Hacking n' Coding :: Use case

Felipe Zimmerle

felipe.costa@openbossa.org

Hacking n' Coding :: Use case

Felipe Zimmerle

felipe.costa@openbossa.org

Hacking n' Coding :: Use case

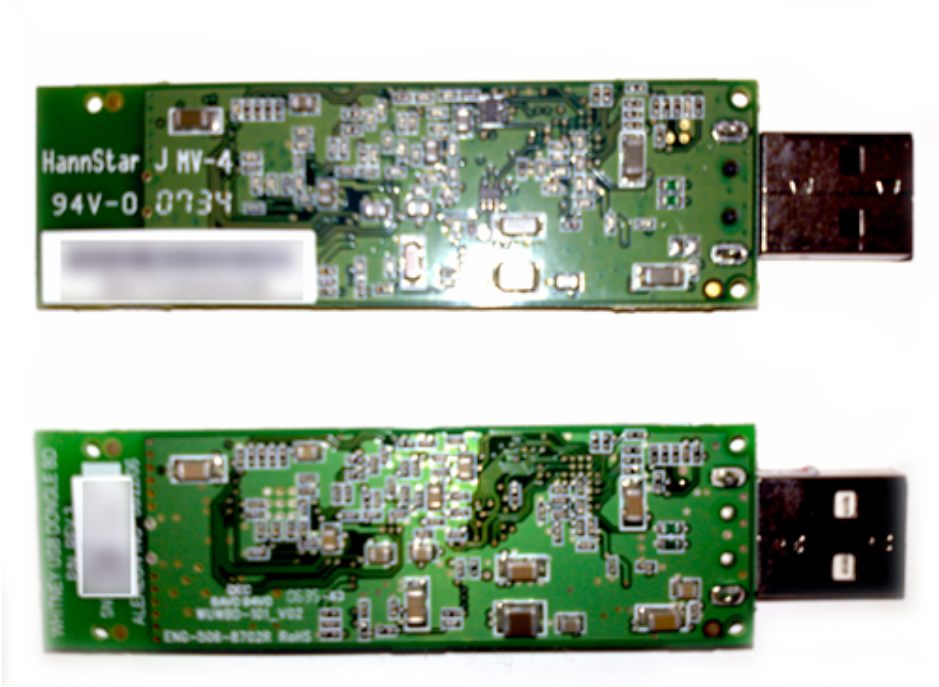
Conexões de periféricos

- **Escopo**: Linux, Windows, MacOS?! Dispositivos?!

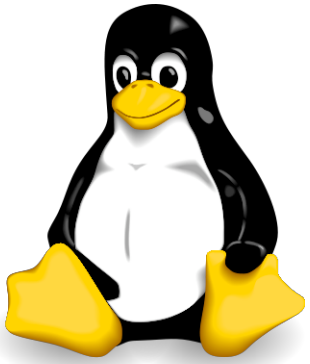
WUSB?

- **Gostaríamos de responder às perguntas:**
  - **É seguro?**
    - Canal de transmissão está OK?
      - Privacidade, Autenticidade e Integridade
    - Funciona? (resistente a ruído)
    - Possui bom desempenho? (a criptografia atrapalha?)

- Objetivo :: Análise de Segurança
  - “É seguro?”
- Tudo o que tínhamos:



WUSB, hã?



Linux

Funcionamento parcial

Windows

Driver e firmwares fechados e que sofriam várias alterações durante o ciclo de análise

MacOS

Existe alguma coisa? Alguém aqui sabe o quê?

E agora?

- **Fazer funcionar no Linux**
- **Drivers/Aplicações de terceiros, novas versões sendo lançadas sem controle (Windows)**
- **Nenhum suporte no MacOS**
- **Dispositivo com novas versões de hardware e firmware, updates sem preocupação com a compatibilidade**

Fazer funcionar no Linux

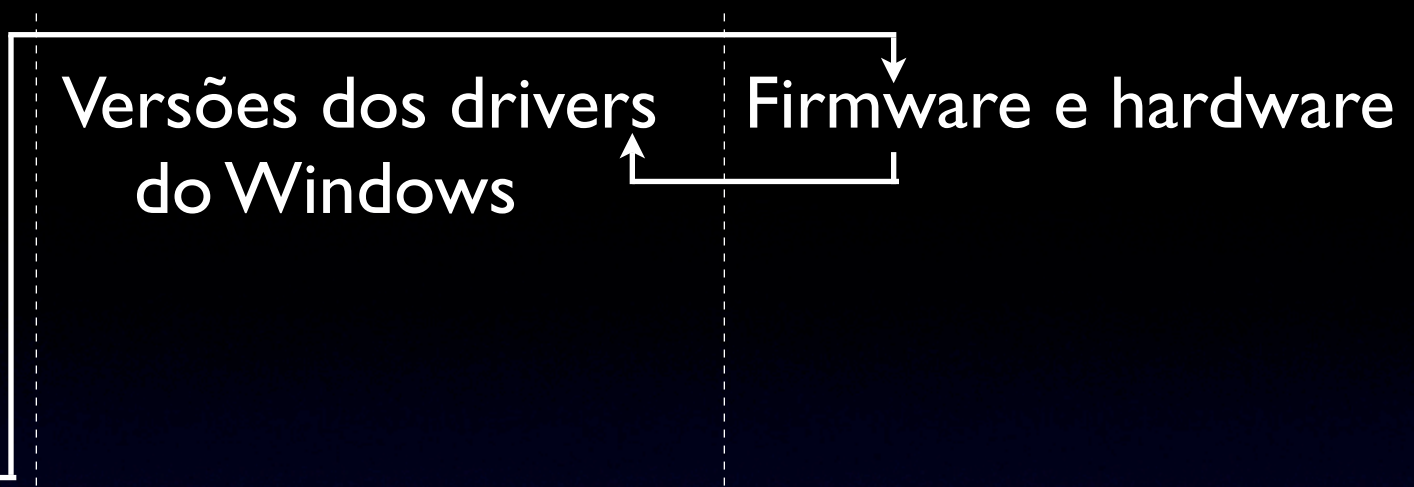
- Como funciona o upload de firmware?
- Arquivo do firmware?

Versões dos drivers do Windows

Firmware e hardware

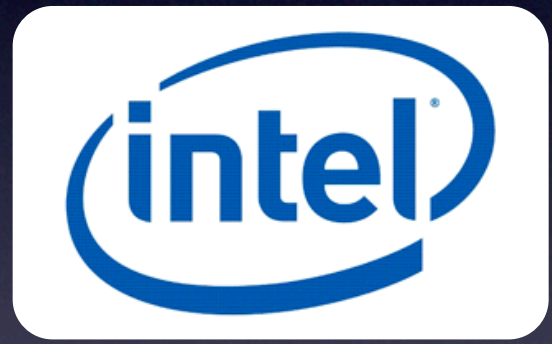
Fazer funcionar no Linux

- Como funciona o upload de firmware?
- Arquivo do firmware?



Drivers

- Driver da Intel
- Driver da IOGear #1
- Driver da IOGear #2



- Dentro de cada driver provavelmente um firmware !=

Drivers

- Sniff da comunicação...
  - Sniff USB
- Alguma idéia de como funciona o upload, existia alguma coisa no Linux
  - Análise do Sniffer x Implementação do Linux
    - Fácil separar firmware de protocolo de envio

“Pacote” USB

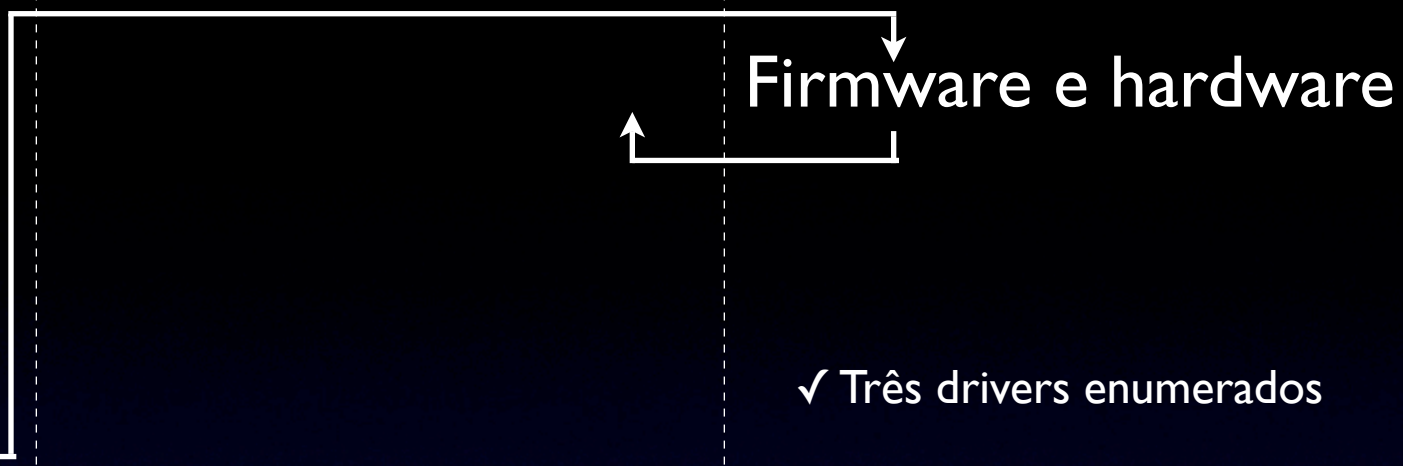
```
[12 ms]  >>>  URB 5 going down  >>>
-- URB_FUNCTION_CLASS_INTERFACE:
  TransferFlags          = 00000001 (USBD_TRANSFER_DIRECTION_IN, ~USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000100
  TransferBuffer         = 897ffd48
  TransferBufferMDL      = 00000000
  UrbLink                 = 00000000
  RequestTypeReservedBits = 00000001
  Request                 = 00000001
  Value                   = 00000000
  Index                   = 00000000
[12 ms] UsbSnoop - MyInternalIOCTLCompletion(bab39db0) : fido=00000000, Irp=896c8008, Context=897bd968, IRQL=2
[12 ms]  <<<  URB 5 coming back  <<<
-- URB_FUNCTION_CONTROL_TRANSFER:
  PipeHandle             = 89923990
  TransferFlags          = 0000000b (USBD_TRANSFER_DIRECTION_IN, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength = 00000019
  TransferBuffer         = 897ffd48
  TransferBufferMDL      = 8a37b4c0
    00000000: 19 00 02 00 00 01 00 01 00 00 00 00 00 00 02

    00000010: 00 01 00 01 00 6c 00 00 00


  UrbLink                = 00000000
  SetupPacket            =
    00000000: a1 01 00 00 00 00 00 01
```

video4linux

Fazer funcionar no Linux



✓ Como funciona o upload de firmware?

✓ Arquivo do firmware?

- Automatizar a extração do firmware

✓ Três drivers enumerados

Firmware

- Firmware proprietário não pode ser distribuído
  - Um script para extraí-lo do driver do windows pode :)
- Encontrar o firmware dentro dos drivers
  - Script em Perl para extrair a informação necessária

Upload via user land

- Mais fácil de “depurar” problemas
- Computador (geralmente) não trava quando algo de errado acontece
- Mais fácil de testar
- API quase igual a do Kernel
- Fácil portabilidade

Fazer funcionar no Linux

- ✓ Como funciona o upload de firmware?
- ✓ Arquivo do firmware?
- ✓ Automatizar a extração do firmware
- ✓ Upload via user land

Versões dos drivers do Windows

- ✓ Três drivers enumerados

Firwmare e hardware

- ✓ Três drivers enumerados

Dispositivos

- ✓ Três dispositivos (aparentemente) funcionando

Apresentação (CBA/DWA)

- Apresentação através da USB com fio
- Apresentação com troca de chaves
  - Apresentação tanto entre o host e o dispositivo quanto entre o dispositivo e o host
- Chave trocada é utilizada

para futuras conexões

Associação com sucesso

dispositivo ligado

USBs enumeradas

descricao da interface enviada

Part #1: USB

requisição de informações p/ associação

informação de associação

requisição sobre informações da conexão

informação da conexão

resposta a associação

Part #2: CBAF

Análise do CBA

- Sniffer a partir da associação do Windows
  - Comparação do resultado com documento de especificação
  - Comparação com a implementação do Linux
    - Modificações para entrada na mainline do Kernel

Análise do canal cifrado

- Equipamentos para "sniffer" dos pacotes WUSB muito caros.
  - Análise apenas da implementação a partir do código fonte

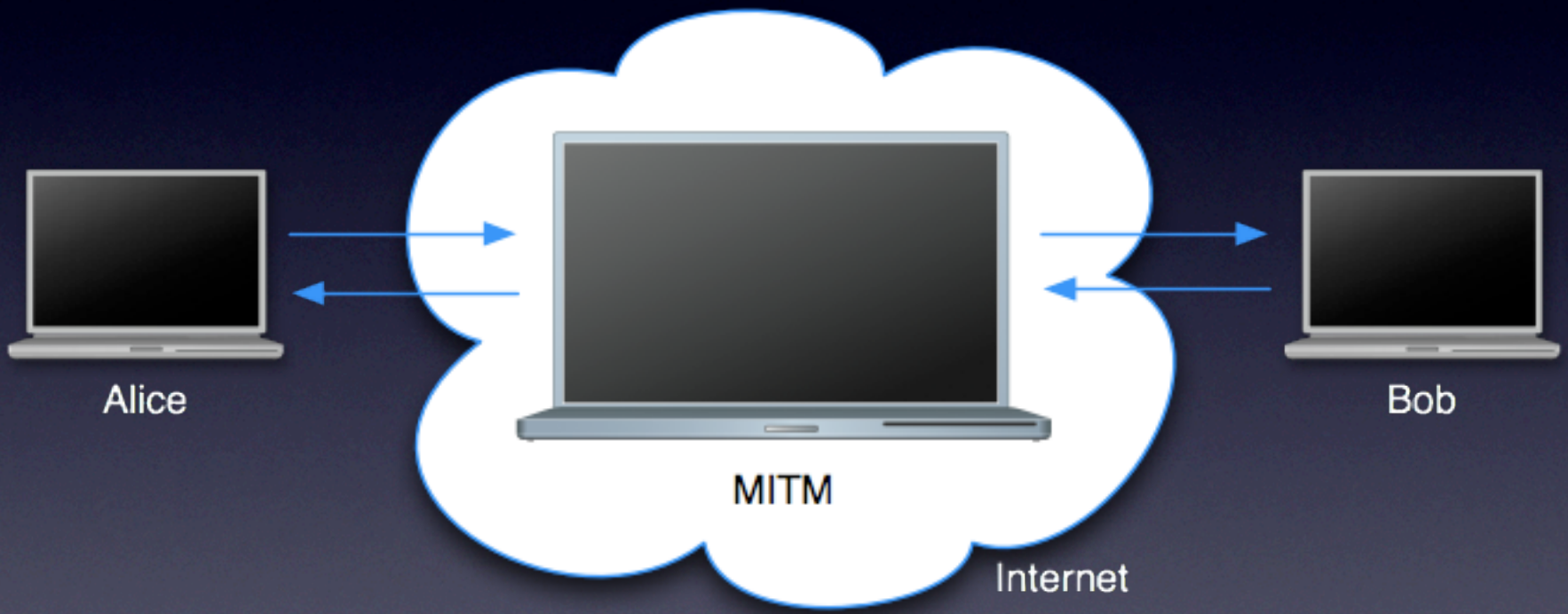
Resultados

- Aspectos práticos
- Aspectos teóricos (updates?)
- Quanto a implementação
  - Windows
  - Linux
  - MacOS

Comparativo

	USB com fio	**USB sem fio**
Ataques passivos	Impossível	Fácil
Integridade	Garantida	Garantida
Autenticidade	Garantida	Garantida
Confidenciabilidade	Garantida	Garantida
Conexão acidental	Impossível	Fácil
MITM	Impossível	Desconhedico

MITM



* MITM: *Man in the middle*

Conexões acidentais

Host

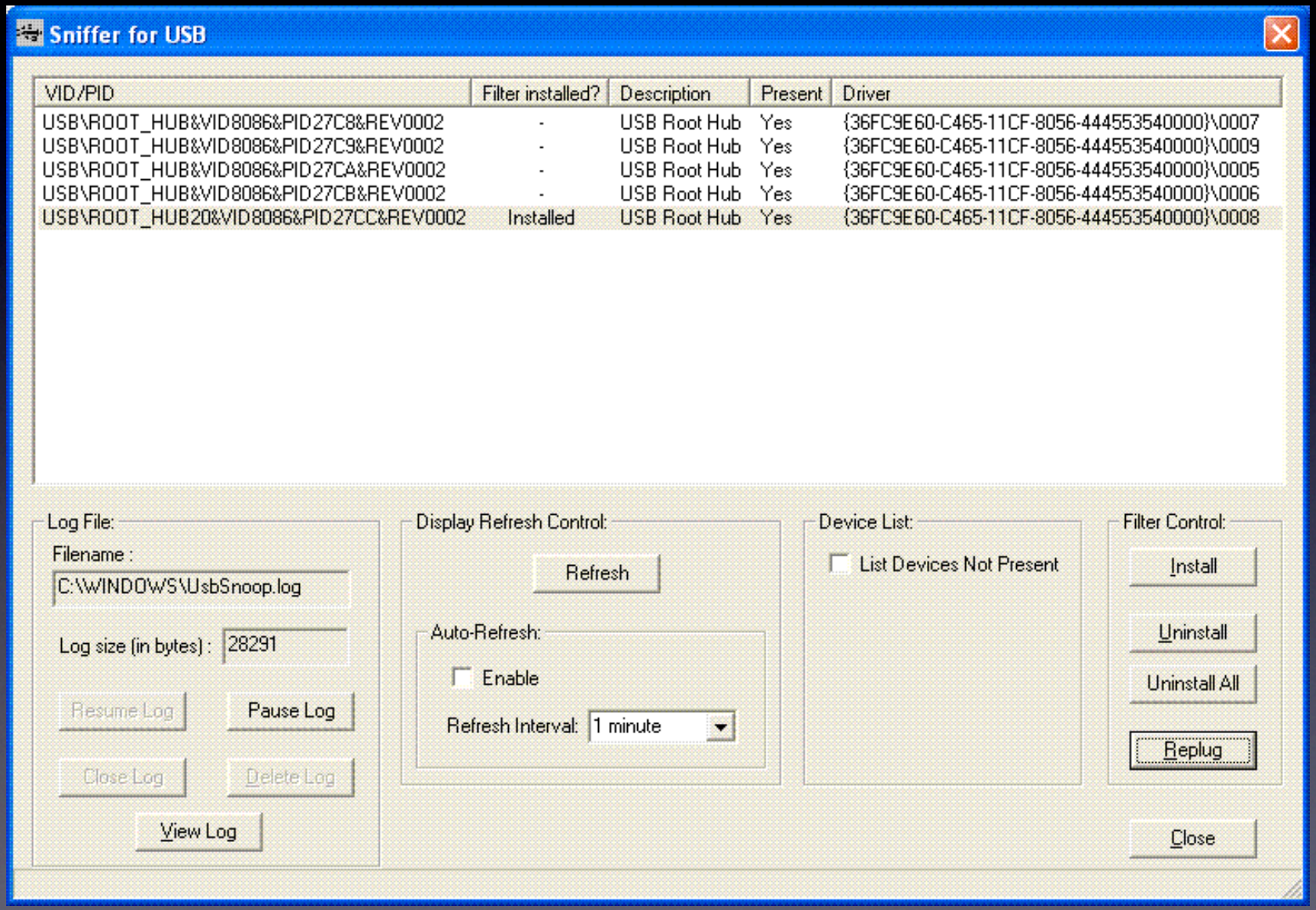
“X”

“X”

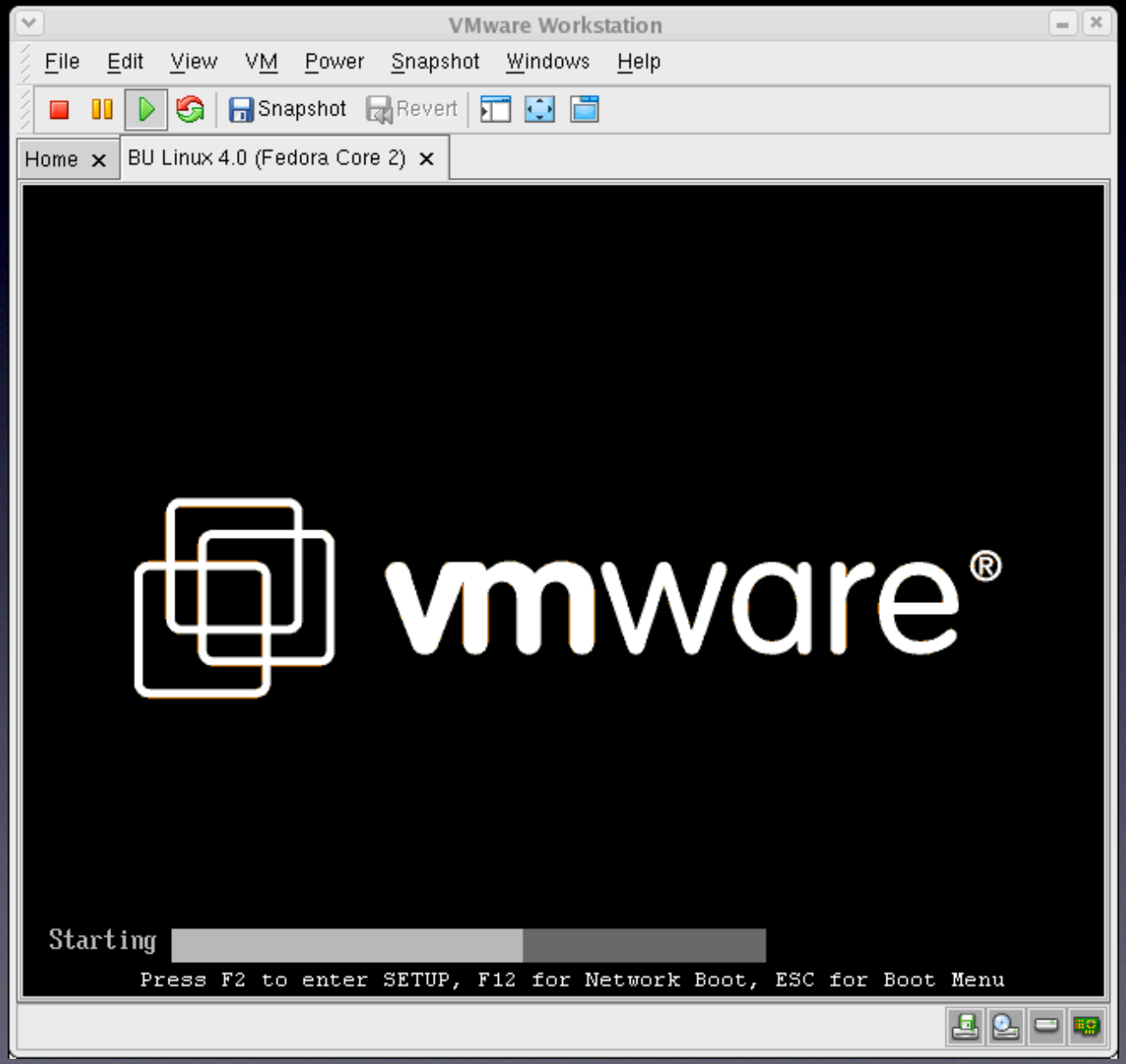
Sniffer USB

- Alguns softwares conhecidos são:
  - USBSnoop
  - VMWare + usbmon
  - WireShark + ubsmon
  - Hardware

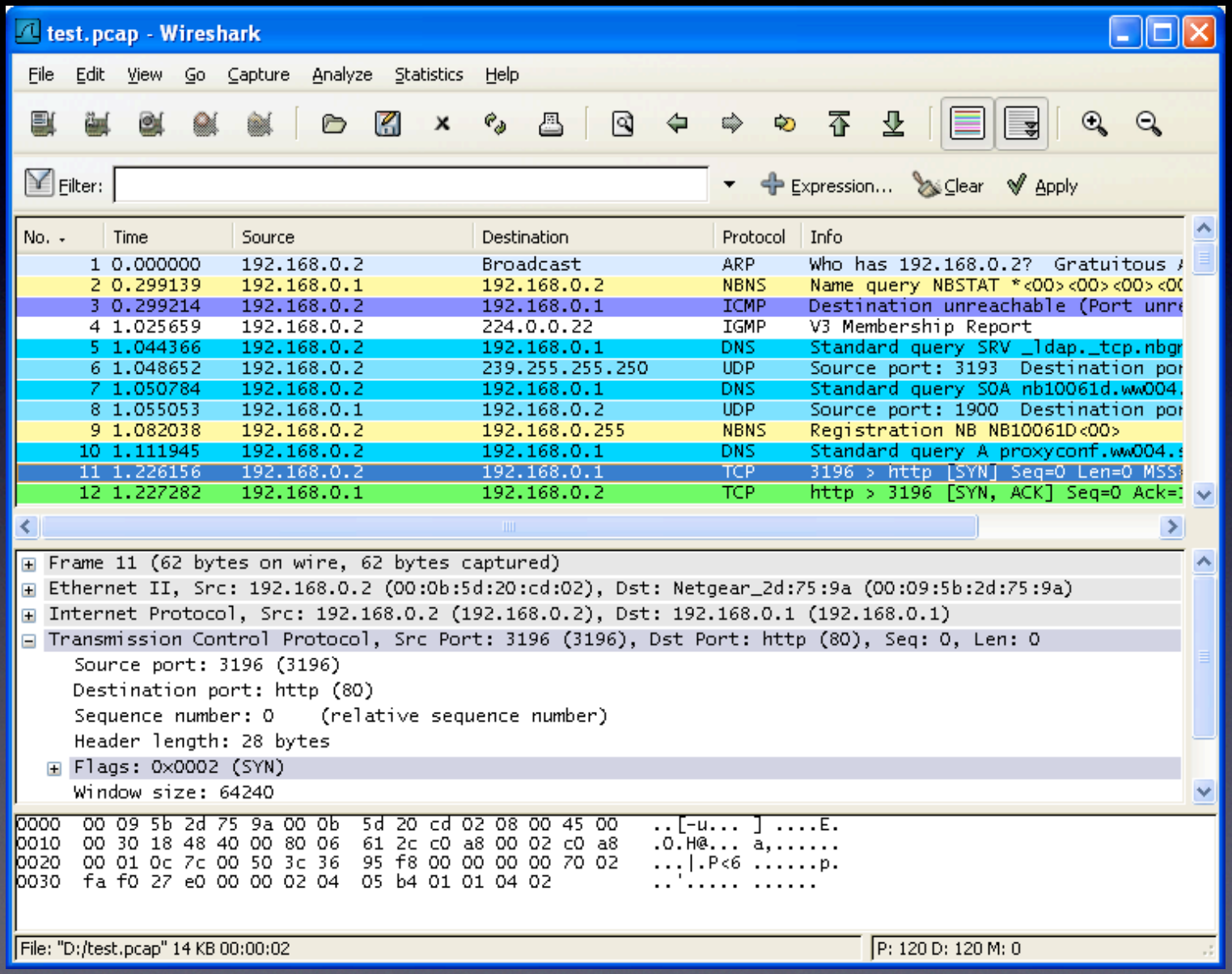
UsbSnoop


Sniffer for USB
VID/PID
Filter installed?
Description
Present
Driver
USB\ROOT_HUB&VID8086&PID27C8&REV0002 - USB Root Hub Yes {36FC9E60-C465-11CF-8056-444553540000}\0007
USB\ROOT_HUB&VID8086&PID27C9&REV0002 - USB Root Hub Yes {36FC9E60-C465-11CF-8056-444553540000}\0009
USB\ROOT_HUB&VID8086&PID27CA&REV0002 - USB Root Hub Yes {36FC9E60-C465-11CF-8056-444553540000}\0005
USB\ROOT_HUB&VID8086&PID27CB&REV0002 - USB Root Hub Yes {36FC9E60-C465-11CF-8056-444553540000}\0006
USB\ROOT_HUB20&VID8086&PID27CC&REV0002 Installed USB Root Hub Yes {36FC9E60-C465-11CF-8056-444553540000}\0008
Log File:
Filename :
C:\WINDOWS\UsbSnoop.log
Log size (in bytes) :
28291
Resume Log
Pause Log
Close Log
Delete Log
View Log
Display Refresh Control:
Refresh
Auto-Refresh:
Enable
Refresh Interval:
1 minute
Device List:
List Devices Not Present
Filter Control:
Install
Uninstall
Uninstall All
Replug
Close

VMWare + usbmon



Wireshark + usbmon



USBKitty

UWBTracer

LeCroy UWBTracer Protocol Analyzer - [C:\Program Files\CATC\UWBTracer\Sample Files\SampleWUSB.uwb]

File Setup Record Report Search View Window Help

WUSB Xaction	Control	Host Id	Addr	Endp	D	Tp	R	bRequest	wValue	wIndex	wLength	Seq #	Data Len	Seq #	Data Len	bvDINAck	ACK'd Pkts	Xact Len
0	Setup+IN	0x0080	10	0	H->D	S	I	0x02	0x0403	0x0605	2055	0	3	1	3	0x00000003	2	6

WUSB Xaction	Isoch	Host Id	Addr	Endp	Seq #	Data Len	Seq #	Data Len	Seq #	Data Len	bvAckCode	ACK'd Pkts	Xact Len	Failed Pkts	Failed Data
1	OUT	0x0080	20	2	0	2	2	2	3	2	0x0000000D	3	6	1	2

WUSB Packet	MMC	Dir	Host Id	Chan TS	IE Id	CTA 0	Addr	Endp	CTA 1	Addr	Endp	EDir	bvDINAck	PHY_ACT	Idle	Time Stamp
4		-->	0x0080	0:022	WCTA	DR	20	2	DT (Hnd)	20	2	OUT	0x00000000	3.045 µs	1.000 ms	0.000024650

MPI		PHY	Rate	Len	Scr	BM	PreTyp	TF Code	BG	RSSI	LQI	MAC	Control	Dest ID	Src ID	Policy	Rtry	Type	Frag#	M Frg	Durati
8	RX		0x07	28	1	0	0	1	0	0x00	0x00			0x0000	0x0080	No	0	App Specific	0x0	0x0	00

WUSB Packet	OUT	Dir	Host Id	IDATA	Addr	Endp	Data	PHY_ACT	Idle	Time Stamp
5		-->	0x0080	1	20	2	2 bytes	2.571 µs	100.004 µs	0.001027700

MPI		PHY	Rate	Len	MAC	Data	Dest ID	Src ID	Payload	FCS	PHY_ACT	Delta Time	Idle	Time Stamp
9	RX		0x07	9			0x0014	0x0080	9 bytes	0x00000000	2.571 µs	102.575 µs	100.004 µs	0.001027700

WUSB Packet	OUT	Dir	Host Id	IDATA	Addr	Endp	Data	PHY_ACT	Idle	Time Stamp
6		-->	0x0080	1	20	2	2 bytes	2.571 µs	1.004 µs	0.001130275

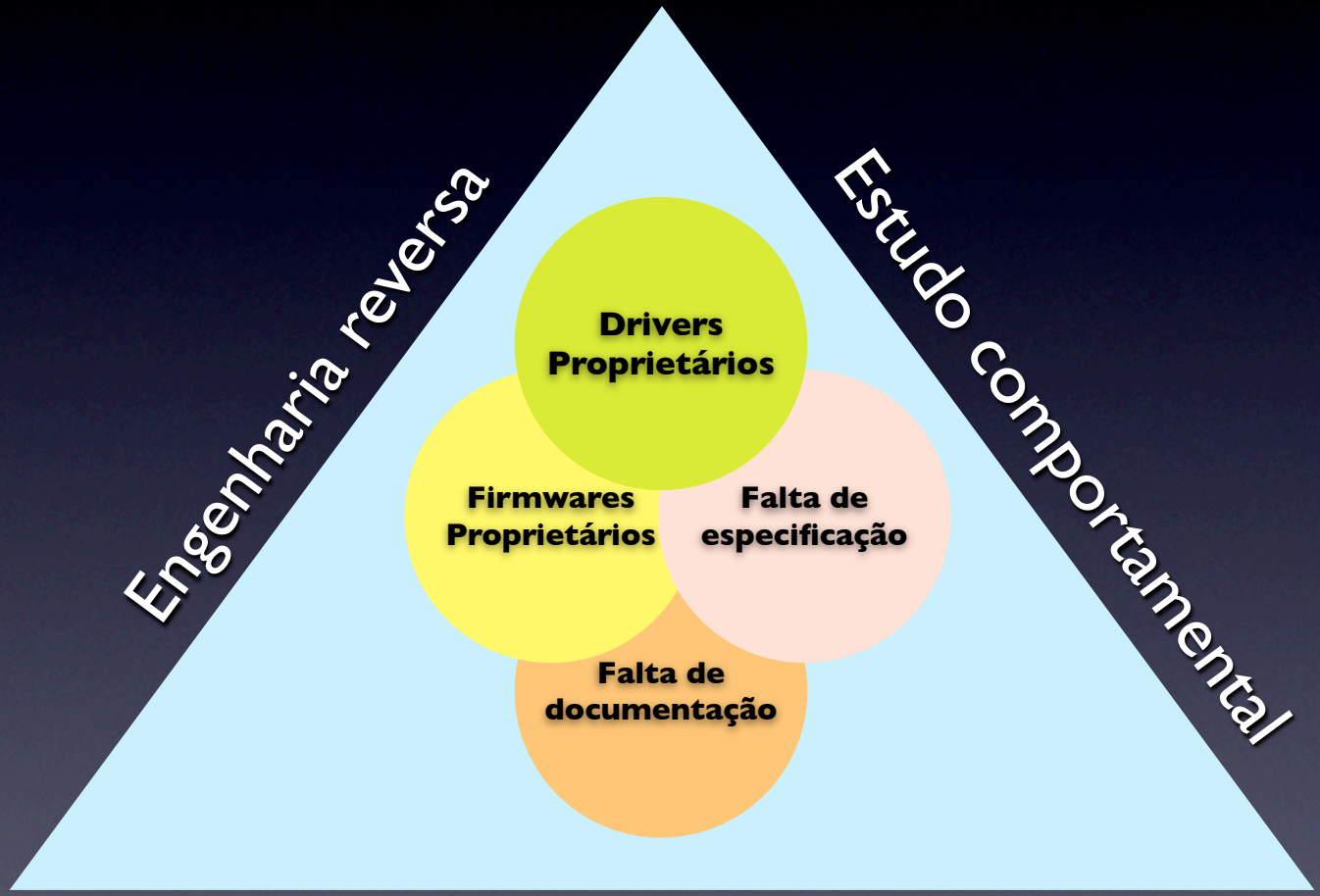
MPI		PHY	Rate	Len	MAC	Data	Dest ID	Src ID	Payload	FCS	PHY_ACT	Delta Time	Idle	Time Stamp
10	RX		0x07	9			0x0014	0x0080	9 bytes	0x00000000	2.571 µs	3.575 µs	1.004 µs	0.001130275

WUSB Packet	OUT	Dir	Host Id	IDATA	Addr	Endp	Data	PHY_ACT	Idle	Time Stamp
7		-->	0x0080	1	20	2	2 bytes	2.571 µs	100.004 µs	0.001133850

MPI		PHY	Rate	Len	MAC	Data	Dest ID	Src ID	Payload	FCS	PHY_ACT	Delta Time	Idle	Time Stamp
11	RX		0x07	9			0x0014	0x0080	9 bytes	0x00000000	2.571 µs	102.575 µs	100.004 µs	0.001133850

Este projeto sem os hacks?

Quebra-cabeças



Hacking n' Coding :: Use case

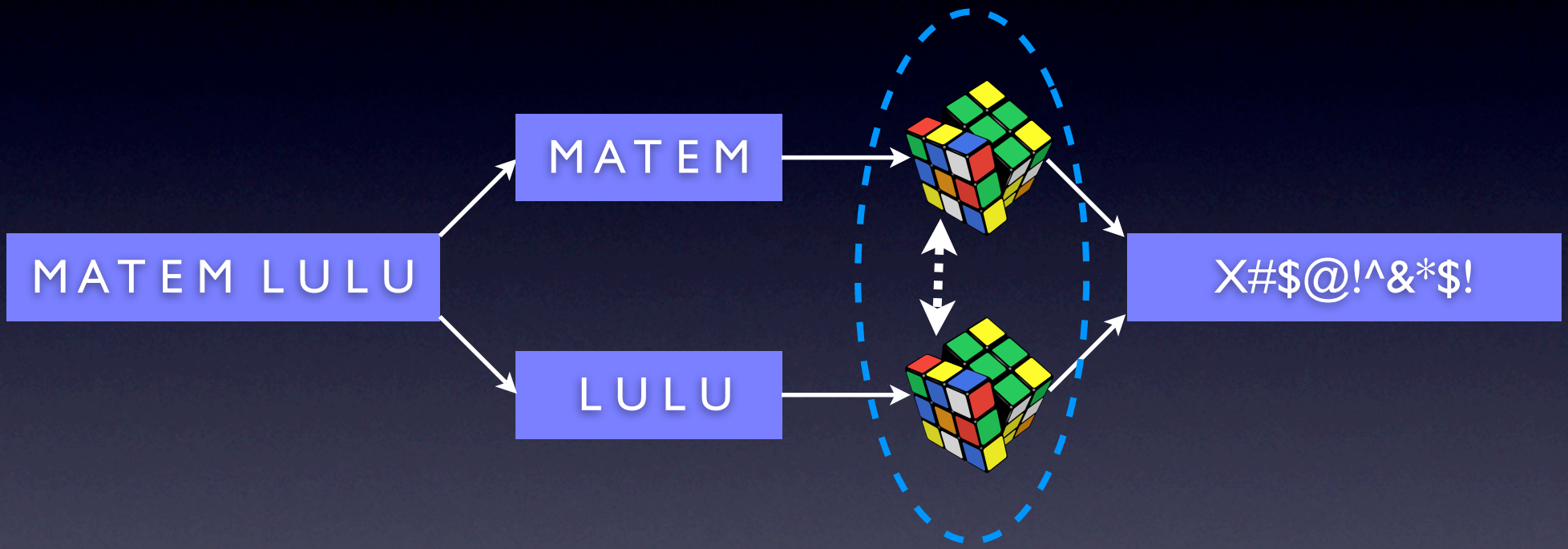
Perguntas?

Felipe Zimmerle

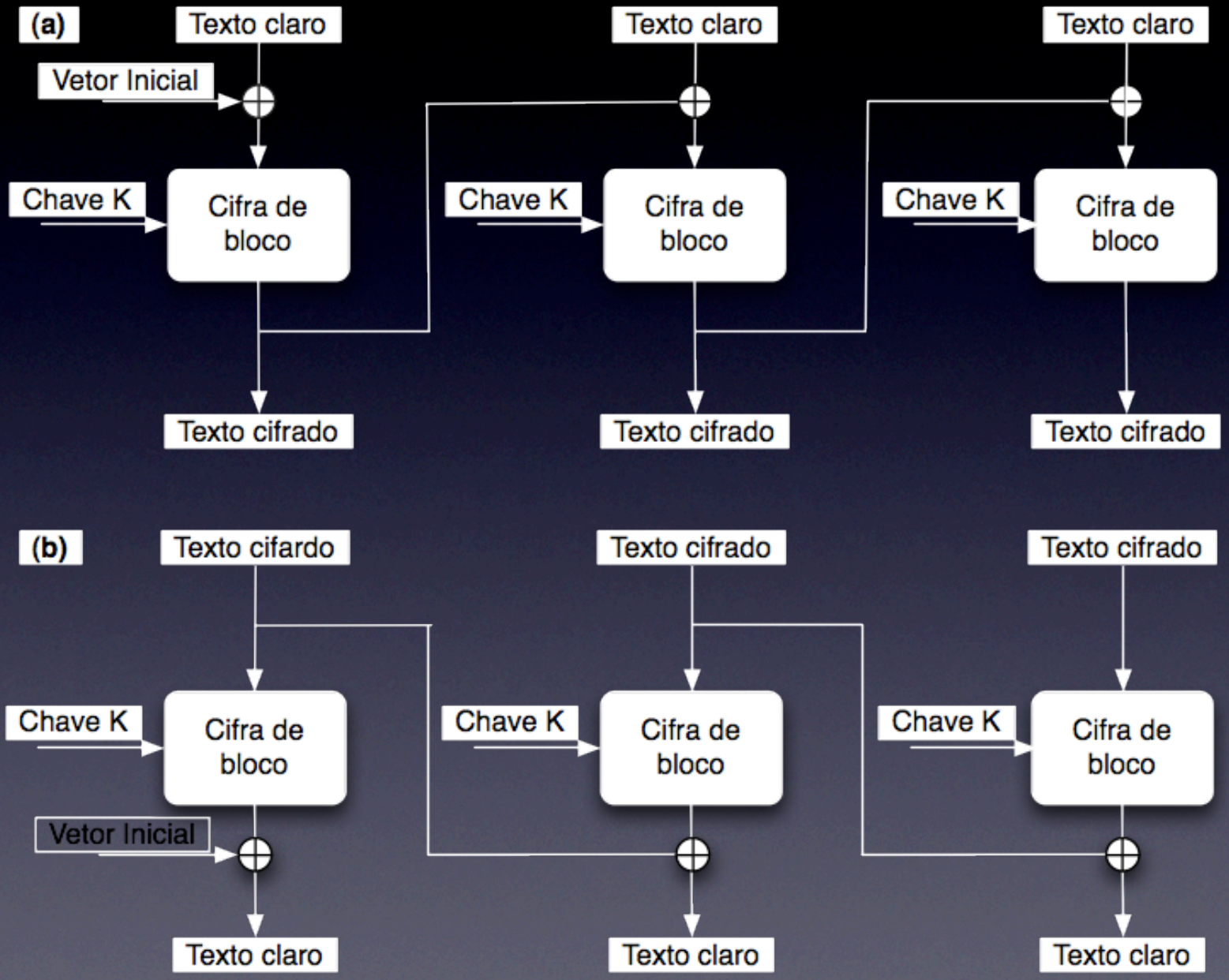
felipe.costa@openbossa.org

Slides de backup

Cifra de blocos

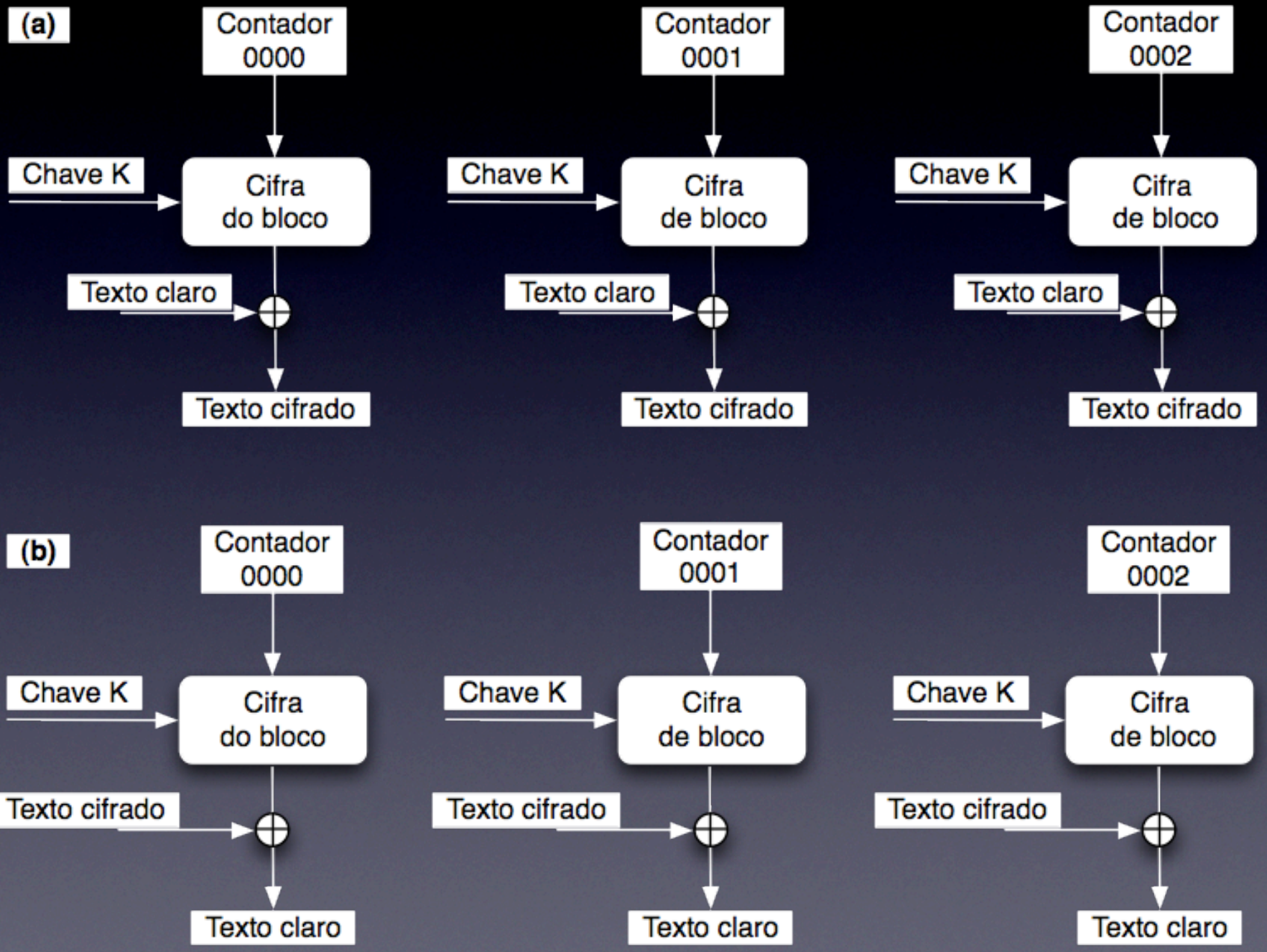


Modo CBC

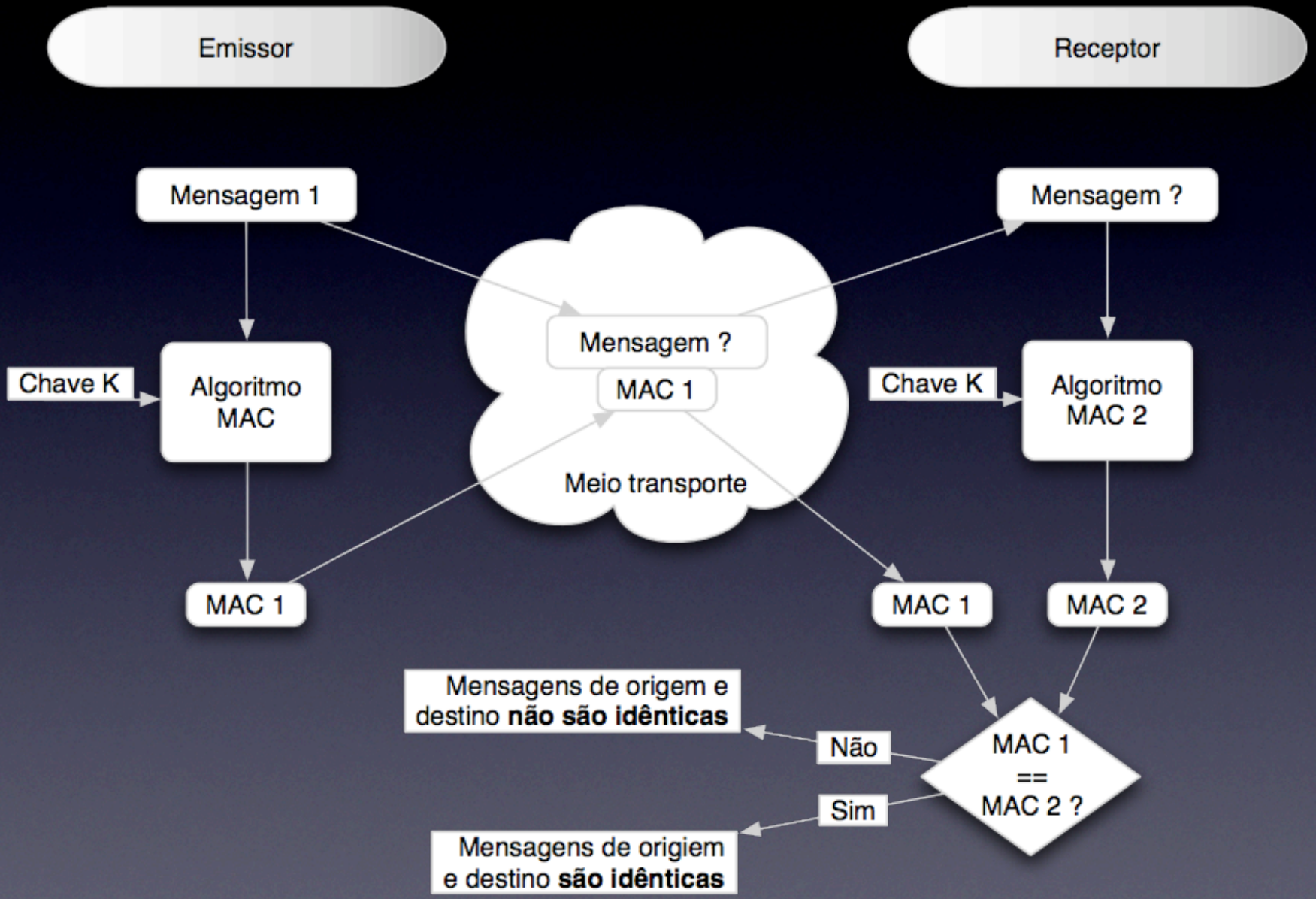


* CBC: *Cipher Block Chaining*

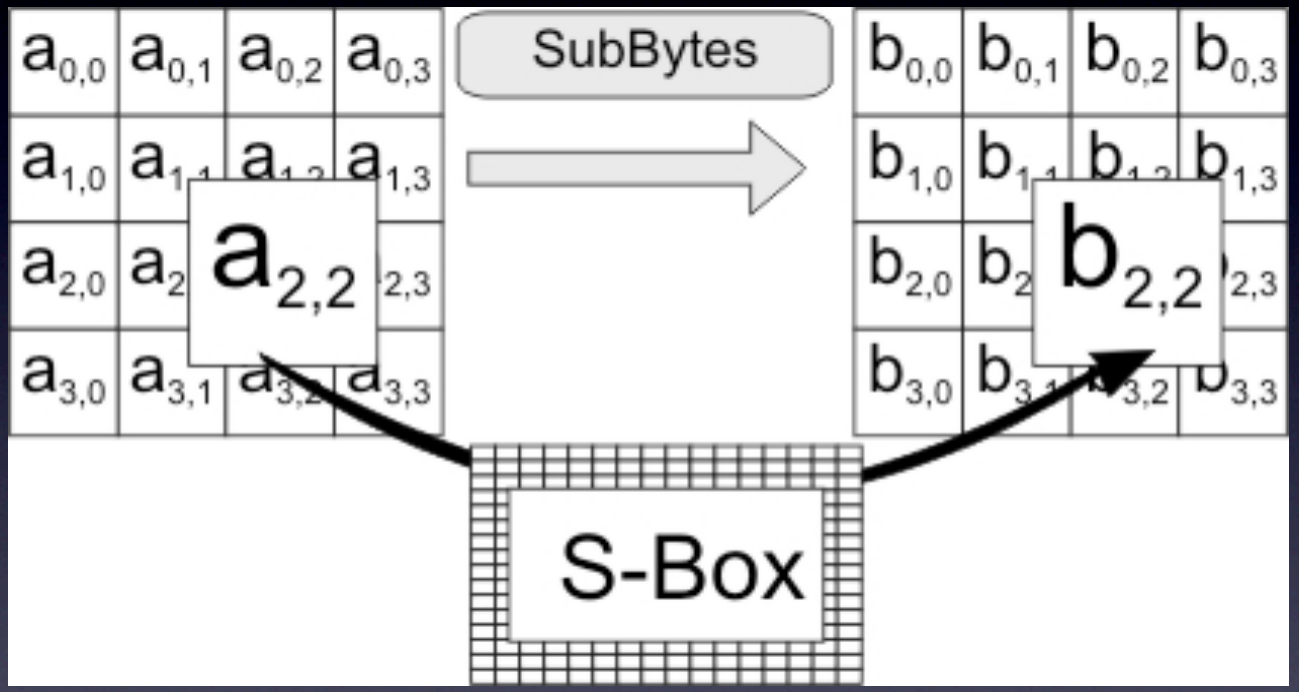
Modo *Counter*

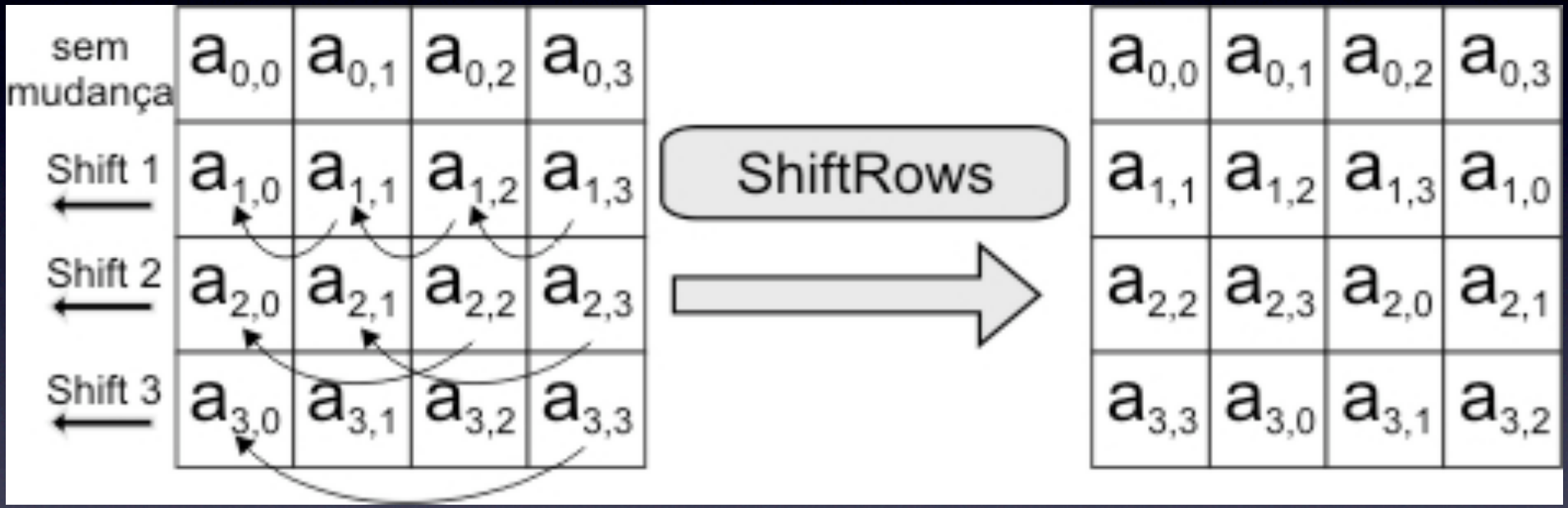


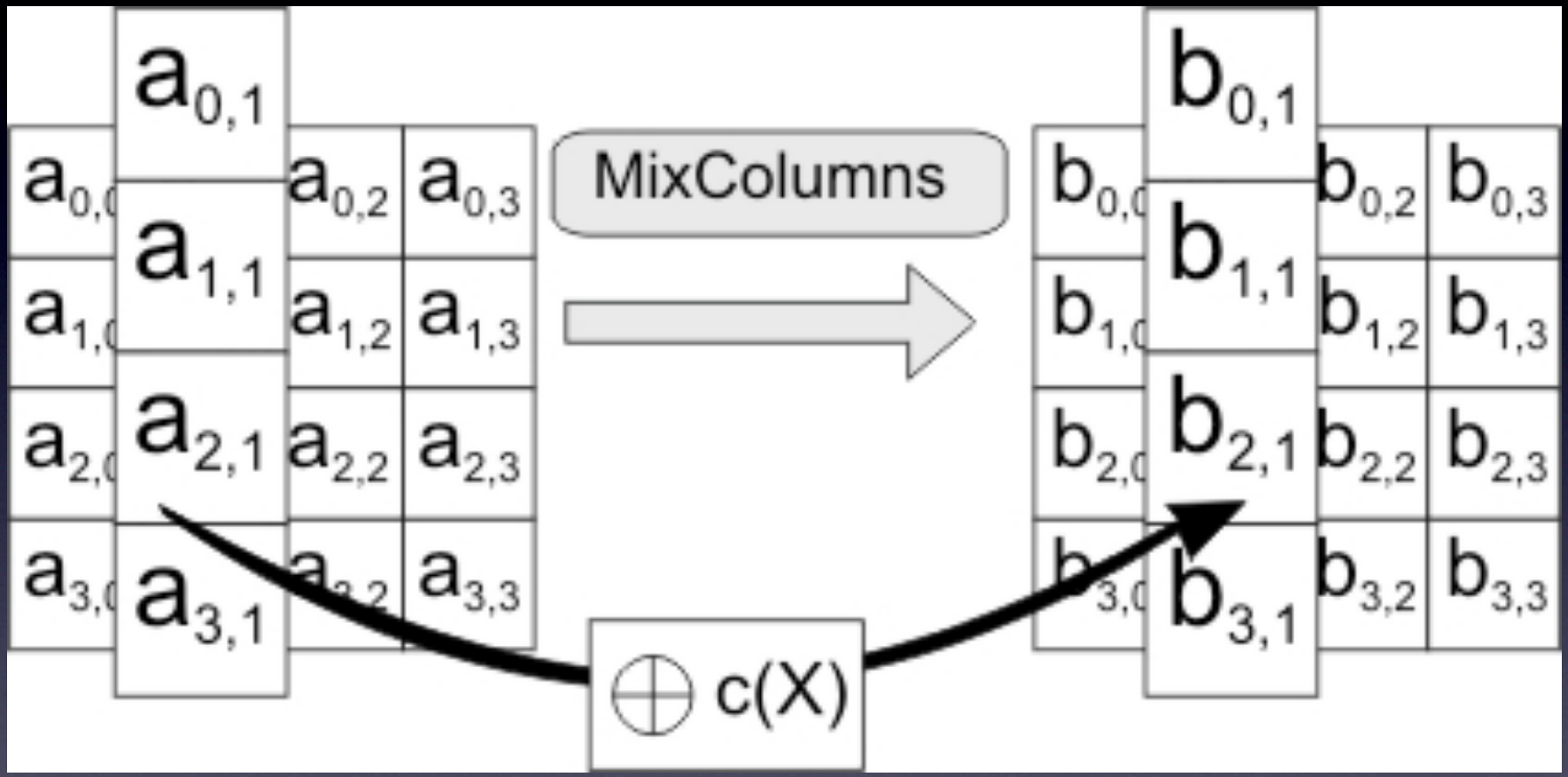
CBC-MAC / MAC ?

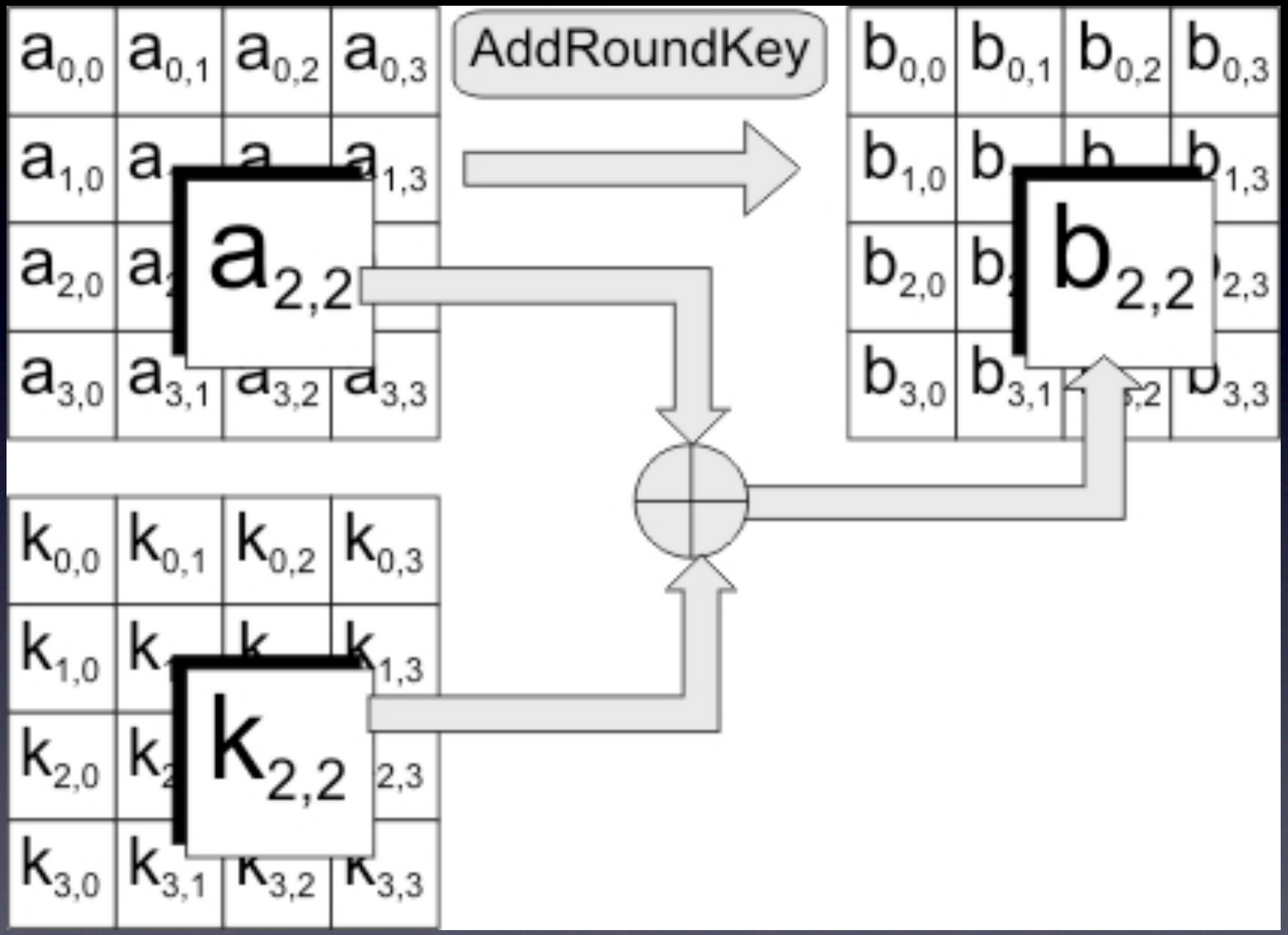


* MAC: *Message Authentication Code*


SubBytes
S-Box


sem mudança
Shift 1
Shift 2
Shift 3
a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
ShiftRows
a0,0 a0,1 a0,2 a0,3
a1,1 a1,2 a1,3 a1,0
a2,2 a2,3 a2,0 a2,1
a3,3 a3,0 a3,1 a3,2


MixColumns
c(X)


AddRoundKey