

Bahan kuliah IF4020 Kriptografi

06 - Serangan Terhadap Kriptografi



Oleh: Rinaldi Munir

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2024**

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!**

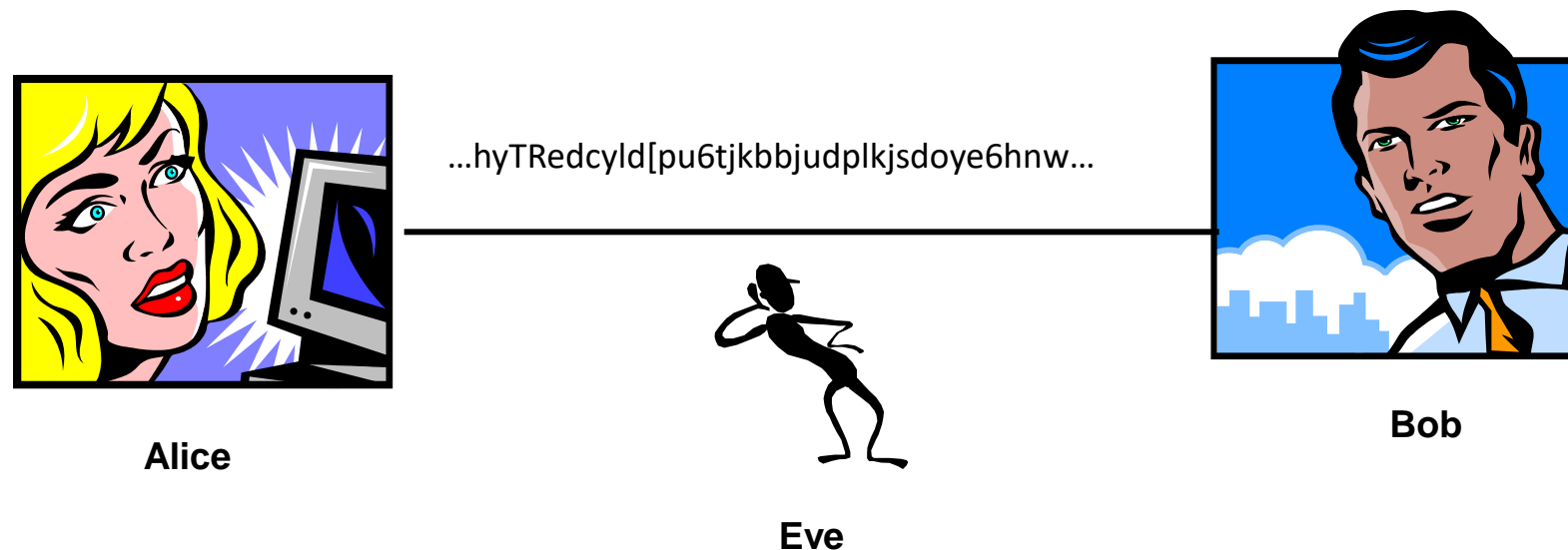
Jenis-jenis Serangan

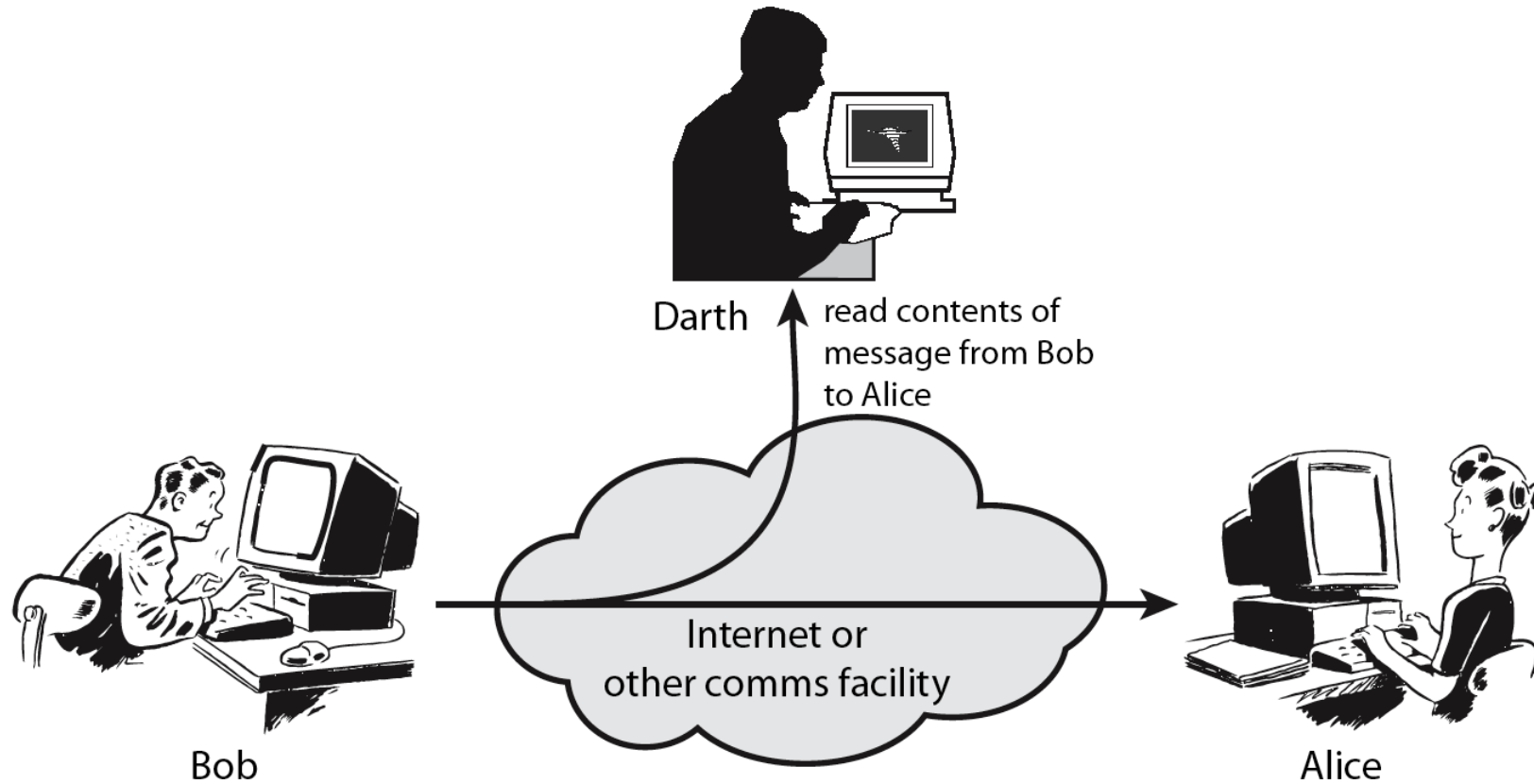
Berdasarkan keterlibatan penyerang dalam komunikasi pesan:

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

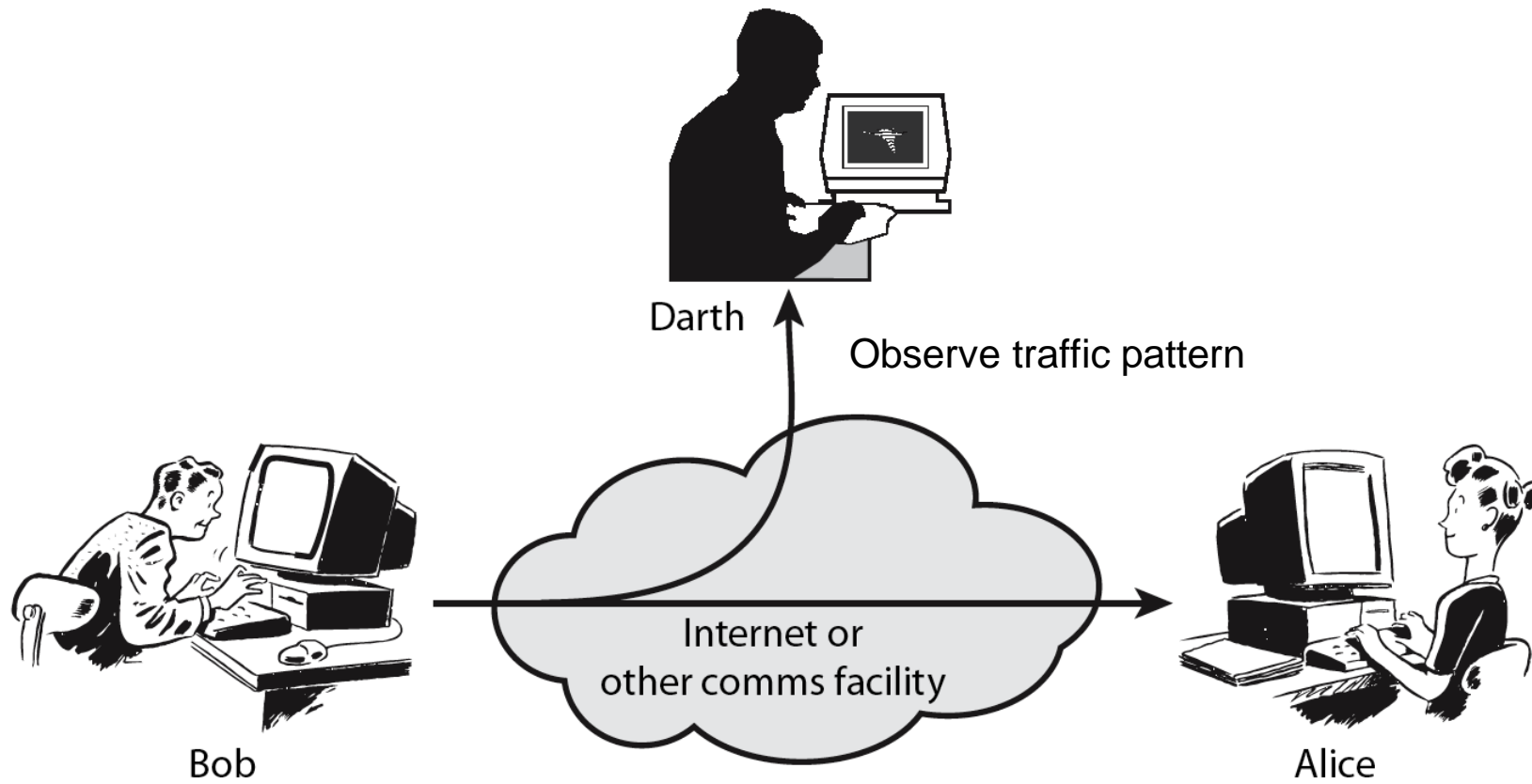
1. Serangan pasif (*passive attack*)

- 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





Passive Attack : Interception



Passive Attack : Traffic Analysis

Screenshot Wireshark (memantau network traffic)

The screenshot shows the Wireshark interface capturing network traffic 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 details pane for the selected packet (No. 1) shows the following structure:

- Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
- IEEE 802.3 Ethernet
- Logical-Link Control
- Internetnetwork Packet exchange
- NetBIOS over IPX

The packet bytes pane shows the following hex and ASCII data:

```
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 ff ff ....P...
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 57 52 45 5f 5f 02 ..... MSB POWER
```


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>

The screenshot shows a YouTube video player with the title "cara menggunakan wireshark untuk monitoring jaringan". The video content displays a Wireshark network capture interface. The main pane shows a list of network packets. The selected packet (No. 342) is an HTTP POST request to /wp-login.php. The packet details pane shows the request body containing the username "u53rn4m3" and password "p455word".

No.	Source	Destination	Protocol	Length	Info
315	192.168.43.40	142.250.4.188	TCP	55	24931 → 5228 [ACK] Seq=1 Ack=1 Win=254 Len=1
316	142.250.4.188	192.168.43.40	TCP	66	5228 → 24931 [ACK] Seq=1 Ack=2 Win=265 Len=0 SLE=1 SRE=2
317	192.168.43.40	104.237.53.82	TCP	66	25387 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
318	192.168.43.40	104.237.53.82	TCP	66	25388 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
319	104.237.53.82	192.168.43.40	TCP	66	80 → 25387 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1 WS=128
320	192.168.43.40	104.237.53.82	TCP	54	25387 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
321	192.168.43.40	104.237.53.82	HTTP	1403	POST /wp-login.php HTTP/1.1 (application/x-www-form-urlencoded)
322	104.237.53.82	192.168.43.40	TCP	66	80 → 25388 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1 WS=128
323	192.168.43.40	104.237.53.82	TCP	54	25388 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
324	104.237.53.82	192.168.43.40	TCP	66	80 → 25389 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1400 SACK_PERM=1 WS=128
325	192.168.43.40	104.237.53.82	TCP	54	25389 → 80 [ACK] Seq=1 Ack=1 Win=65792 Len=0
326	192.168.43.40	104.237.53.82	HTTP	1403	TCP Retransmission] 25387 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65792 Len=1349
327	104.237.53.82	192.168.43.40	TCP	54	80 → 25387 [ACK] Seq=1 Ack=1350 Win=32128 Len=0
328	104.237.53.82	192.168.43.40	TCP	1456	[TCP Previous segment not captured] 80 → 25387 [ACK] Seq=1401 Ack=1350 Win=32128 Len=1400 [TCP segment of a ...
329	104.237.53.82	192.168.43.40	HTTP	1456	[TCP Previous segment not captured] Continuation
330	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=2001 Ack=1350 Win=32128 Len=1400
331	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=5001 Ack=1350 Win=32128 Len=1400
332	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=1 Ack=1350 Win=32128 Len=1400
333	104.237.53.82	192.168.43.40	TCP	1456	[TCP Previous segment not captured] 80 → 25387 [ACK] Seq=12001 Ack=1350 Win=32128 Len=1400 [TCP segment of a ...
334	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=4201 Ack=1350 Win=32128 Len=1400
335	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=8401 Ack=1350 Win=32128 Len=1400
336	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=9001 Ack=1350 Win=32128 Len=1400
337	104.237.53.82	192.168.43.40	TCP	1456	[TCP Out-Of-Order] 80 → 25387 [ACK] Seq=11201 Ack=1350 Win=32128 Len=1400
338	192.168.43.40	104.237.53.82	TCP	66	[TCP Dup ACK 32081] 25387 → 80 [ACK] Seq=1350 Ack=1 Win=65792 Len=0 SLE=1401 SRE=2001
339	192.168.43.40	104.237.53.82	TCP	74	[TCP Dup ACK 32082] 25387 → 80 [ACK] Seq=1350 Ack=1 Win=65792 Len=0 SLE=7001 SRE=8401 SLE=1401 SRE=2001
340	192.168.43.40	104.237.53.82	TCP	74	[TCP Dup ACK 32083] 25387 → 80 [ACK] Seq=1350 Ack=1 Win=65792 Len=0 SLE=1401 SRE=4201 SLE=7001 SRE=8401
341	192.168.43.40	104.237.53.82	TCP	74	[TCP Dup ACK 32084] 25387 → 80 [ACK] Seq=1350 Ack=1 Win=65792 Len=0 SLE=5601 SRE=8401 SLE=1401 SRE=4201
342	192.168.43.40	104.237.53.82	TCP	66	25387 → 80 [ACK] Seq=1350 Ack=4201 Win=65792 Len=0 SLE=5601 SRE=8401

Frame 1: 74 bytes on wire (592 bits) ... 74 bytes captured (592 bits) on interface ... id ...

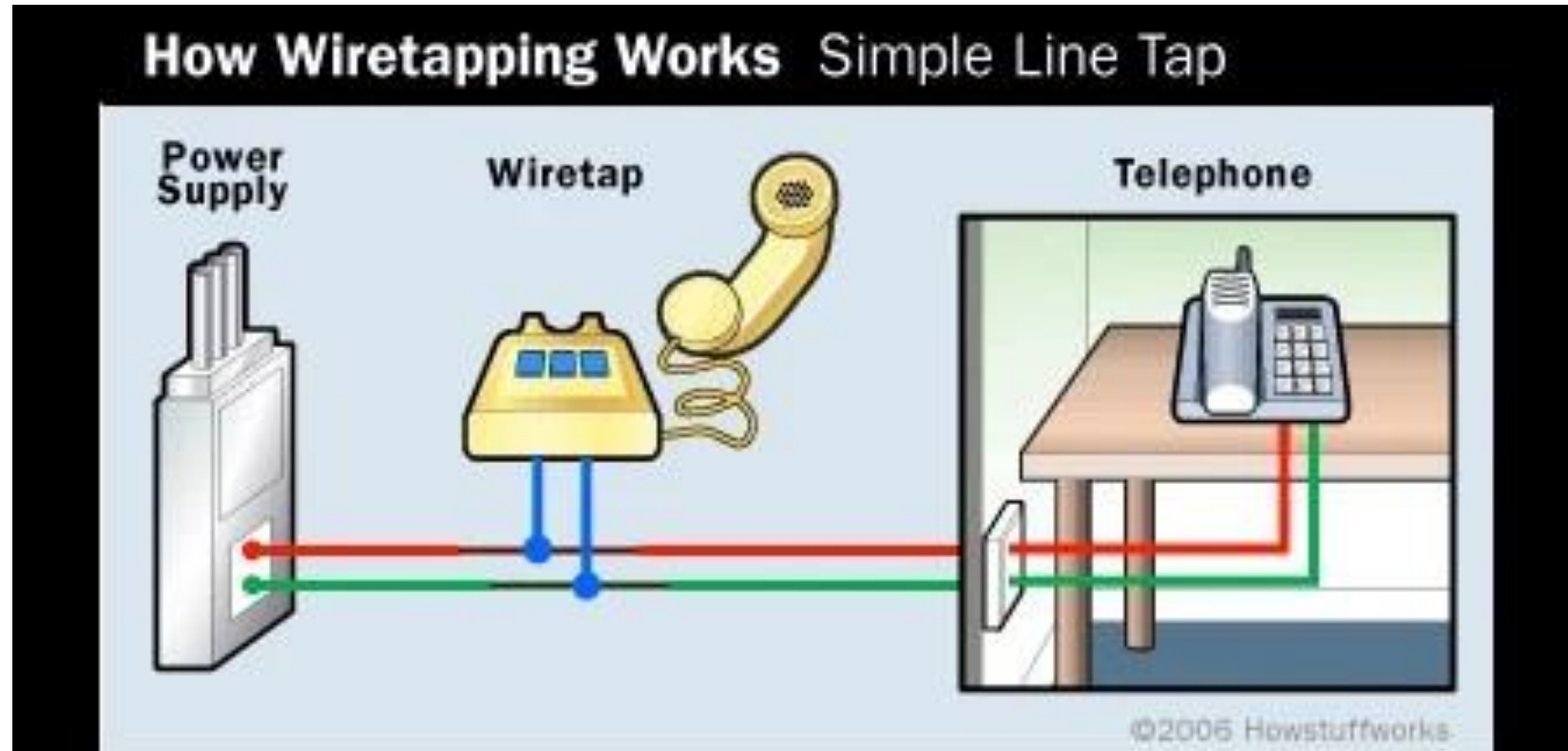
```
0000 b8 57 08 27 fe 02 dc 8b 28 0a 77 83 00 00 00 00  M.....(w...
0010 00 3c 0c 5f 00 00 80 11 56 d8 c0 a8 2b 28 c0 a8  <.....V....{...
0020 2b 01 c3 24 00 35 00 28 c6 3f 0f e3 01 00 00 01  *..$ (-?...
0030 00 00 00 00 00 03 77 77 77 06 67 6f 6f 6f 6c  .....ww.googl
0040 65 03 63 6f 6d 00 00 01 00 01  e.com.....
```

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

Metode penyadapan:

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

- *Wiretapping*

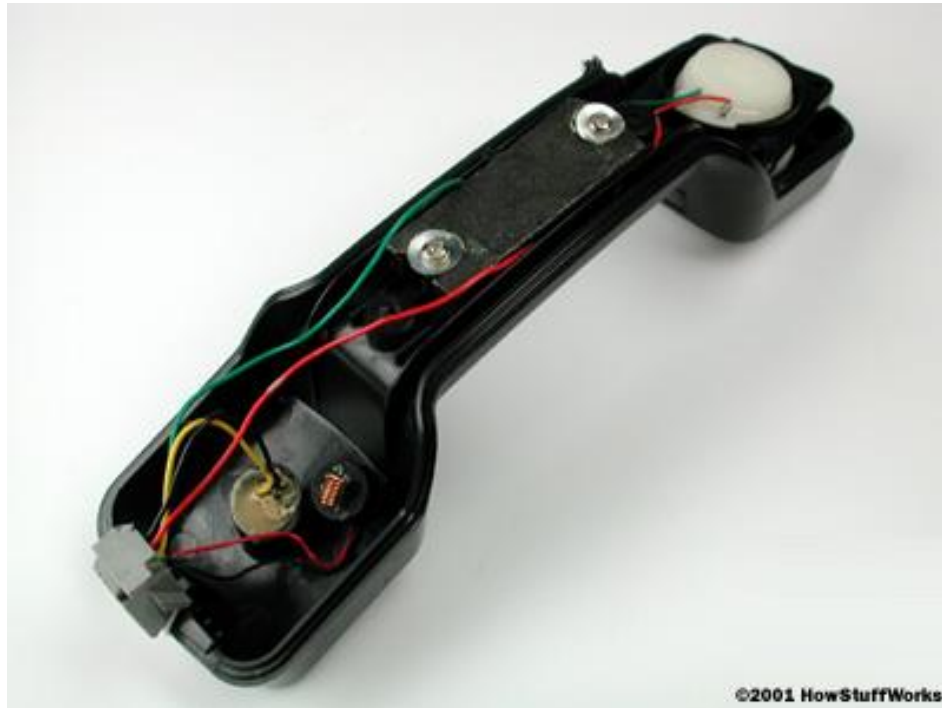


Baca di sini:

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>)

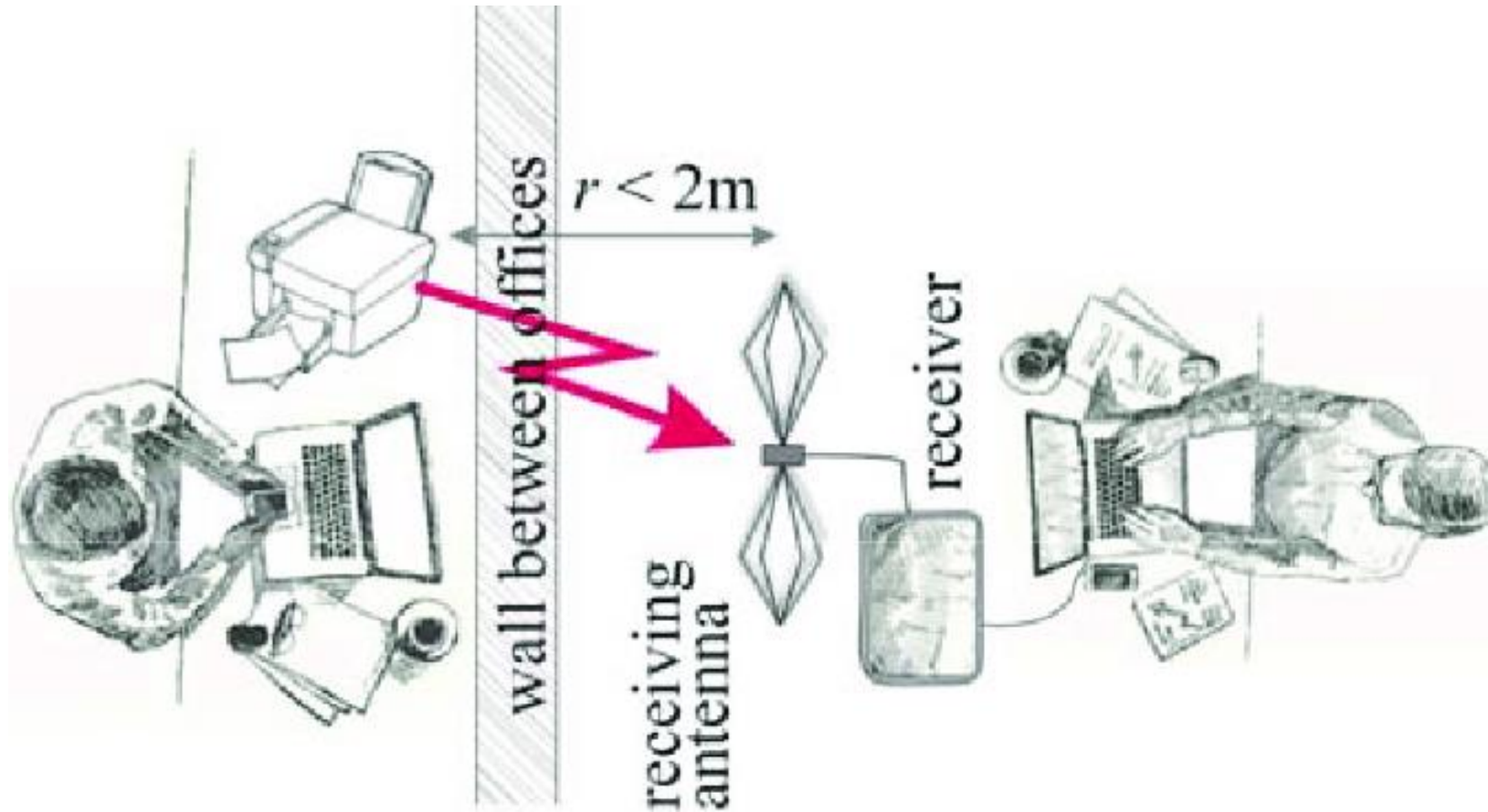


When you open up a phone, you can see that the technology inside is very simple. The simplicity of design makes the phone system vulnerable to surreptitious eavesdropping.



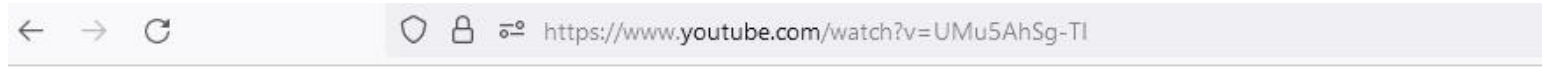
Inside a standard phone cord, you'll find a red wire and a green wire. These wires form a circuit like the one you might find in a flashlight. Just as in a flashlight, there is a negatively-charged end and a positively-charged end to the circuit. In a telephone cord, the green wire connects to the positive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



Telusuri



INTERCEPT ANY RADIO SIGNAL!!!!



andy kirby
105 rb subscriber

Subscribe

7,4 rb



Bagikan



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

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

The screenshot displays a YouTube video player with the URL <https://www.youtube.com/watch?v=krJJKjYdwgc>. The video content shows a software-defined radio (SDR) interface. The interface includes a terminal window with the following output:

```
root@kali: ~  
root@kali:~# grgsmLivemon  
Linux: GNU C++ version 5.4.0 20160609; Boost_105800; UHD_003.009.004-0-unknown  
gr-osmosdr 0.1.4 (0.1.4) gnuradio 3.7.10  
built-in source types: file osmosdr fcd rtl rtl_tcp uhd miri hackrf bladerf rfspace  
ace airspy redpitaya  
Using device #0 Realtek RTL2838UHIDIR SN: 00000001  
Detached kernel driver  
Found Ettus E4000 tuner  
Exact sample rate is: 2000000.052982 Hz
```

The interface also shows a network traffic capture window with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
1	0...	127.0.0.1	127.0.0.1	TCP	74	40592 → 5444 [SYN] Seq=0 win=0
2	0...	127.0.0.1	127.0.0.1	TCP	54	5444 → 40592 [RST, ACK] Seq=...
3	0...	127.0.0.1	127.0.0.1	TCP	74	40594 → 5444 [SYN] Seq=0 win=0

The interface also shows a spectrum analyzer window with the following data:

Frequency (MHz)	Relative Gain (dB)
935.454	3.96

The video player interface includes a browser address bar, YouTube logo, and search bar. The video title is "GSM Sniffing: Voice Decryption 101 - Software Defined Radio Series #11". The channel name is "Crazy Danish Hacker" with 26,9 rb subscribers. The video has 2 rb likes and 1 comment. The video player interface includes a browser address bar, YouTube logo, and search bar.

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



Crazy Danish Hacker
26,9 rb subscriber

Subscribe

2 rb



Bagikan

Download



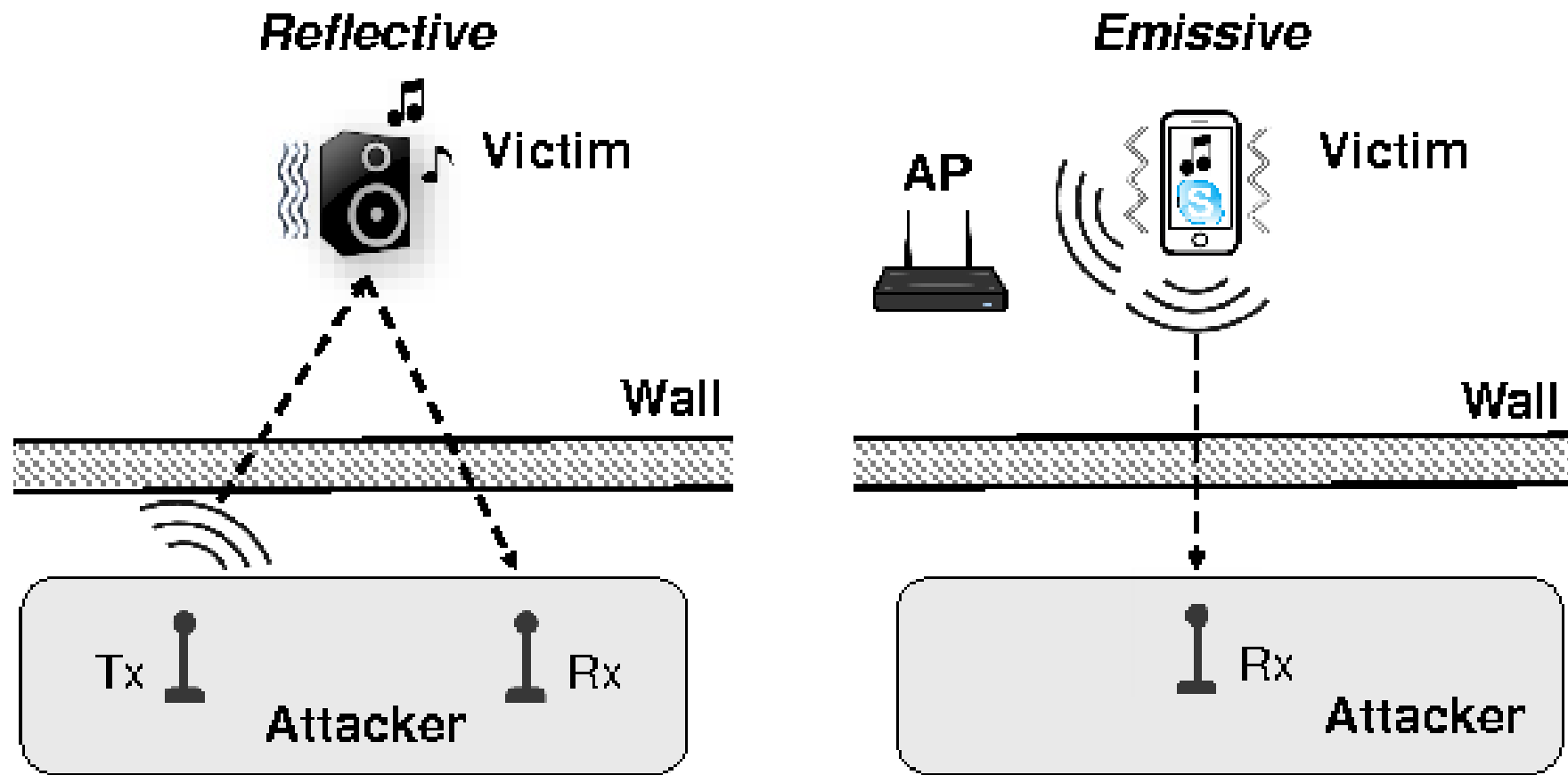
Acoustic Eavesdropping



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

Creatas

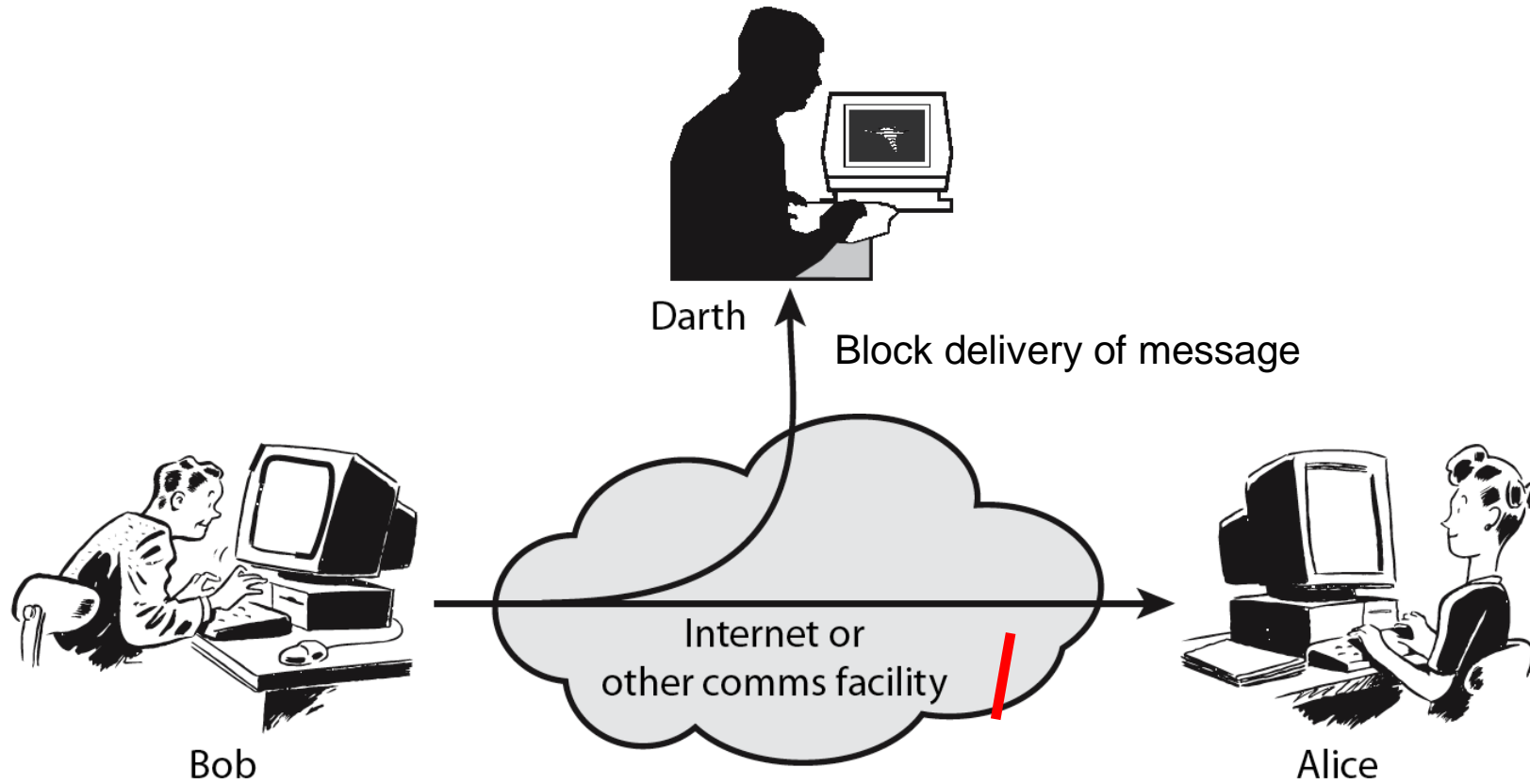
www.comstock.com



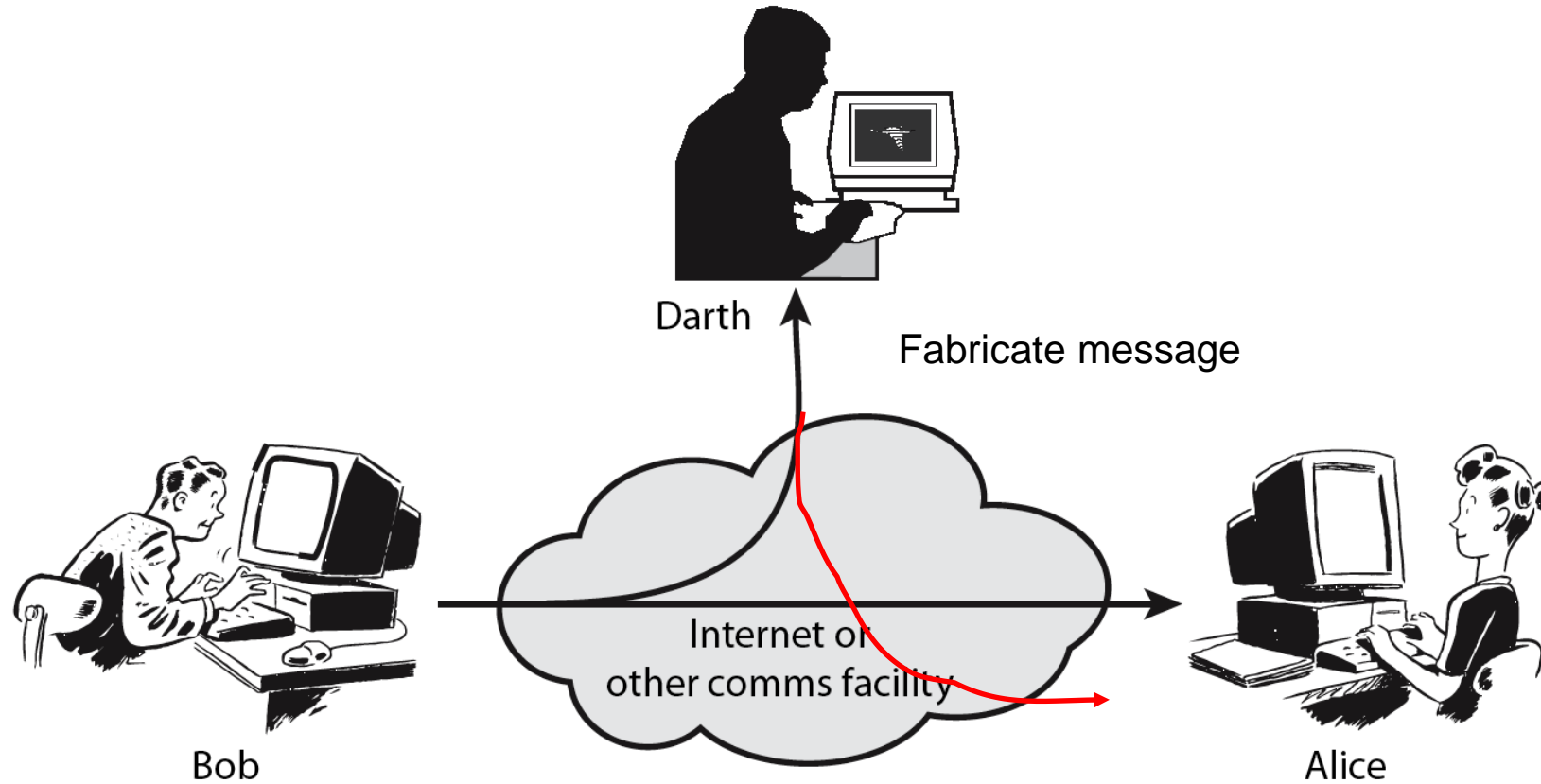
Sumber: <https://www.semanticscholar.org/paper/Acoustic-Eavesdropping-through-Wireless-Vibrometry-Wei-Wang/8afd80726c54ed7b95d30d1230bef633d128c930>

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*

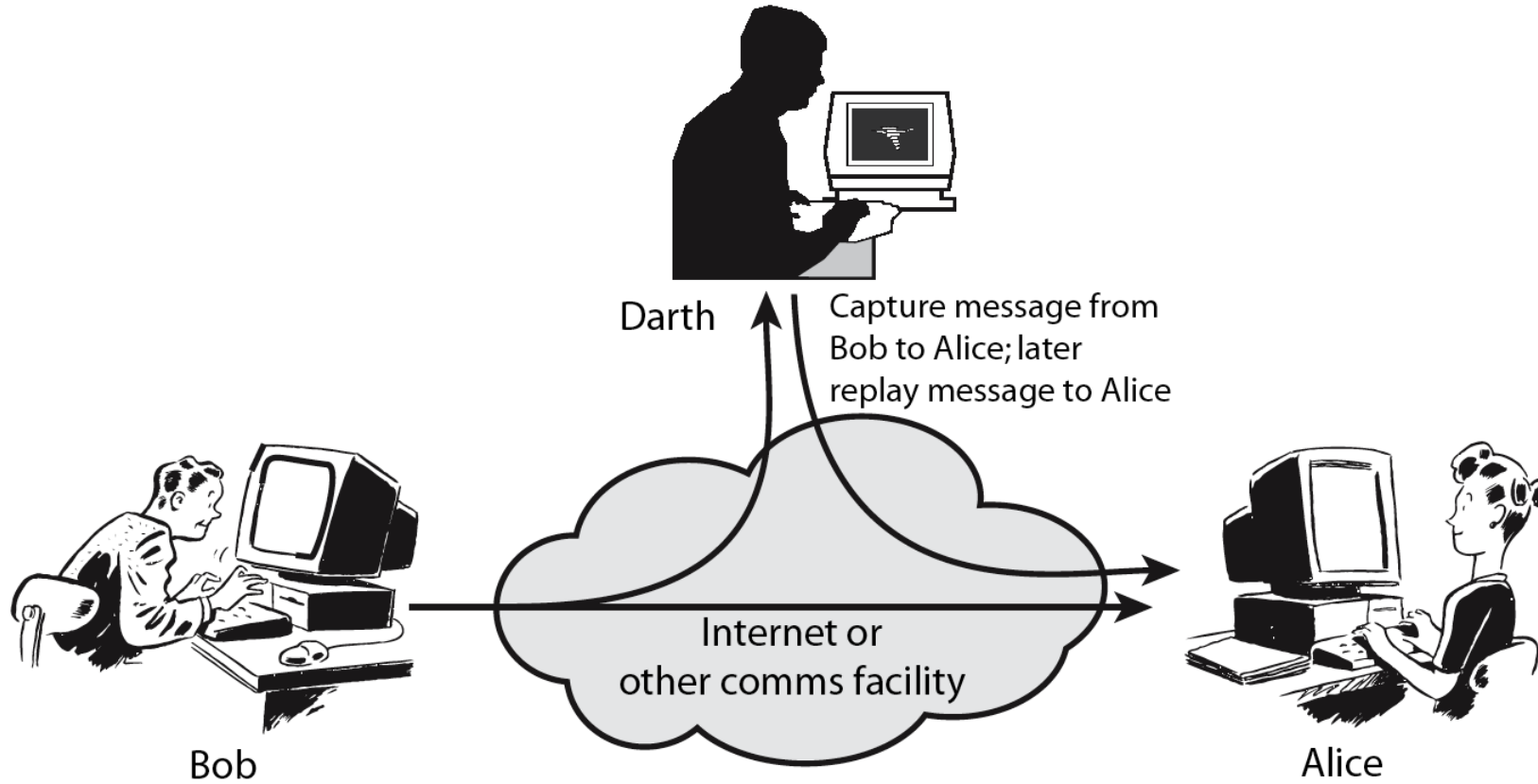


Active Attack: Interruption

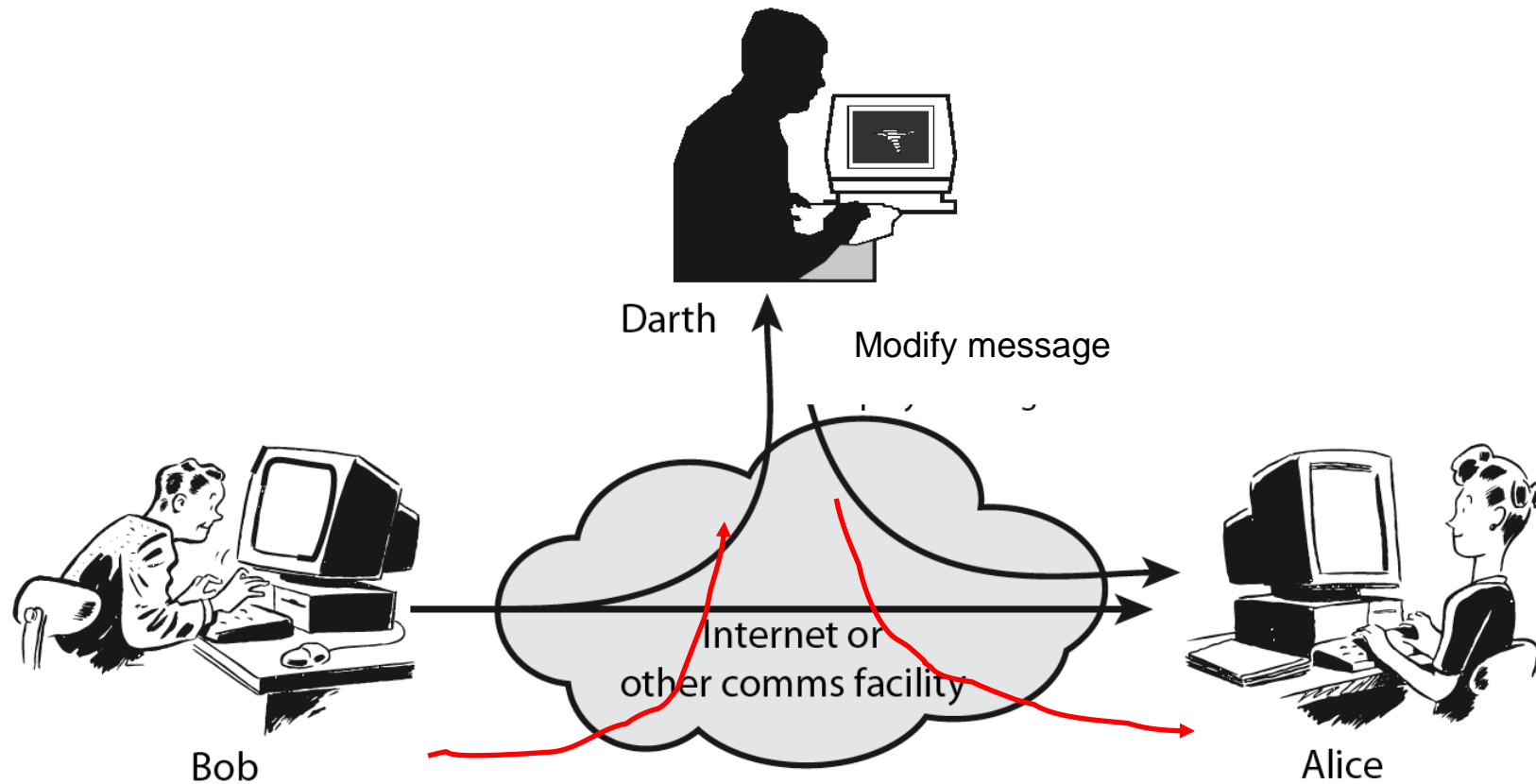


Active Attack: Fabrication

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

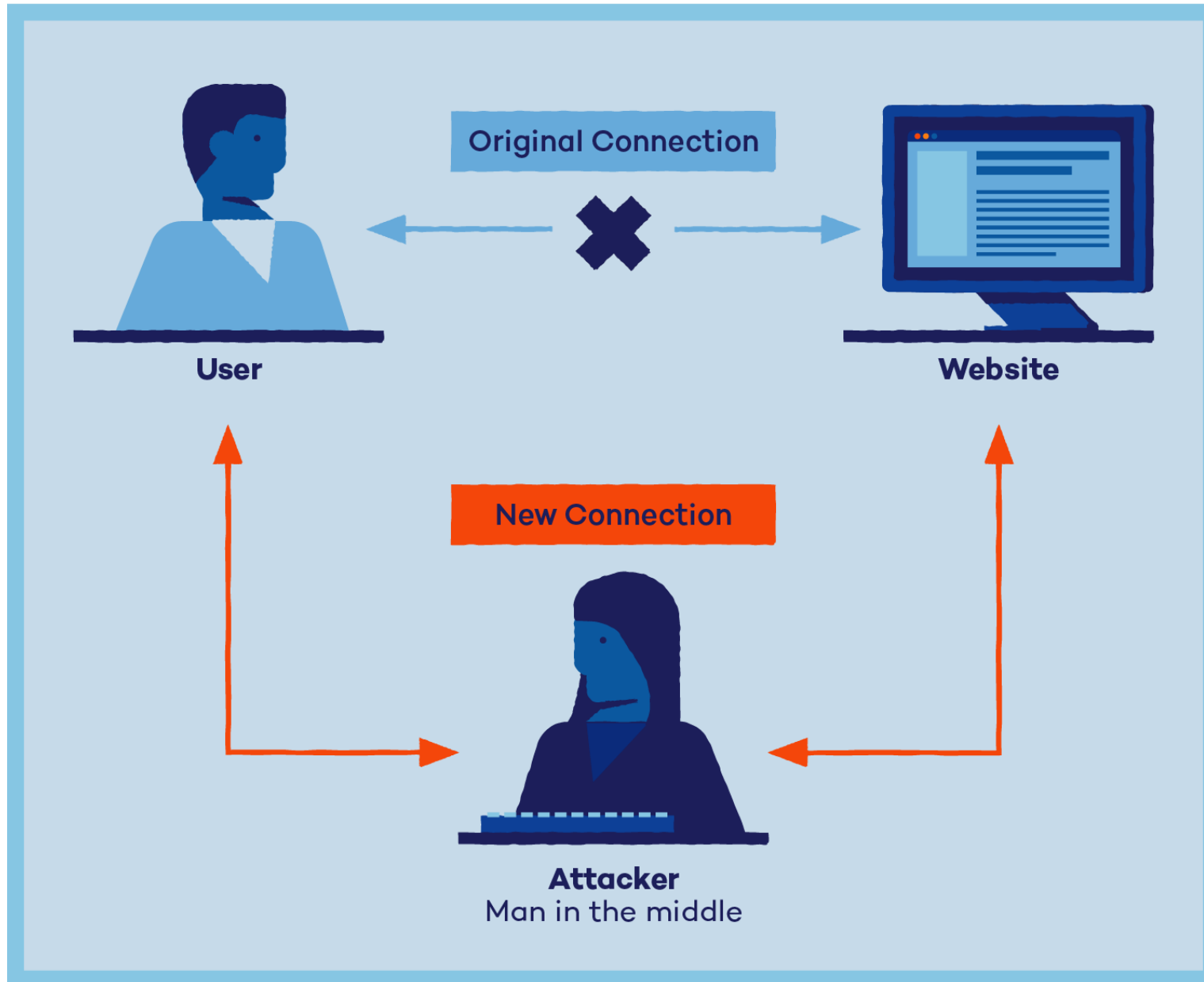


Active Attack: Replay

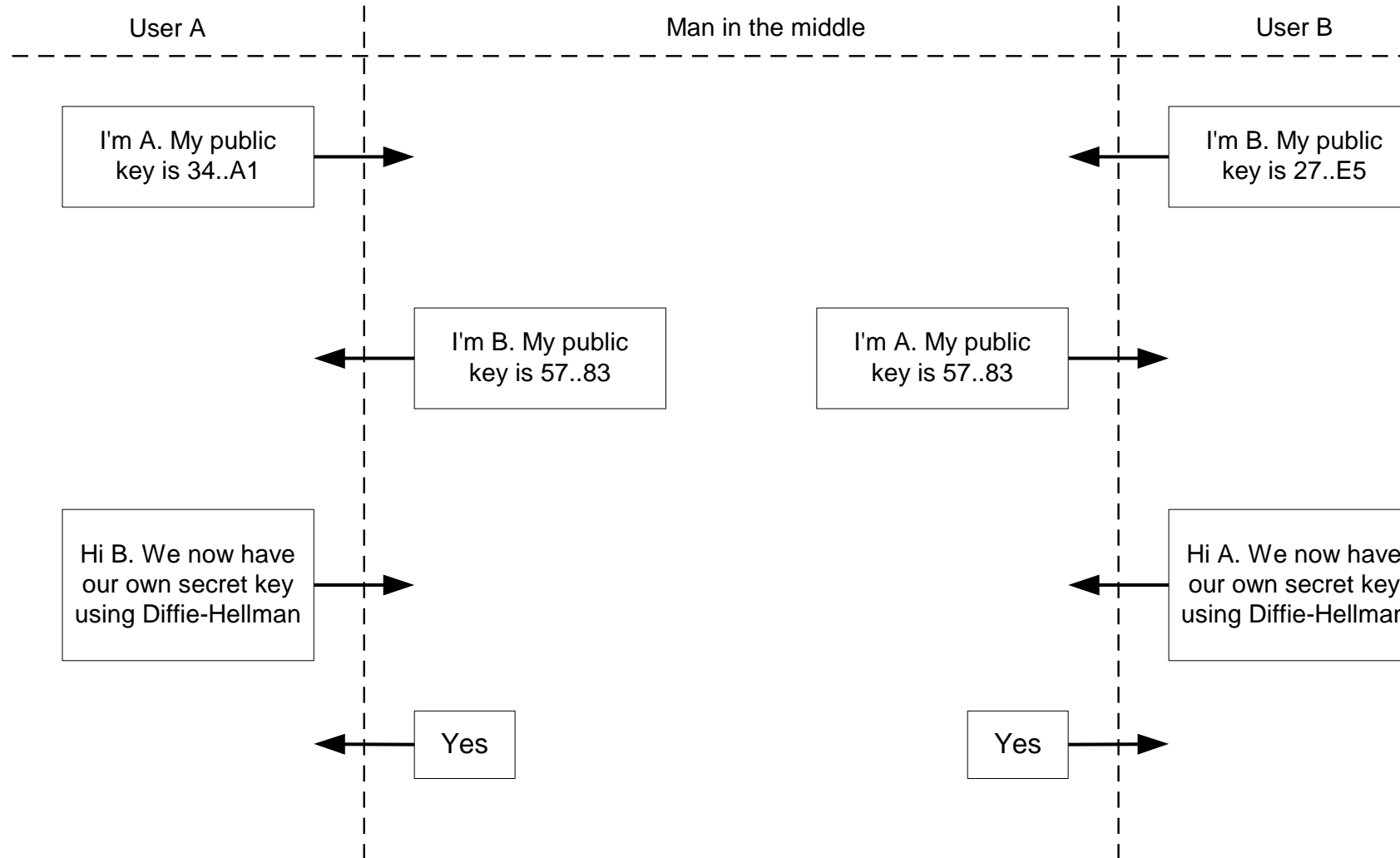


Active Attack: Modification

Man-in-the-middle-attack

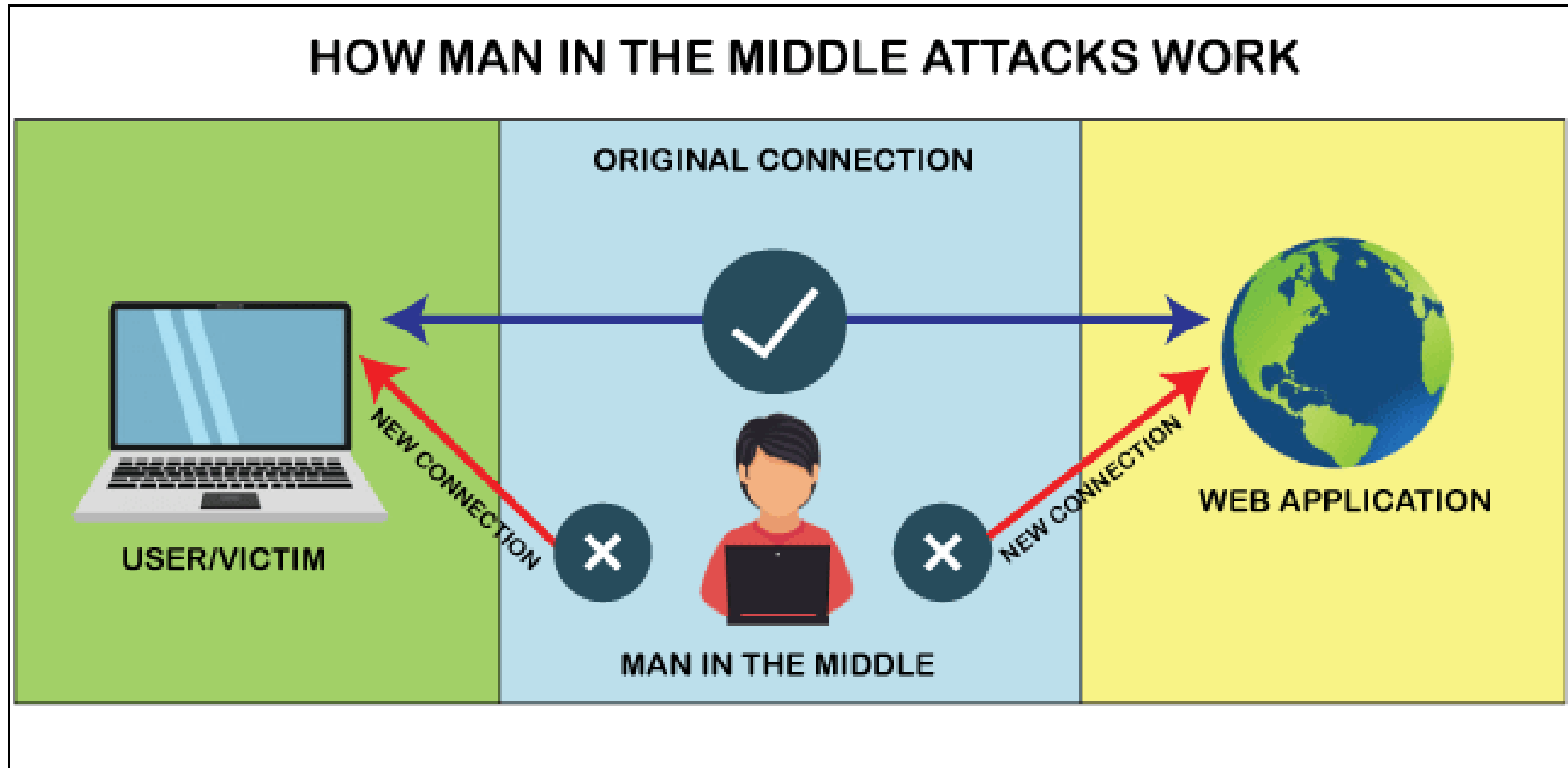


Man-in-the-middle-attack



Man-in-the-middle-attack

Serangan aktif yang berbahaya



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* 
$$\begin{aligned} \text{Enkripsi: } C &\equiv mP + b \pmod{n} \\ \text{Dekripsi: } P &\equiv m^{-1}(C - b) \pmod{n} \\ \text{Kunci: } m &\text{ dan } b \end{aligned}$$

m bilangan bulat yang relatif prima dengan n

b adalah jumlah pergeseran

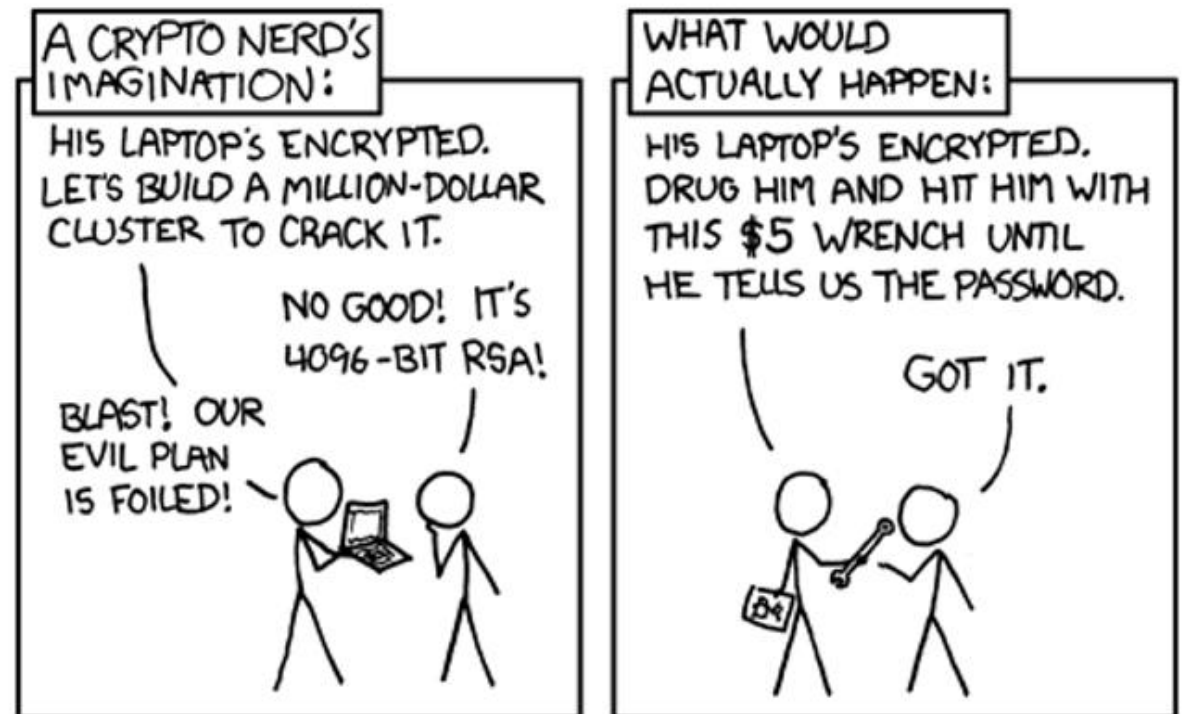
m dan b dapat ditemukan dengan memecahkan dua buah persamaan linier yang memuat peubah m dan b , asalkan diketahui dua pasang plainteks dan cipherteks yang berkoresponden.

- 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. *Ciphertext-only attack*

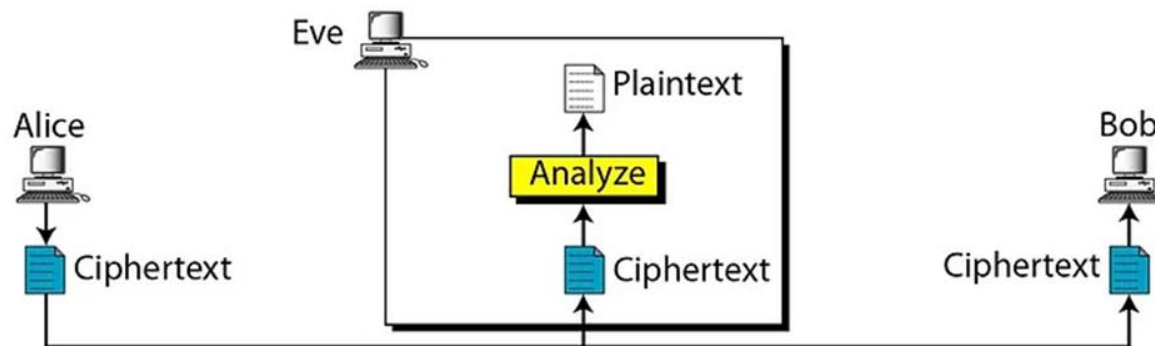
Kriptanalis hanya memiliki cipherteks saja.

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
4DQUYbINRmJVu1JJC/acGvgOze66pHuRgSCxxHDscefjXenh/XejSYTo7aMi+Es7
DCcD49zH6ZLDQN6B1N9q2oFI8QIhQ2y1QJbat1dWi/4yYWlKZcLKRSm8eo/gNCdL
h9MncXBBSfgbvbu67CDZ9G05geZOn3LzQOpJ8hrZq/6K/uMcUKeZjW3RCo0T754f
E5zYe1wUgtwS/lmQ2w5PQF/89bpshtDSYuL1fZgzrsE6DwophuCri5zwCGbEKlsI
g6REIETFbZ2aCL4N2pZVunCIEuoP0zgEB6+M9egdpyxMsMqEBVg3AH7SalAtEguP
T/MCxi0bZHCUhPupEKT8slbSrDNxTWMUXQt3XpL0bGCCrDMKLSOWYfDiNnRkFbWK
iiqw9hx4Q9CJg7xX7JRnVgwOeREiFnMYSbFlvPSxEou6FdBYhdqSefKin4Wnkmdw
qrS18fjIW/kZ2v72uz0buEKkY9ubBox76yjlRo9KUQMs3em03kc64959gTDiZ0qF
AgwDrosDPQ2BeYQBD/9H5VKFw0an5j5MX1JpOSBAqNGKWq2bcEFnwJfk0DDlhyHD
owHiG7gDowCS+5y/pf56v36HkzpJZATKqoRyKVxmQOxU9l3YnPc5fw8iFhxlrfcG
ywzkJh/BRDQ/uy5fhGc/PbSm6iLv/SkkWTK8PSUD+g1yZyK0W7WkMh9QYS2OE7lQ
qbwPNiy57reWkUWCoE4QmKqqpe7NXXM0eLT9l2D0hG2lthyvTvspkpxszl8+HMJv
M2LMcY2FmmZWAJSdxsQSq9NQdyvCJX2D8oa89WQyXmp7mPXL7BQfoQNPndmn6Obi
0EQojoemRNh14XNhMjPjxW7m34rH2gtvdN3Dg8iFrtocoVJqXqU3N+9T2sNe/bS8

Cipherteks

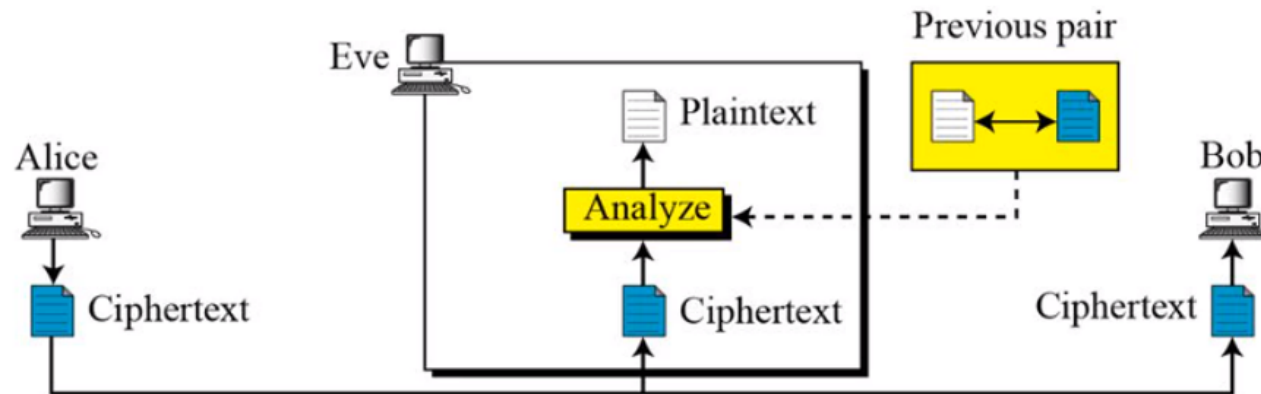
2. *Known-plaintext attack*

Diberikan sejumlah pasangan plainteks dan cipherteks yang berkoresponden:

$$P_1 \text{ dan } C_1 = E_k(P_1), \quad P_2 \text{ dan } C_2 = E_k(P_2), \quad \dots, \dots, \quad P_i \text{ dan } C_i = E_k(P_i)$$

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

Known-Plaintext Attack



- 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

TFJOXUPOUXYTRDSXQMONIYPEUFJDQUBGIMOCJQTNBEHCZEKROV
BNTWLMVXMOWZLUCHOXYGSKBQGUAOBQZKIXYJIETSWVXHVKCUAOT
OFYIZAKJGXKAWGQTRVFDZAJNQDUIWZCMYWNFIUPYMCZXIAKYUCQ
IAZPIQMGAMGUAKKKHMWKDUXQDUAAKYOWEHLJPWYFKXSARBLLHGA
JKTQNTTRTPWSCIZASCSLKVDTUZSWBNBTJGYYUPQMFSYZAUTOQC
DNGQMFSRLRTUWEMKADIVYLTJKFHLKJUWTSSHMHJFGTRIBYIDAHQ
EPMPIQCROWDYRYZNSPNOJHQVKKTOCBPNFAJNLYJZNVBAYJWRGMC
HJPWBDHHTPOXSIJVQWDMSIGMTRVEVXDILKVAYTNUNJXEZLAPGYE
TRVZNVHSVWLGICDXQFOALDVPASUSYXPFHUWTILUQHTJQVGWFS
PAEKBRBNIINYKHNTNUKJVDHVLXQKUZNVQXUOZZOJZYNPIVYSVFVTZ
MMUUPWTGHRIOWCBKZYAGUMRCKHIQZSIGISPGBXPYXMOAWGAGHQV
UWTEIGPBMOMBWIOPOQEVKMRQATNBMILHHLVUXGMOUWTZCLBKGWIJ
HFRNGOSCMUHDWHBB

wassalam

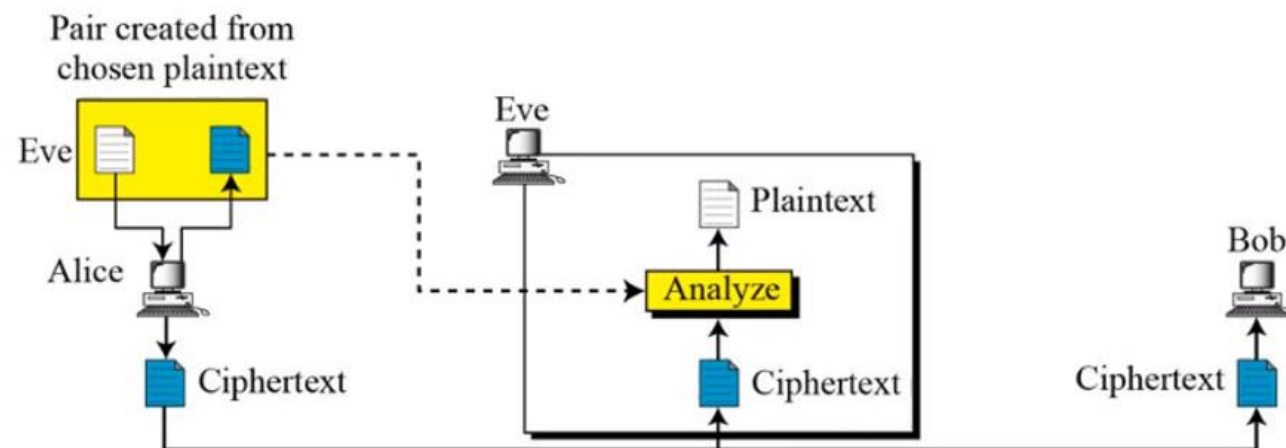
3. *Chosen-plaintext attack*

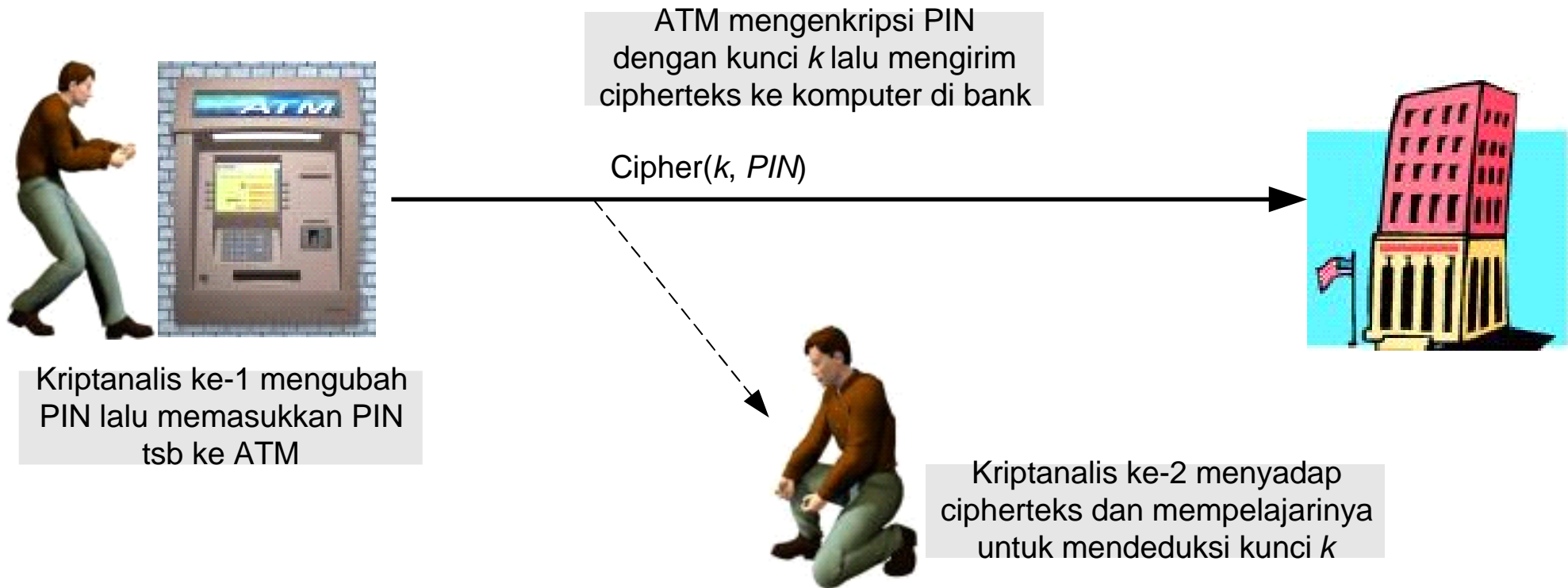
Kriptanalisis dapat memilih plainteks tertentu untuk dienkripsikan, yaitu plainteks-plainteks yang lebih mengarahkan penemuan kunci.

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





Chosen-plaintext attack

4. Adaptive-chosen-plaintext attack

Kriptanalis memilih blok plainteks yang besar, lalu dienkripsi, kemudian memilih blok lainnya yang lebih kecil berdasarkan hasil serangan sebelumnya, begitu seterusnya.

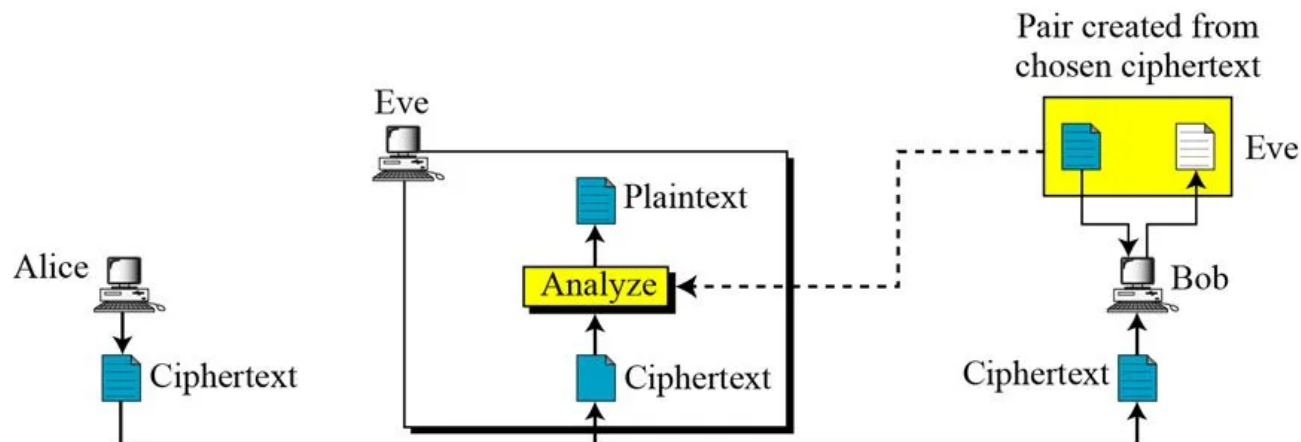
5. Chosen-ciphertext attack

Diberikan:

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

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

Chosen-Ciphertext Attack



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.