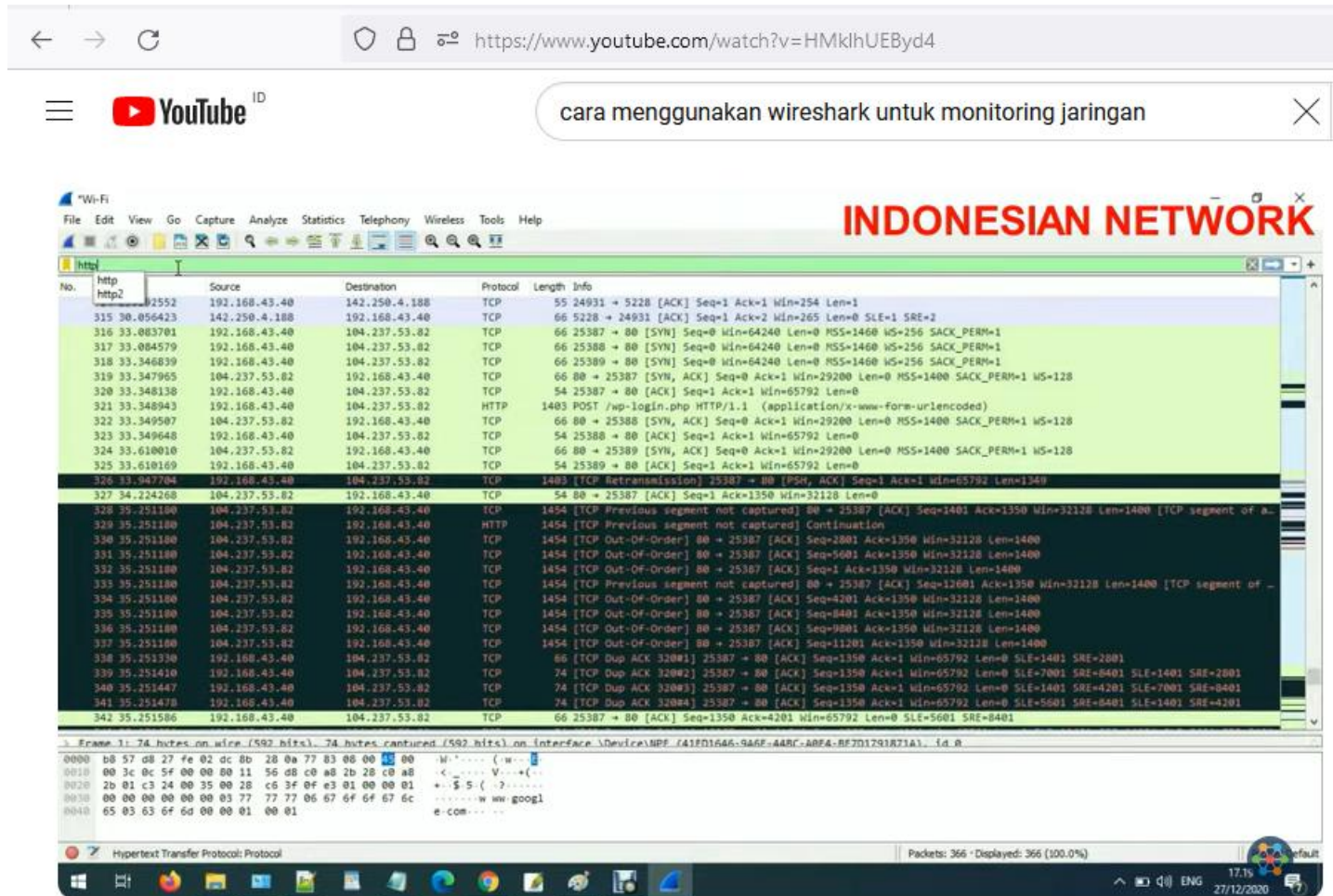
http.request.method=="POST"						
No.	Time	Source	Destination	Protocol	Length	Info
1034	8.148165	172.99.96.253	160.153.129.234	HTTP	617	POST /sign
[Full request URI: http://www.sababank.com/signin.php] [HTTP request 1/1] [Response in frame: 1129] File Data: 53 bytes HTML Form URL Encoded: application/x-www-form-urlencoded Form item: "username" = "Ibrahim_Diyeb" Form item: "password" = "yemen_123" Form item: "actn" = "signin"						
01a0	63 6f 64 65 64 0d 0a 43	6f 6e 74 65 6e 74 2d 4c	coded..Content-L			
01b0	65 6e 67 74 68 3a 20 35	33 0d 0a 43 6f 6f 6b 69	ength: 5 3..Cooki			
01c0	65 3a 20 50 48 50 53 45	53 53 49 44 3d 34 31 32	e: PHPSESSID=412			
01d0	33 35 34 31 32 30 63 35	36 37 34 35 61 63 66 34	354120c5 6745acf4			
01e0	31 62 38 65 32 39 36 34	63 32 62 65 35 3b 20 6c	1b8e2964 c2be5; l			
01f0	61 6e 67 3d 61 72 61 62	69 63 0d 0a 43 6f 6e 6e	ang=arabic..Conn			
0200	65 63 74 69 6f 6e 3a 20	6b 65 65 70 2d 61 6c 69	ection: keep-ali			
0210	76 65 0d 0a 55 70 67 72	61 64 65 2d 49 6e 73 65	ve..Upgrade-Inse			
0220	63 75 72 65 2d 52 65 71	75 65 73 74 73 3a 20 31	cure-Requests: 1			
0230	0d 0a 0d 0a 75 73 65 72	6e 61 6d 65 3d 49 62 72	...user name=Ibr			
0240	61 68 69 6d 5f 44 69 79	65 62 26 70 61 73 73 77	ahim_Diyeb&passw			

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 P455WORD PADA LALU LINTAS PACKET JARINGAN:

<https://www.youtube.com/watch?v=HMklhUEByd4>

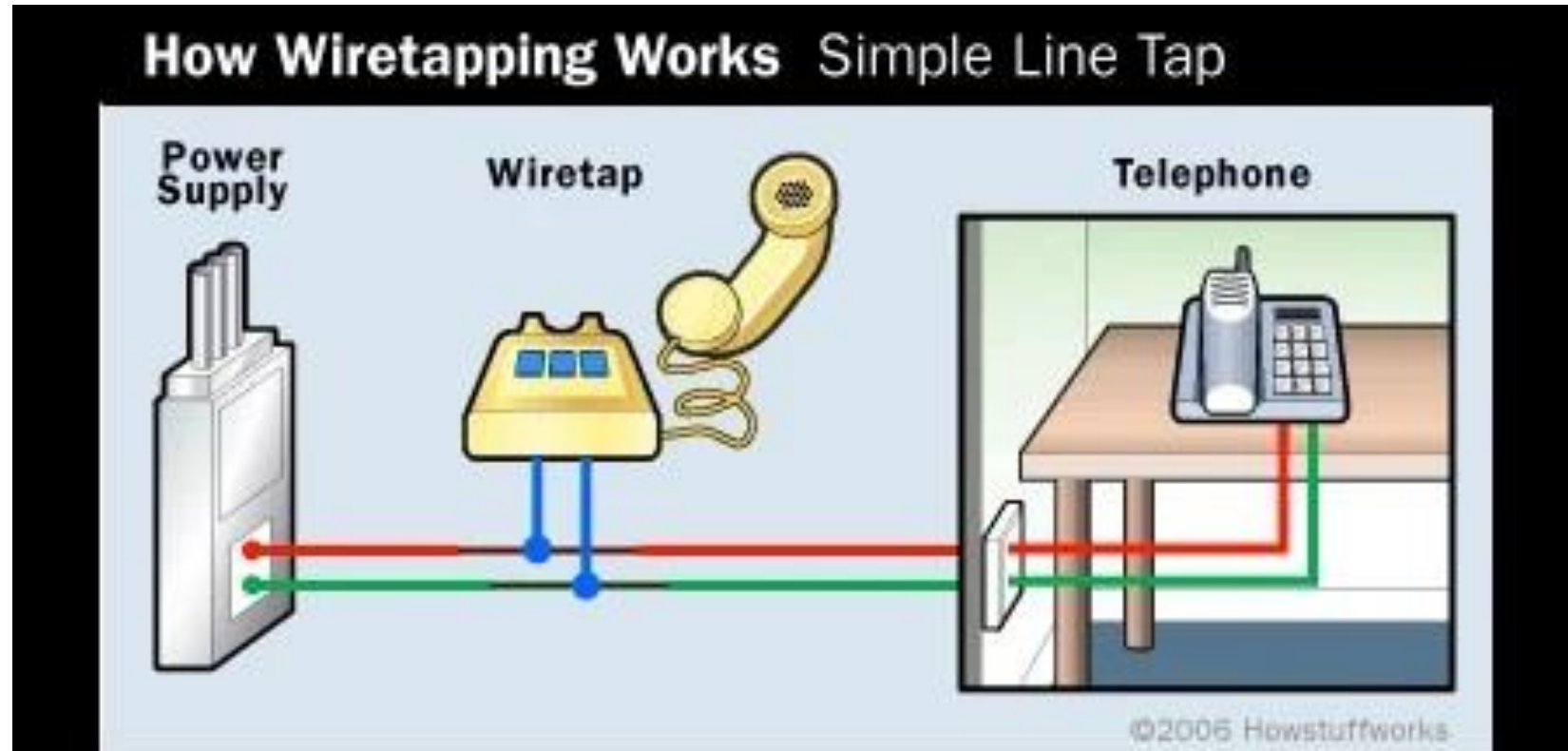


TUTORIAL CARA MENGGUNAKAN WIRESHARK | MENEMUKAN U53RN4M3 P455WORD PADA LALU LINTAS PACKET JARINGAN

Metode penyadapan:

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

- *Wiretapping*

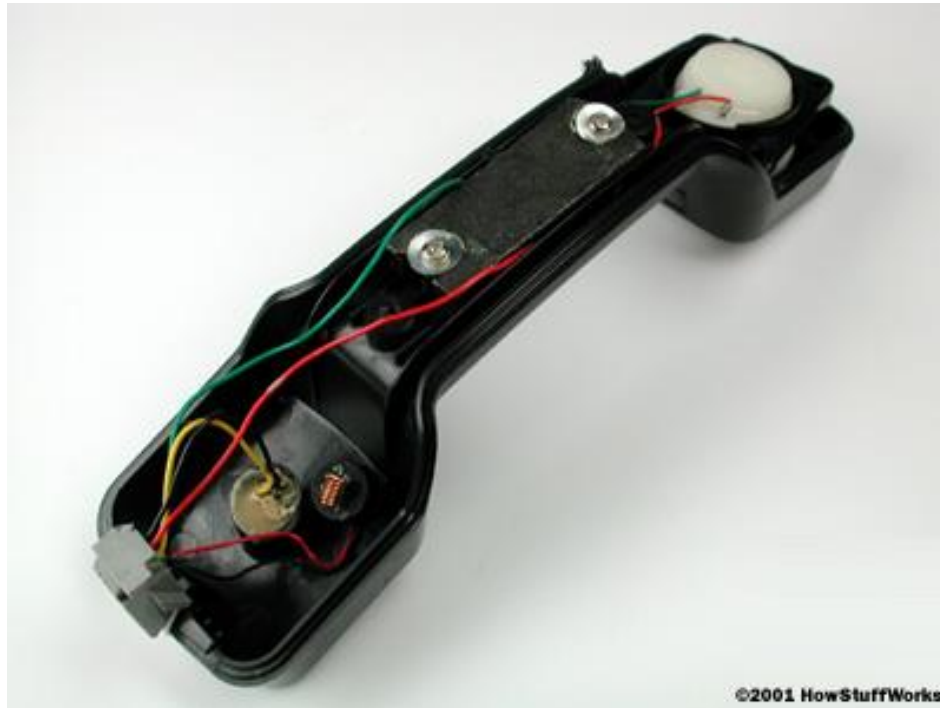


Baca di sini:

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>)

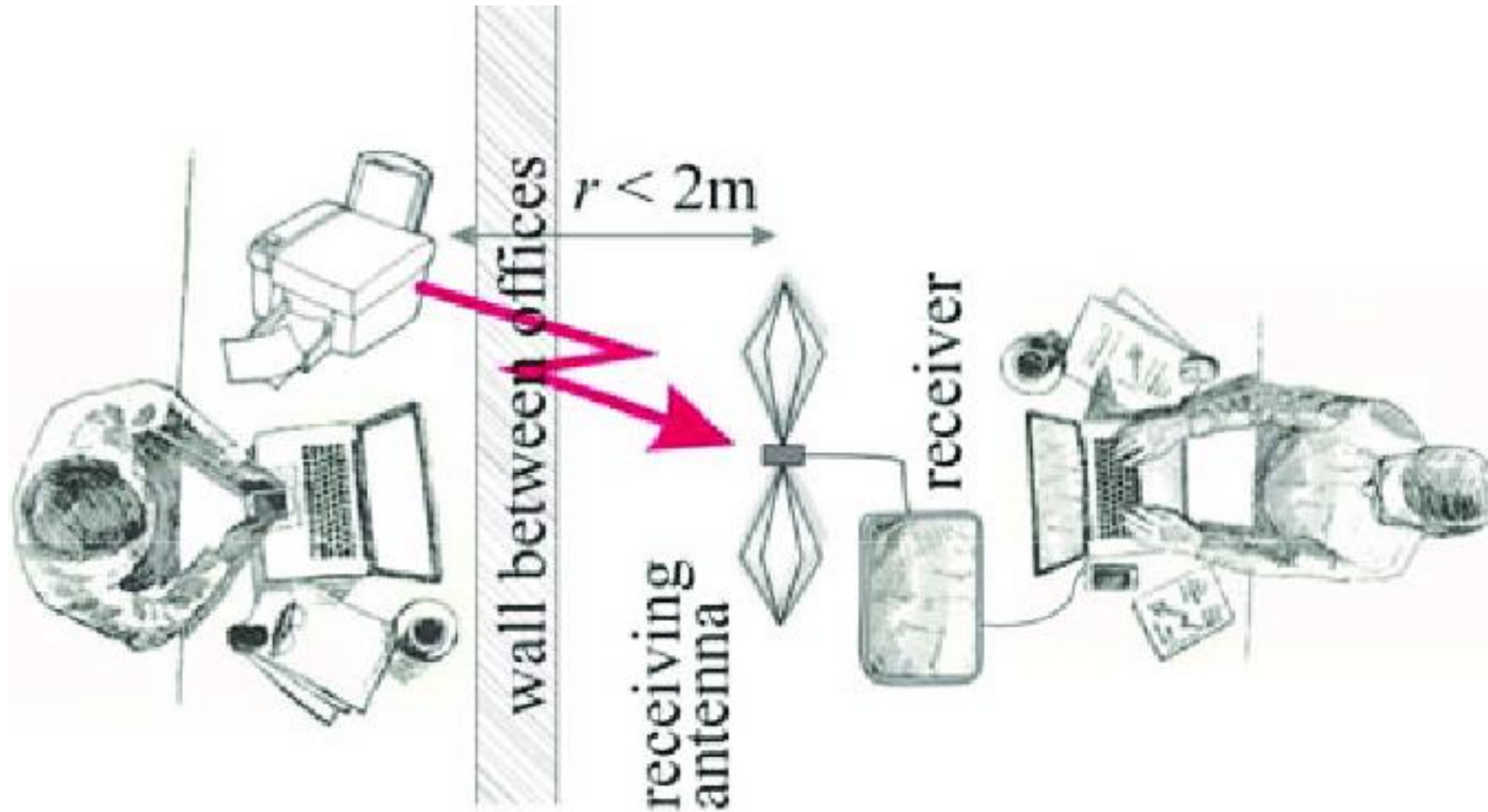


When you open up a phone, you can see that the technology inside is very simple. The simplicity of design makes the phone system vulnerable to surreptitious eavesdropping.



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>

← → ↺ https://www.youtube.com/watch?v=UMu5AhSg-TI

☰ YouTube^{ID}



The video shows a man with a beard and short hair, wearing a black shirt, holding a small black electronic device in his right hand. The device has a screen displaying a yellow waveform on a black background. He is standing in a room with computer monitors and equipment in the background.

INTERCEPT ANY RADIO SIGNAL!!!!

 **andy kirby**
105 rb subscriber

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

<https://www.youtube.com/watch?v=krJJKjYdwgc>

← → ↺ https://www.youtube.com/watch?v=krJJKjYdwgc

☰ YouTube ID

The screenshot displays a complex software interface for GSM sniffing and decryption. It features several overlapping windows:

- Packet Capture Window:** Shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The second packet is highlighted in red.
- Terminal Window:** Displays the command line interface of the 'gr-gsm' software, showing the successful loading of modules and the detection of a Realtek RTL2838UHD radio.
- Frequency Plot:** A graph showing 'Relative Gain (dB)' on the y-axis (ranging from -140 to 0) versus 'Frequency' on the x-axis (ranging from 938.500 to 940.000 MHz). A prominent signal is visible at 938.954 MHz.
- Notes Window:** A text editor on the right side of the interface.
- Video Feed:** A small inset in the bottom right corner shows the presenter, a man wearing a headset and sunglasses.

At the bottom of the interface, there is a status bar indicating 'Loopback: lo: <live capture in progress>', 'Packets: 3 - Displayed: 3 (100.0%)', and 'Profile: Default'.

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



Crazy Danish Hacker
26,9 rb subscriber

Subscribe

2 rb



Bagikan

Download



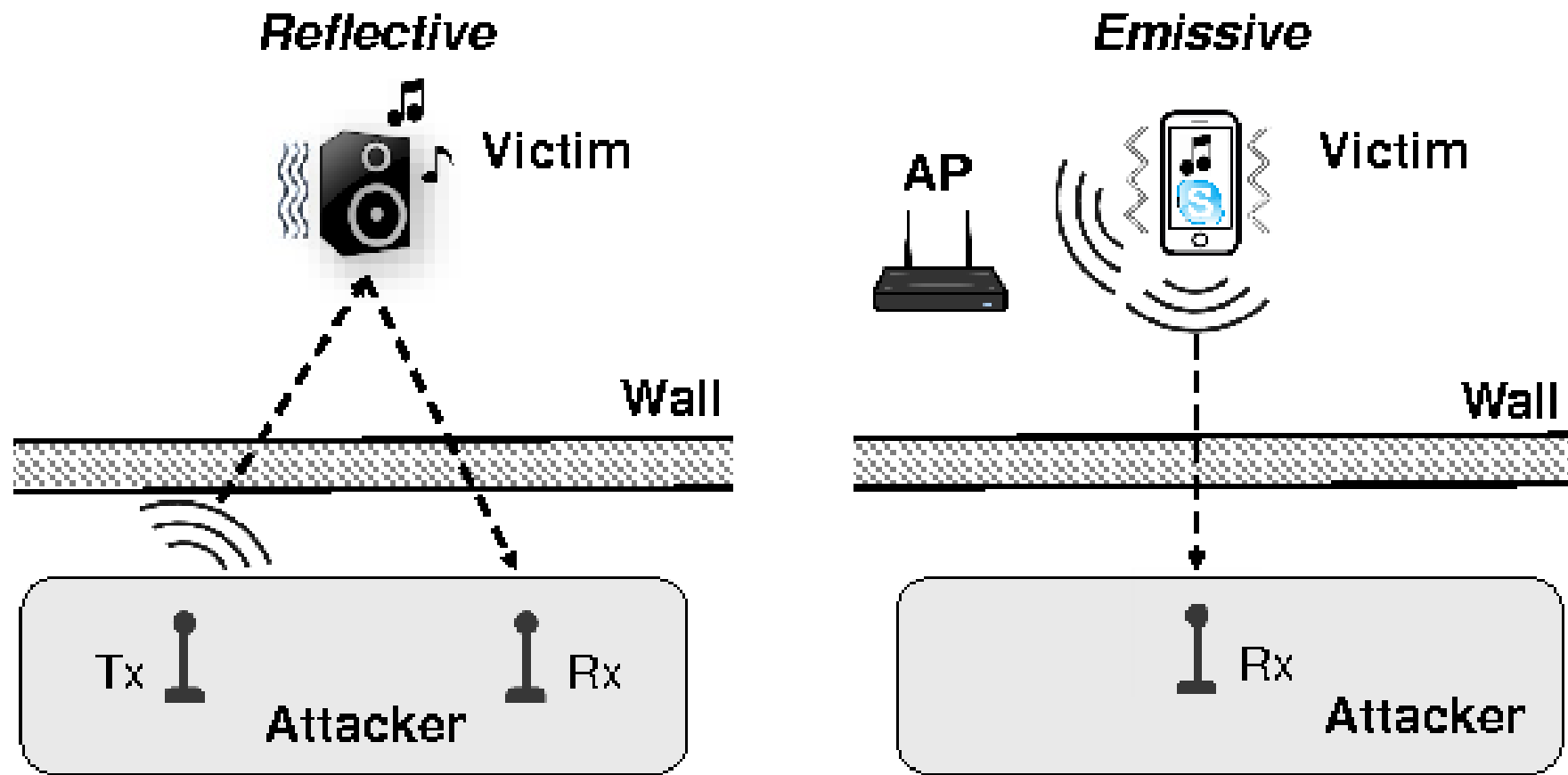
Acoustic Eavesdropping



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

Creatas

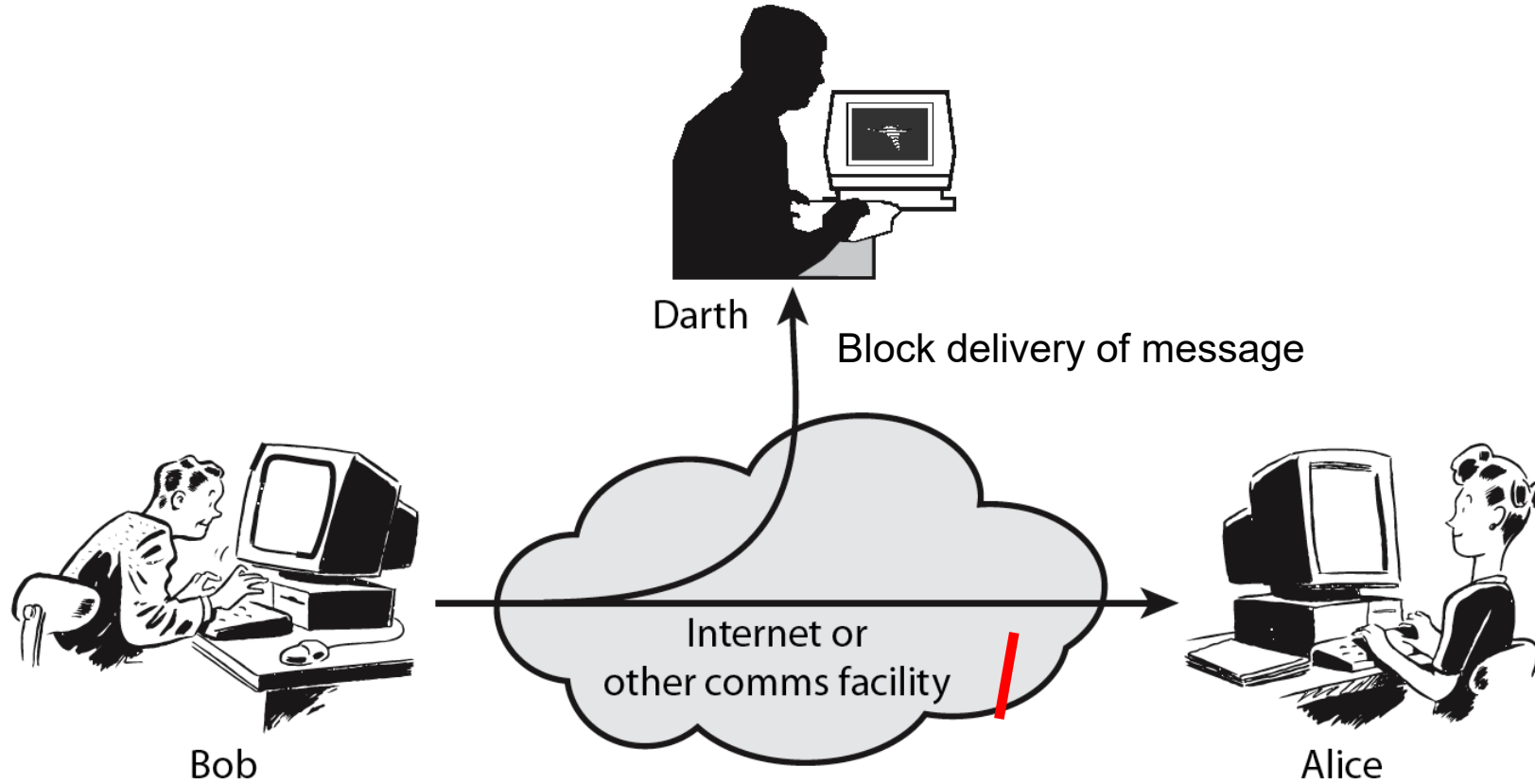
www.comstock.com



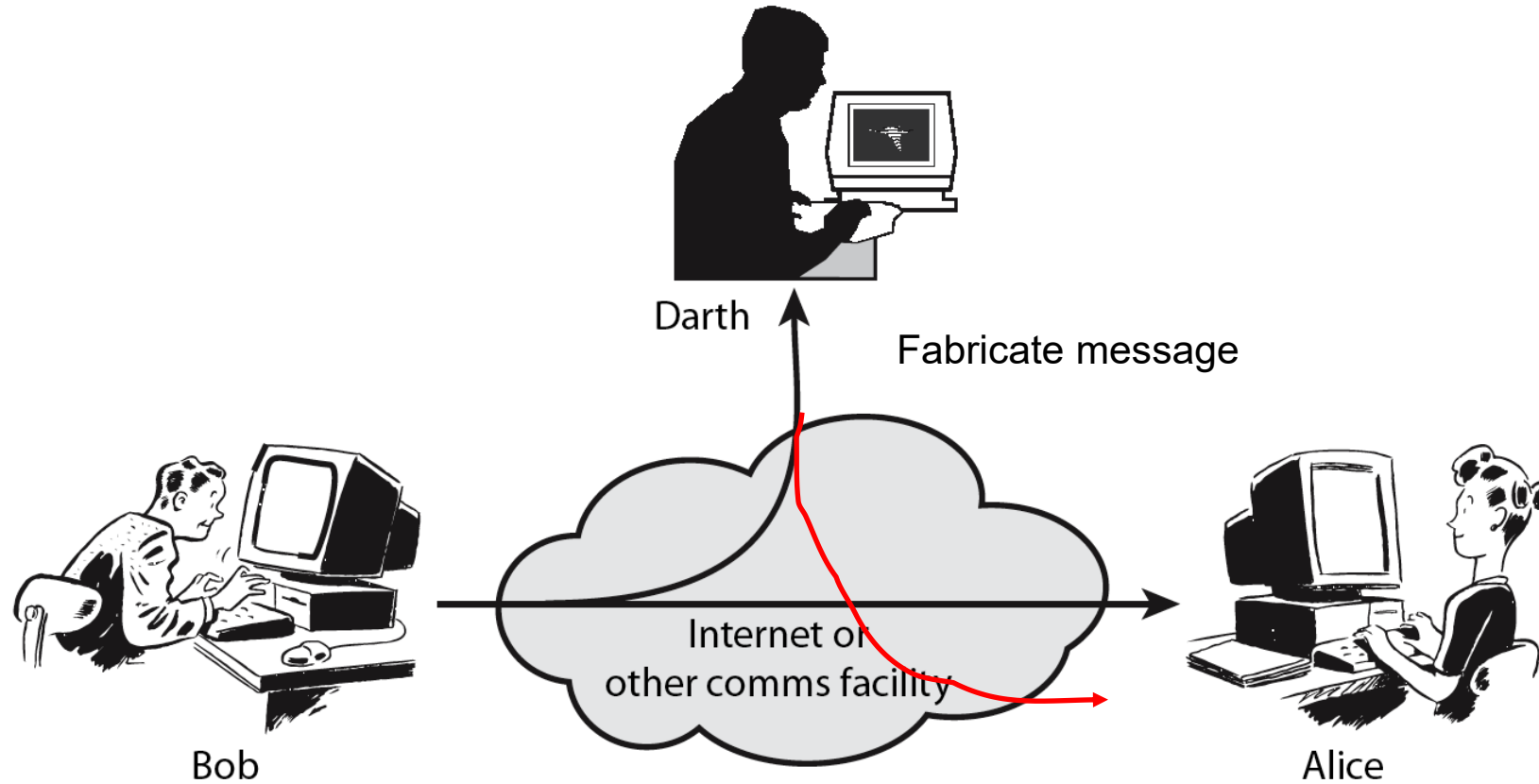
Sumber: <https://www.semanticscholar.org/paper/Acoustic-Eavesdropping-through-Wireless-Vibrometry-Wei-Wang/8afd80726c54ed7b95d30d1230bef633d128c930>

2. Serangan aktif (*active attack*)

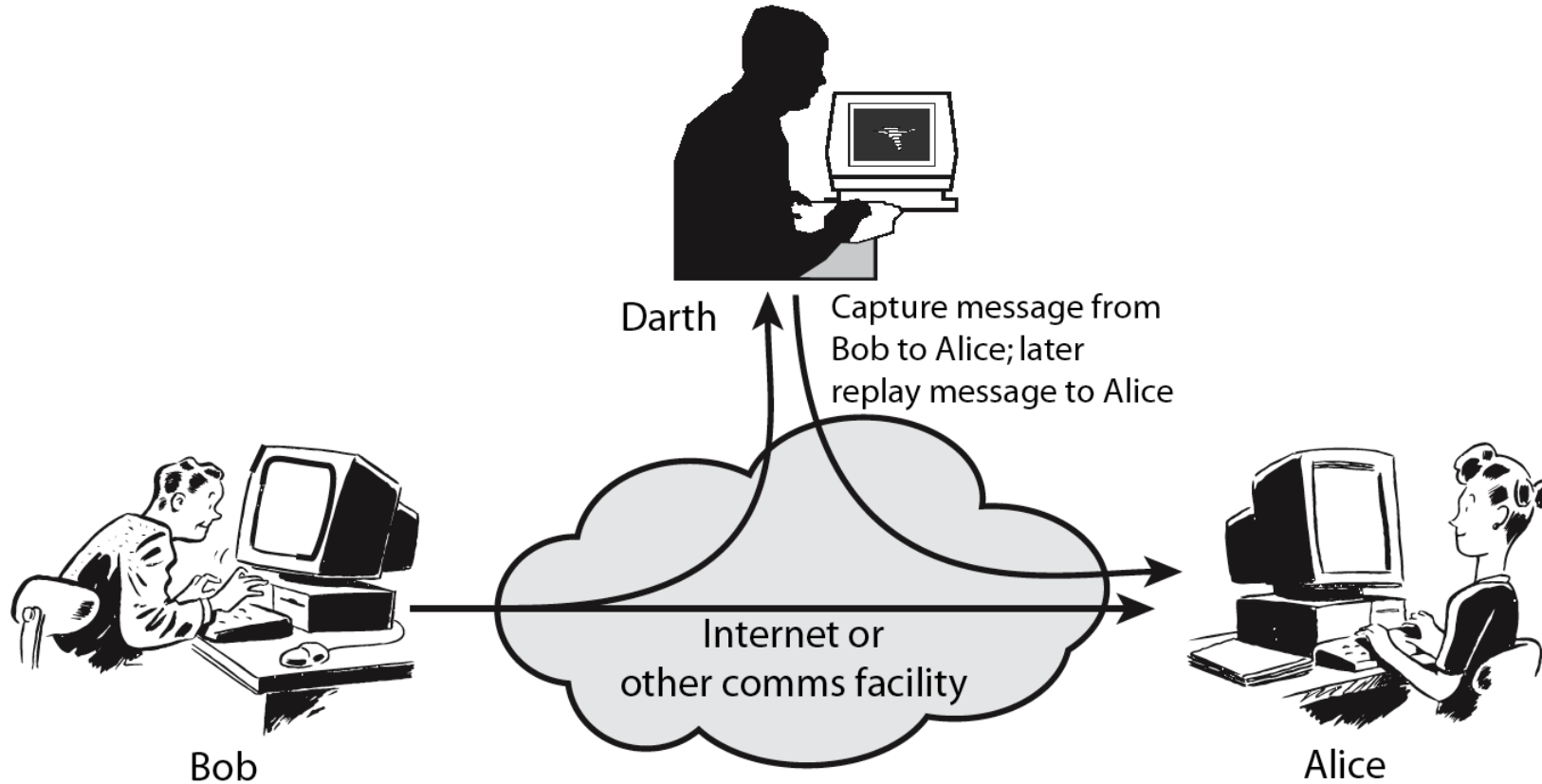
- penyerang mengintervensi komunikasi dan ikut mempengaruhi sistem untuk keuntungan dirinya
- penyerang dapat mengubah aliran pesan seperti:
 - menghapus sebagian cipherteks,
 - mengubah cipherteks,
 - menyisipkan potongan cipherteks palsu,
 - me-*replay* pesan lama,
 - mengubah informasi yang tersimpan, dsb
- Contoh: *man-in-the-middle attack*



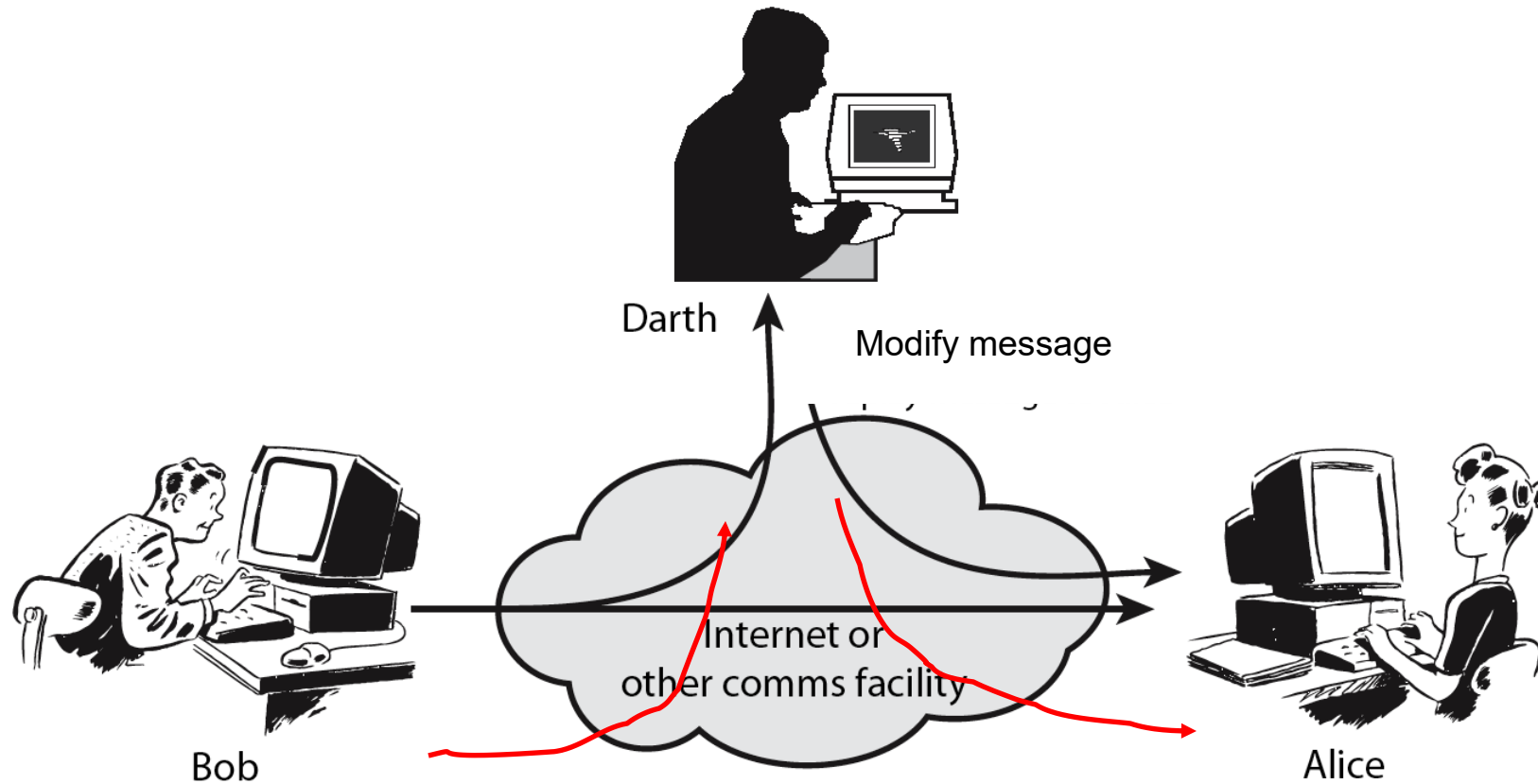
Active Attack: Interruption



Active Attack: Fabrication

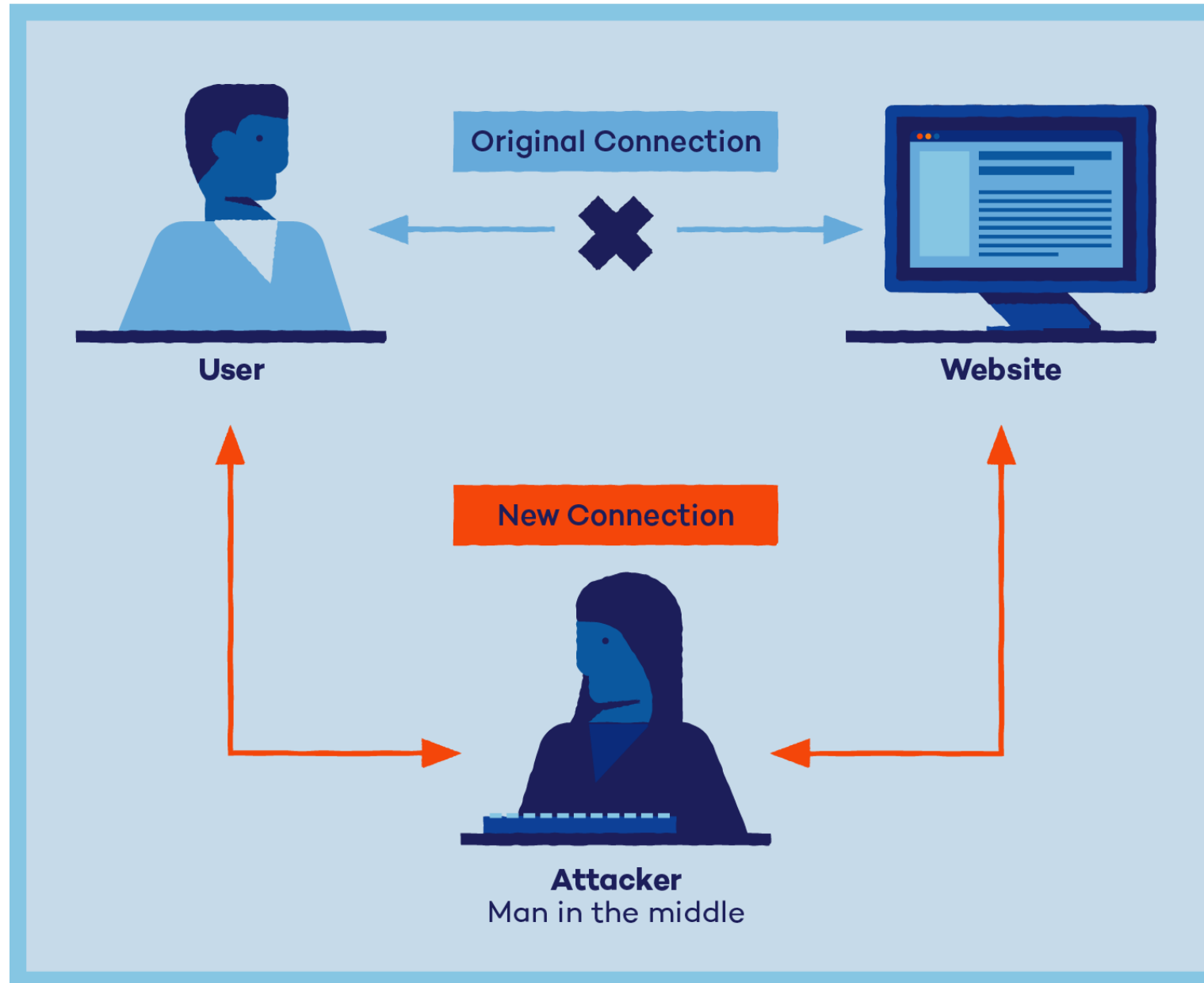


Active Attack: Replay

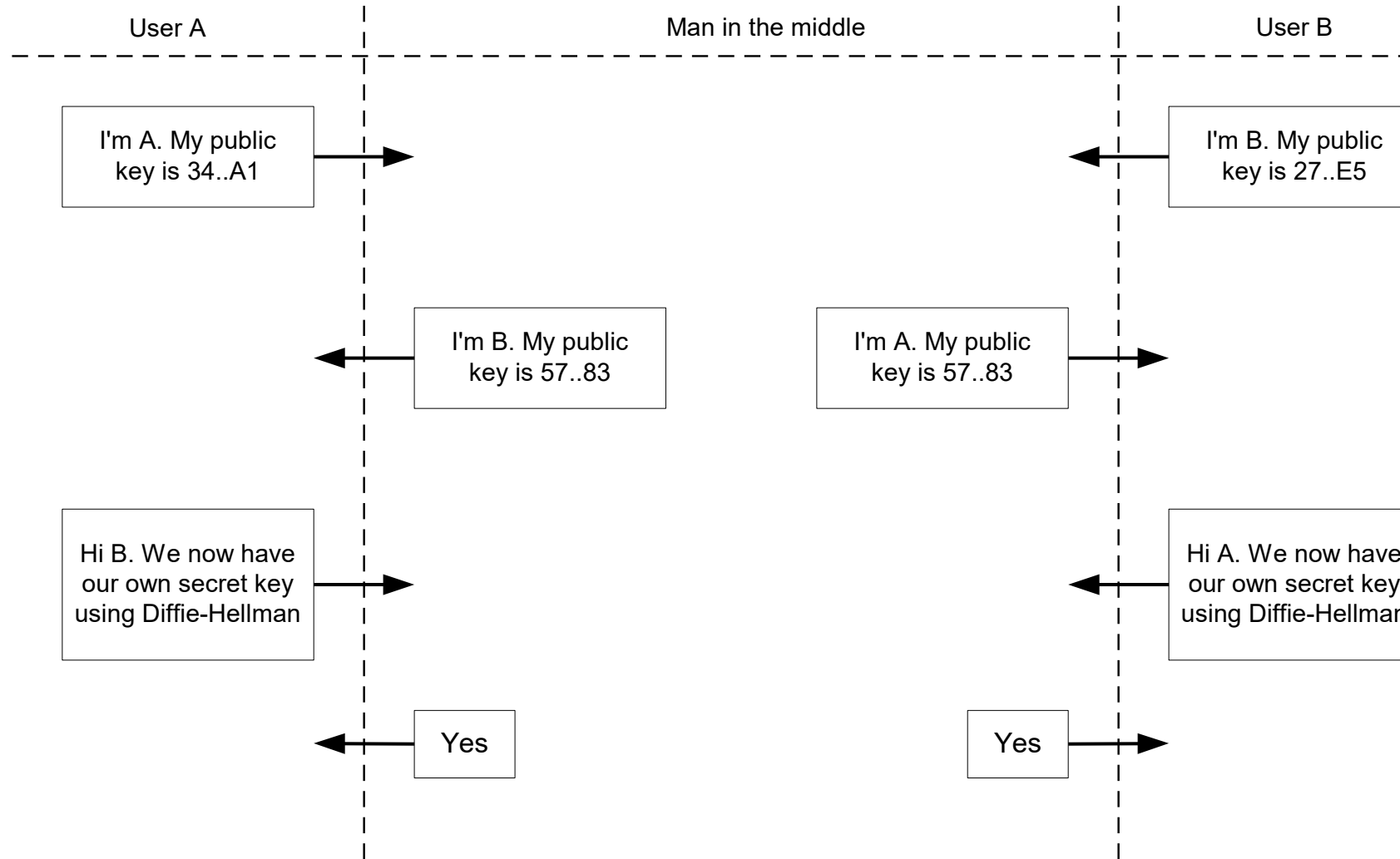


Active Attack: Modification

Man-in-the-middle-attack

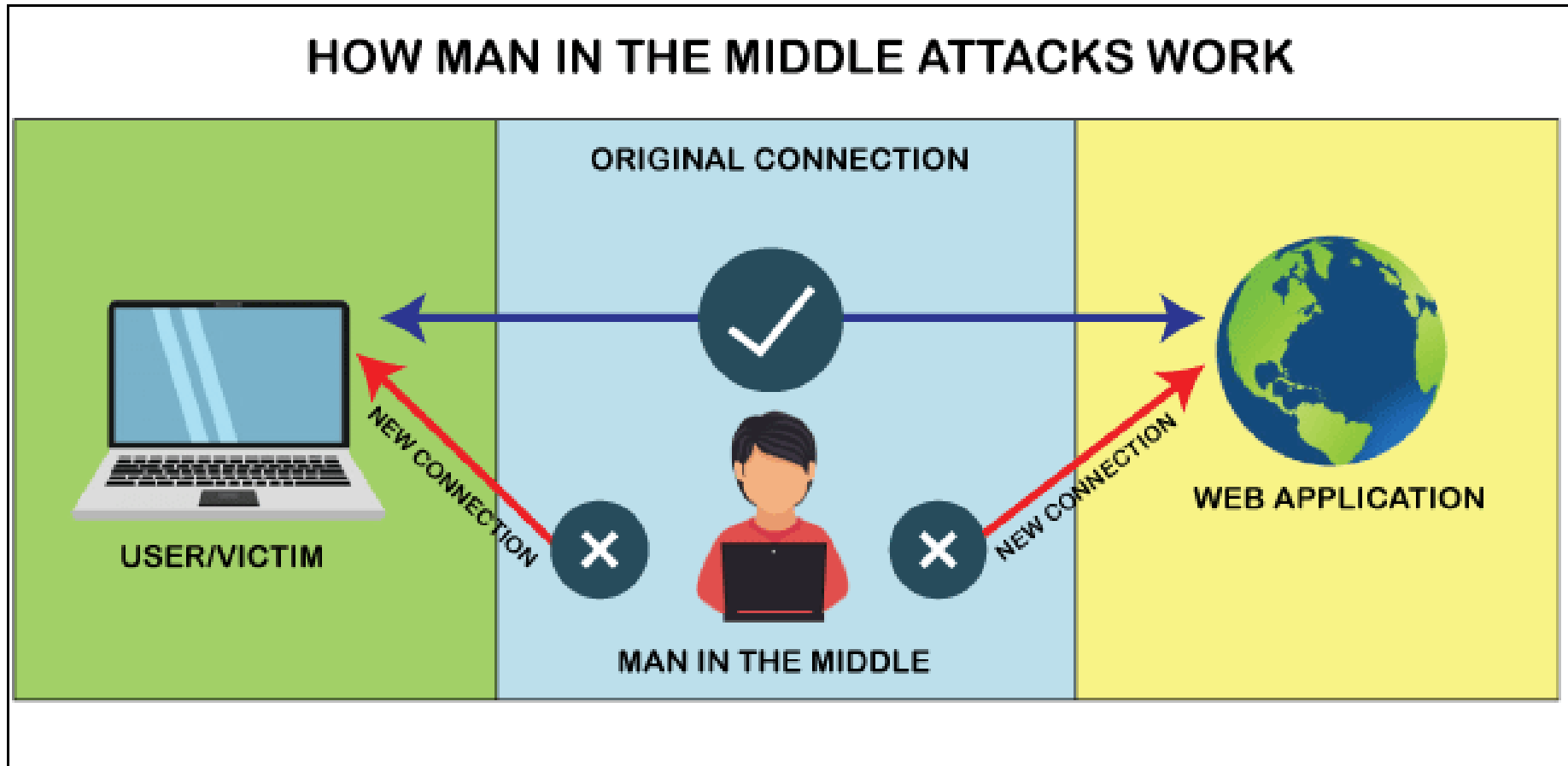


Man-in-the-middle-attack



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)*.