Let the BEAST of CRIME and TIME be not so LUCKY

Pratik Guha Sarkar

Takeaway

iSECpartners part of nccgroup

Background

- BEAST
- Lucky 13
- RC4 biases
- CRIME
- BREACH
- TIME

Prevention

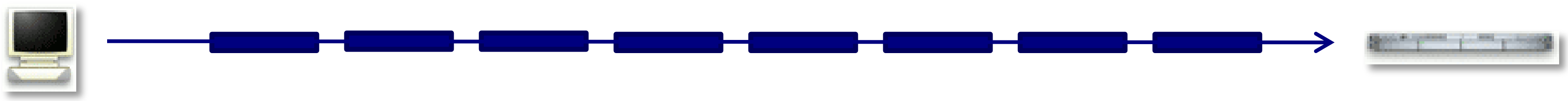
- With and
- without
- TLS 1.2

BEAST

BEAST - What is it ?

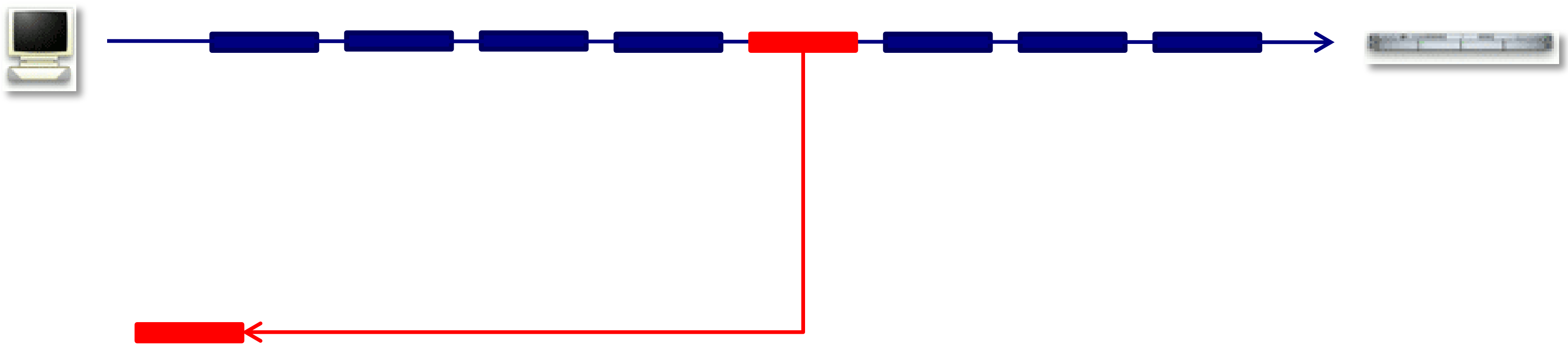
- **B**rowser **E**xploit **A**gainst **S**SL/**T**LS
- Used crypto flaws in SSL to recover plaintext cookies
- Refined previous attacks on CBC in SSL to make them practical
  - Innovation was exploiting **chosen boundary** capability

CBC Attack in SSL



CBC Attack in SSL

Target Block For Decrypting



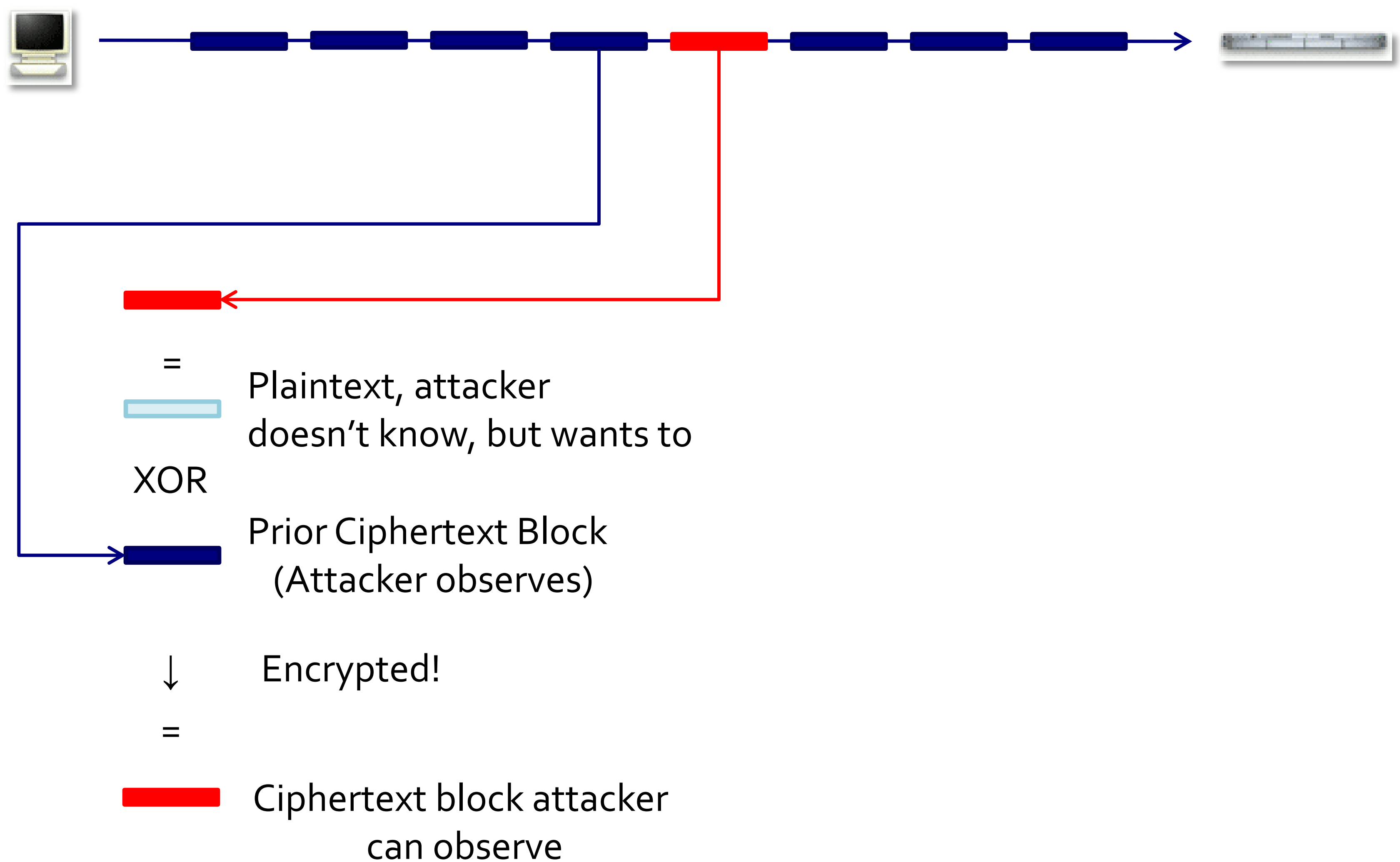
=

?

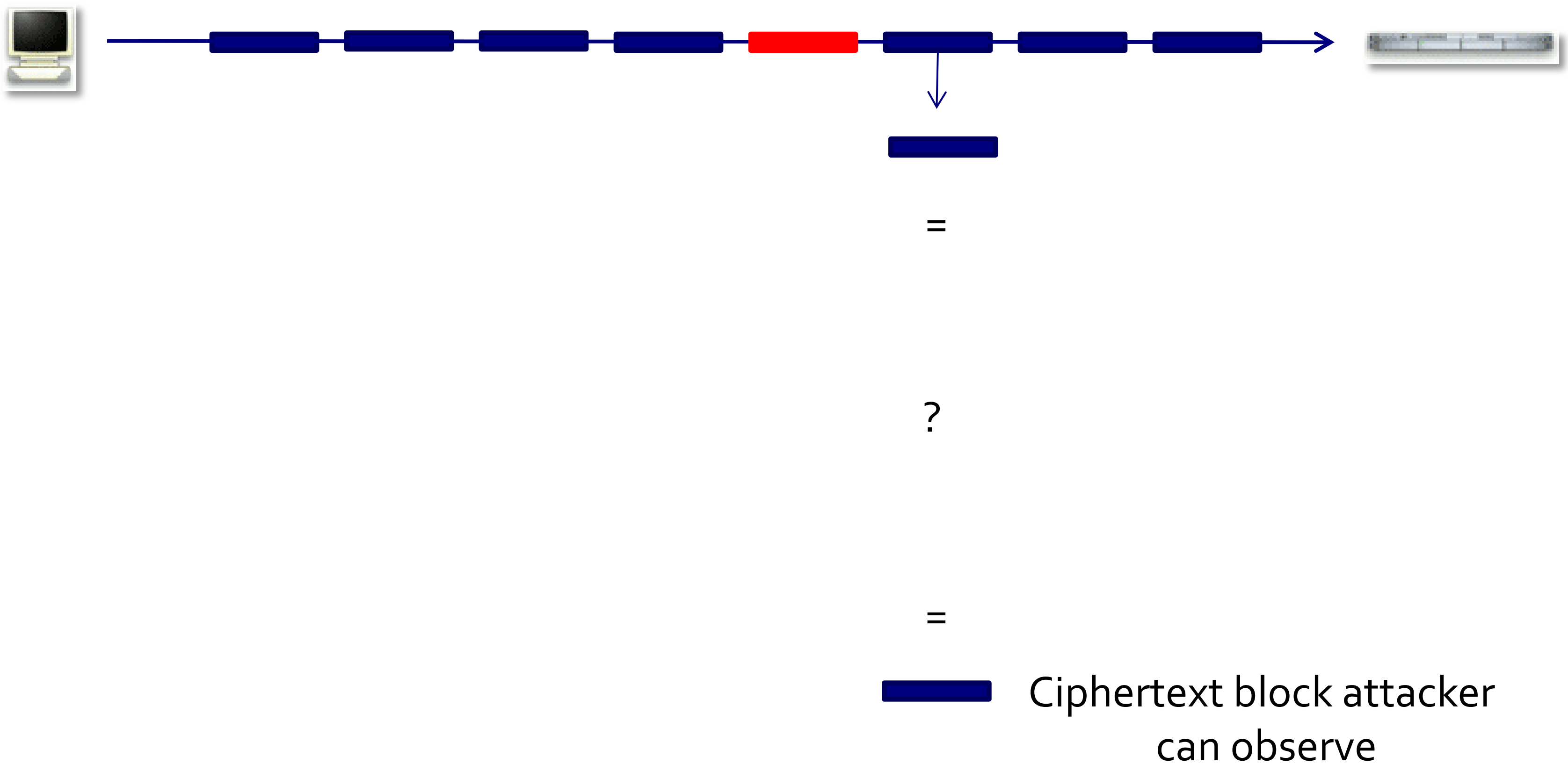
=

Ciphertext block attacker can observe

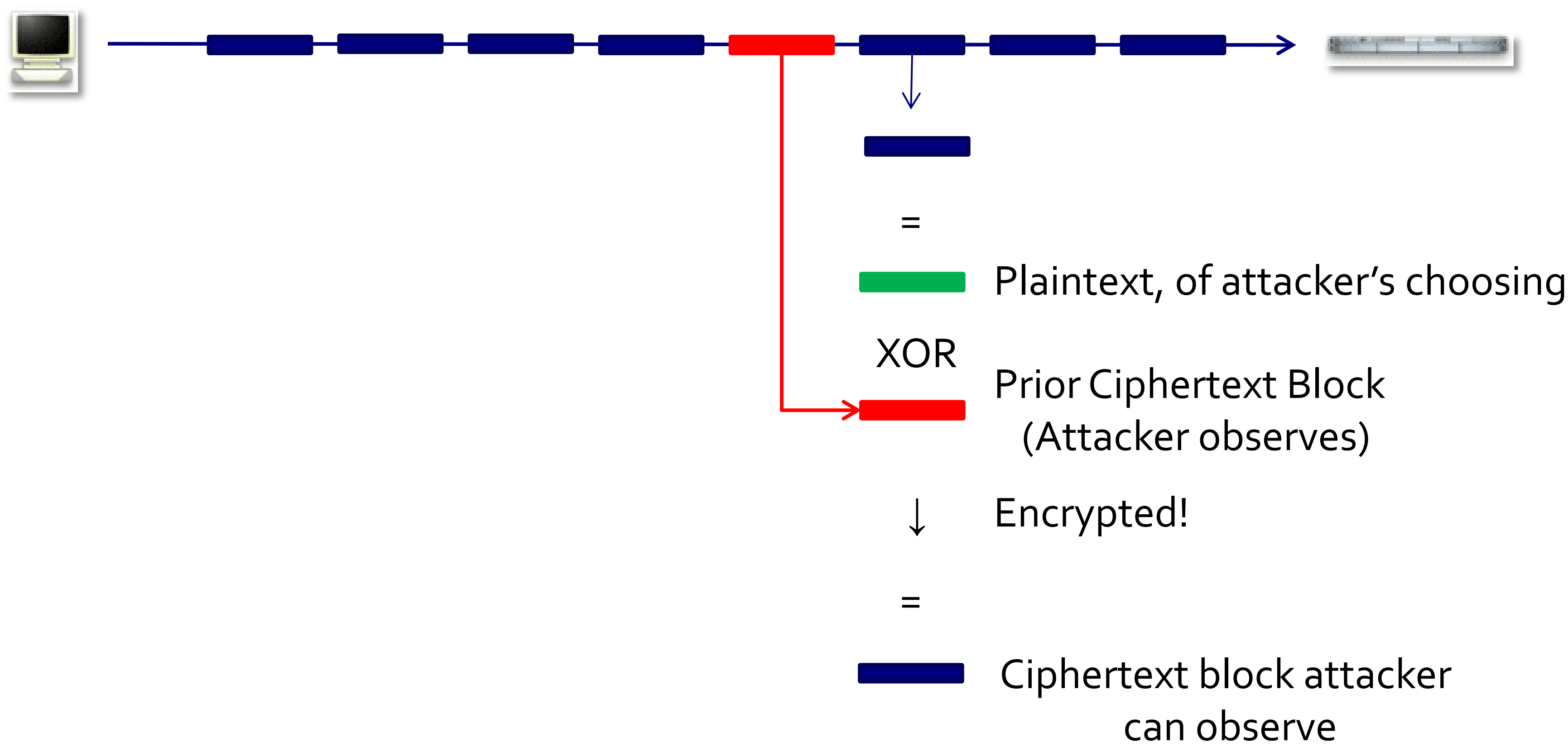
CBC Attack in SSL



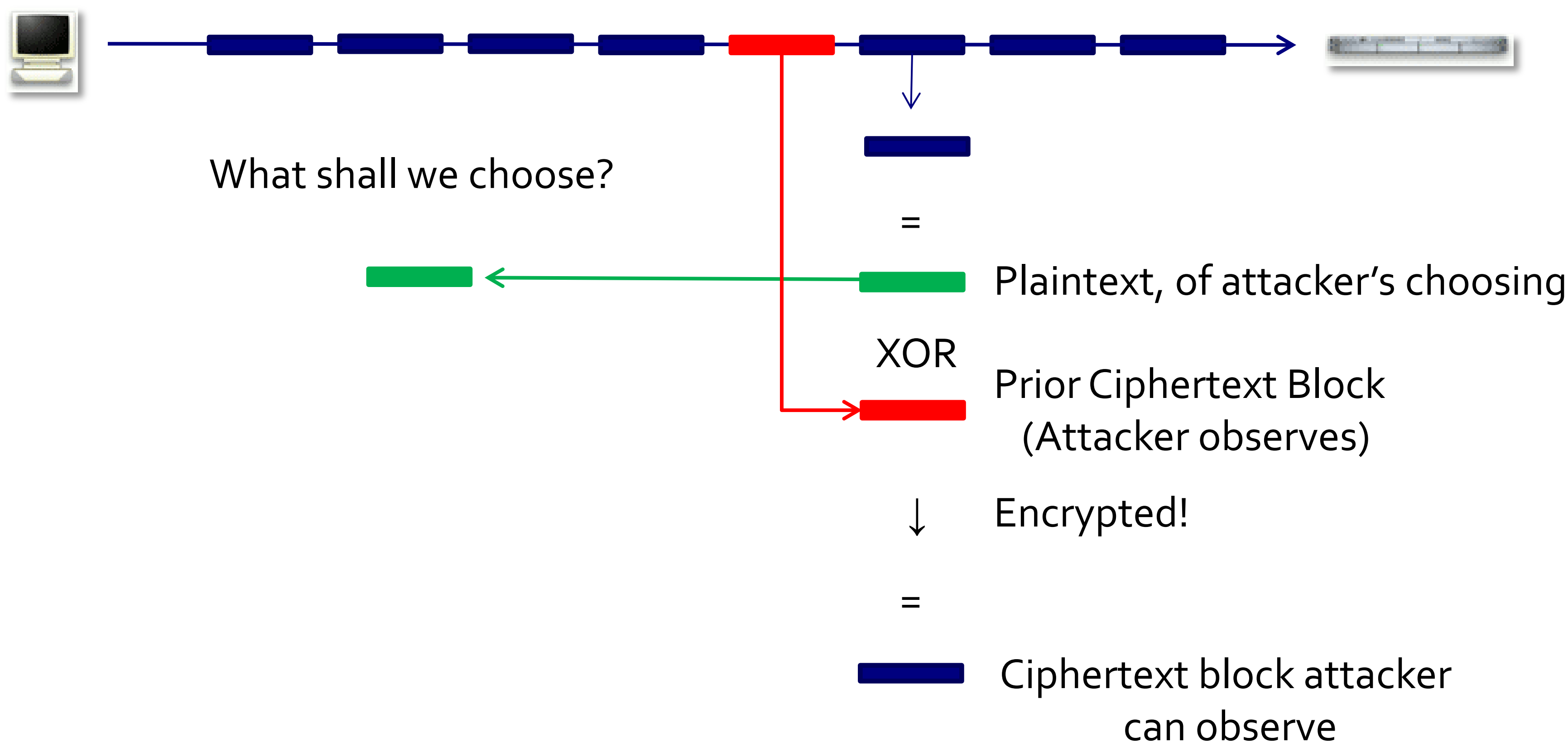
CBC Attack in SSL



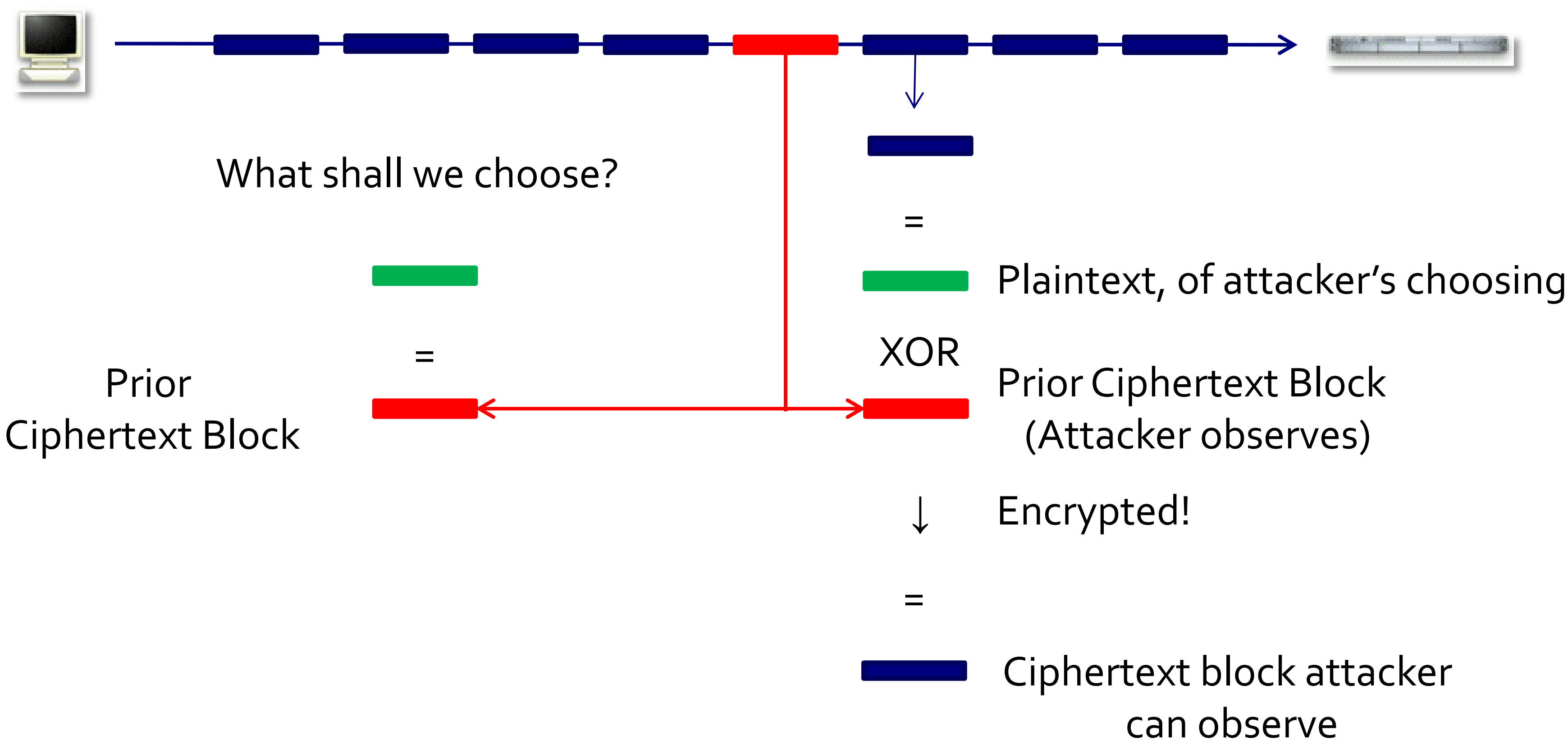
CBC Attack in SSL



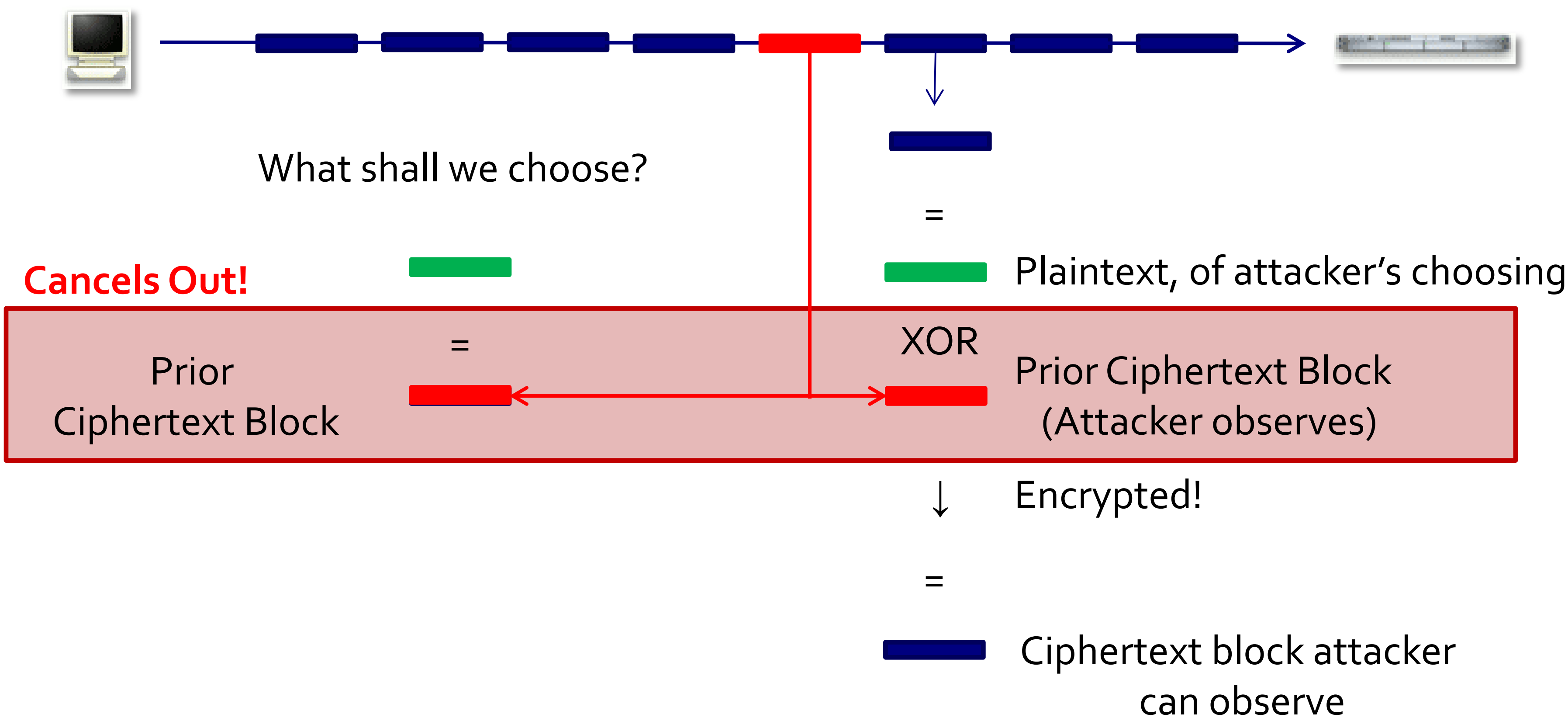
CBC Attack in SSL



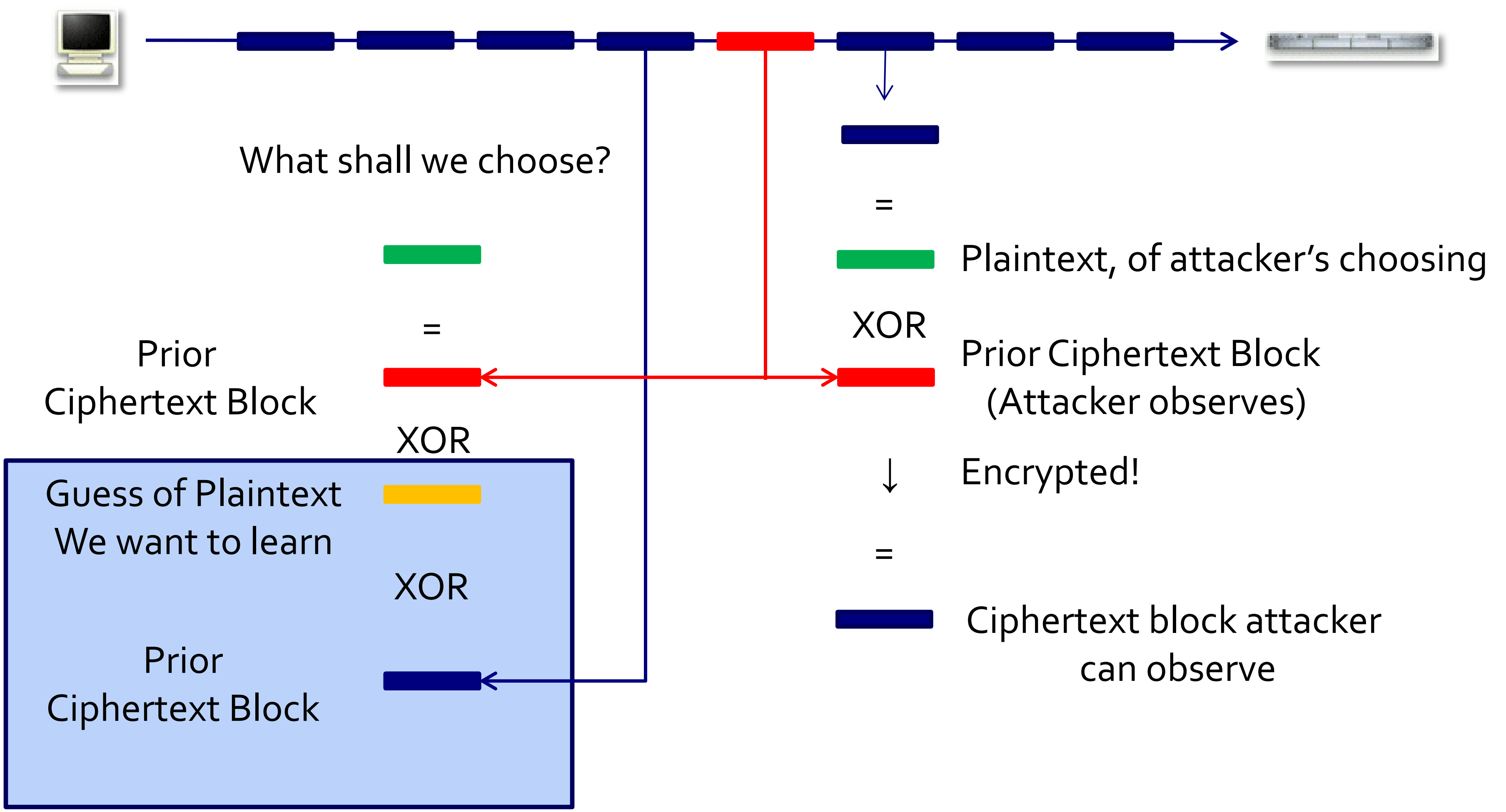
CBC Attack in SSL



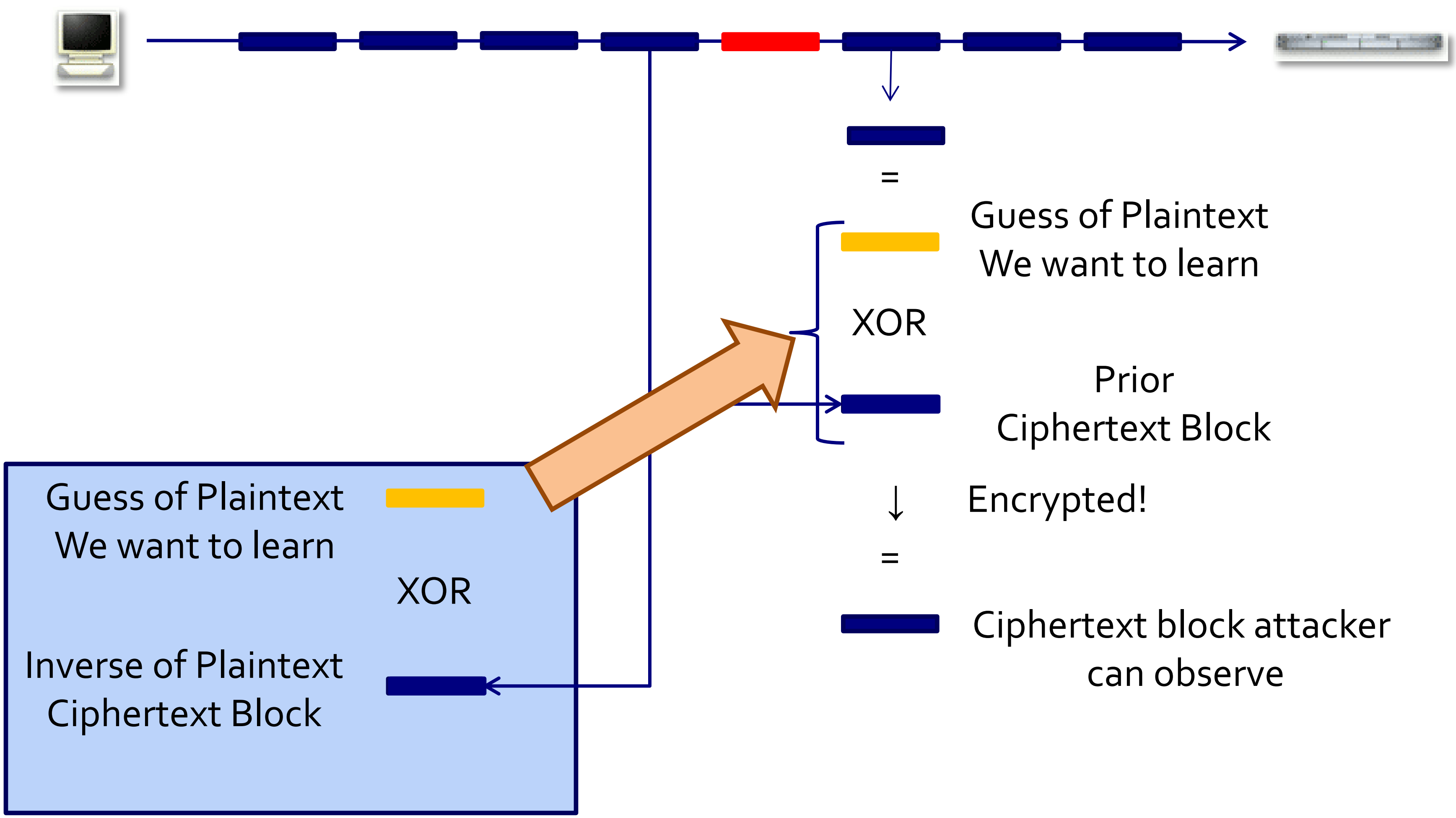
CBC Attack in SSL



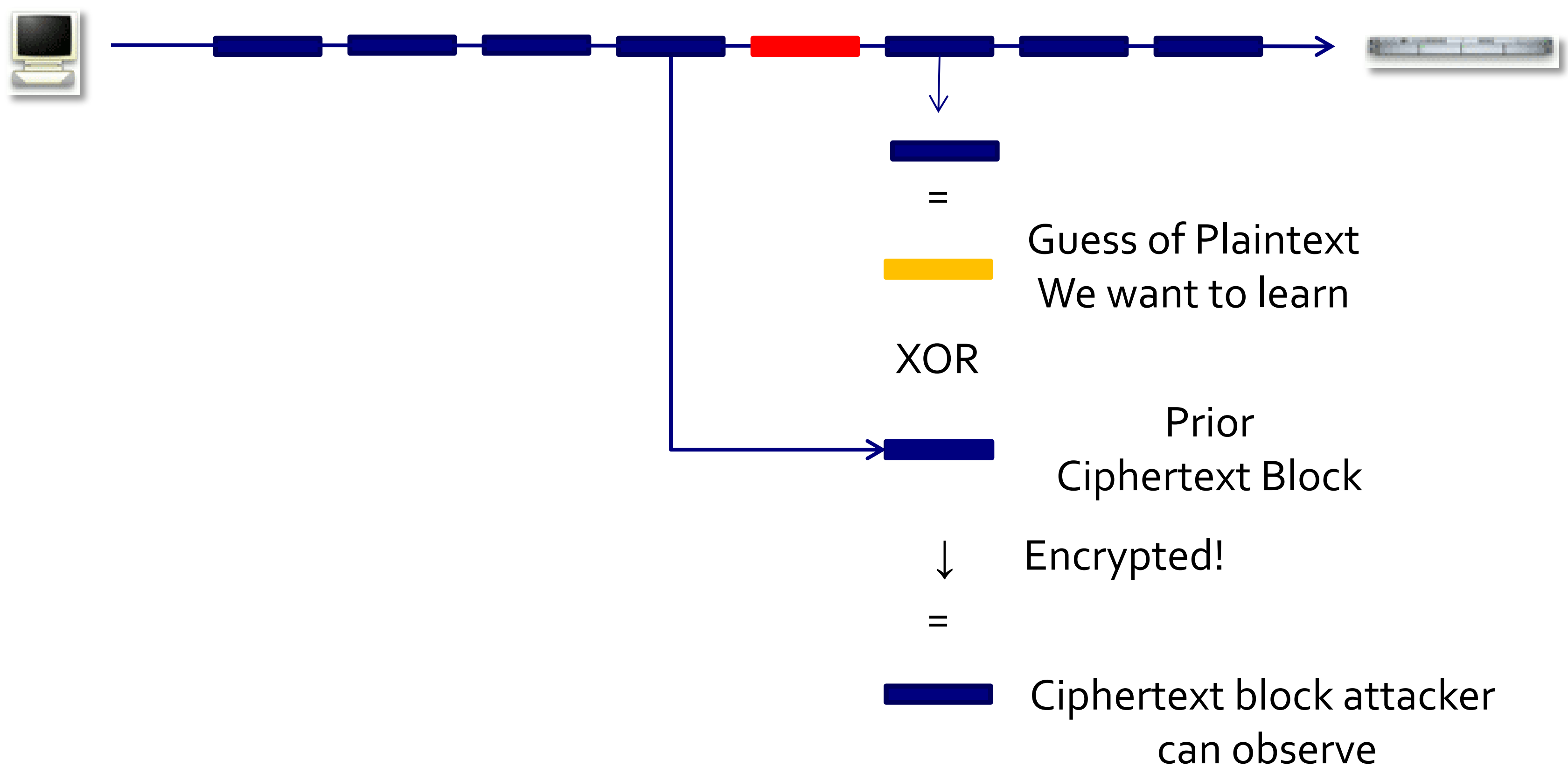
CBC Attack in SSL



CBC Attack in SSL

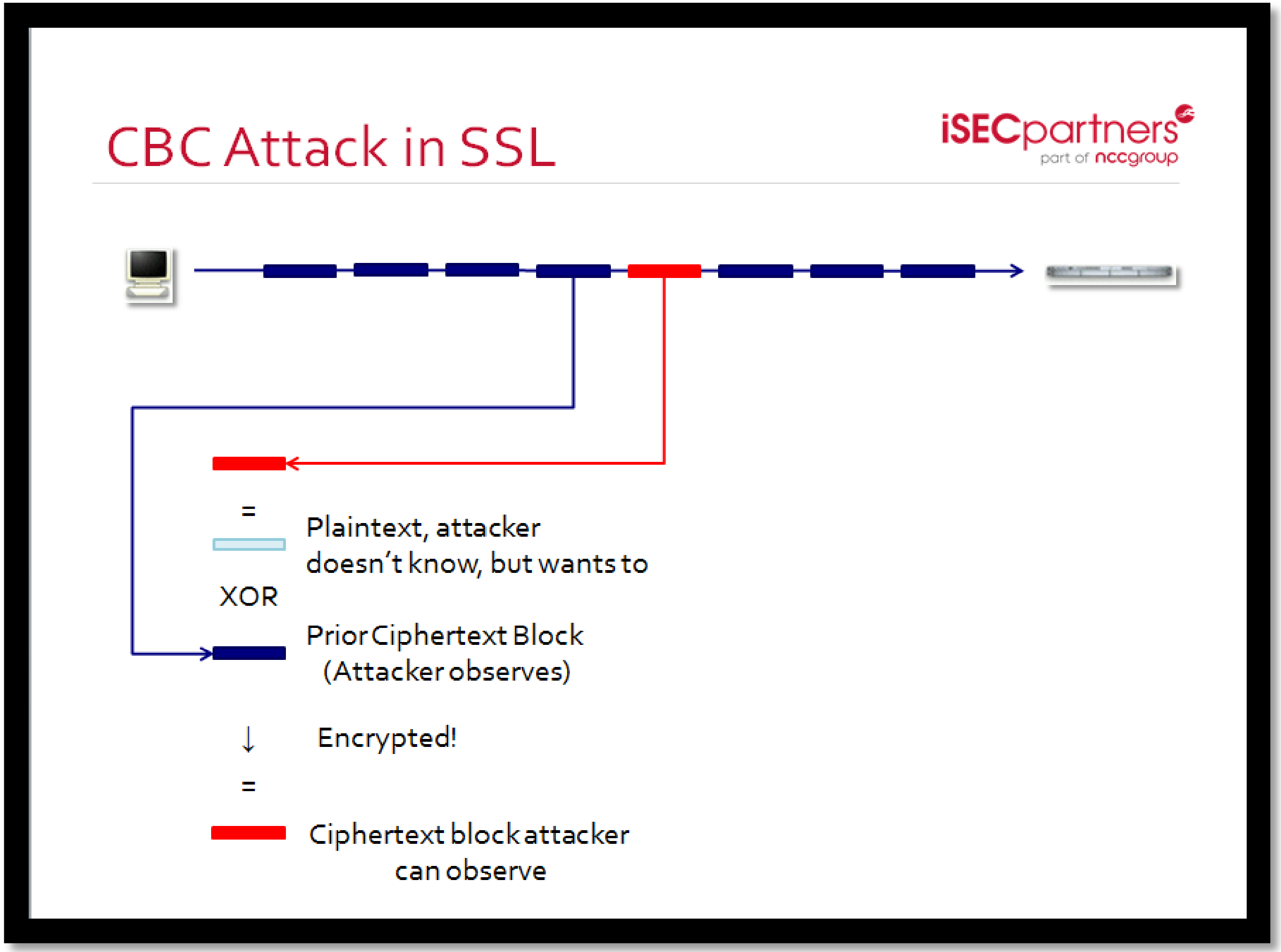


CBC Attack in SSL



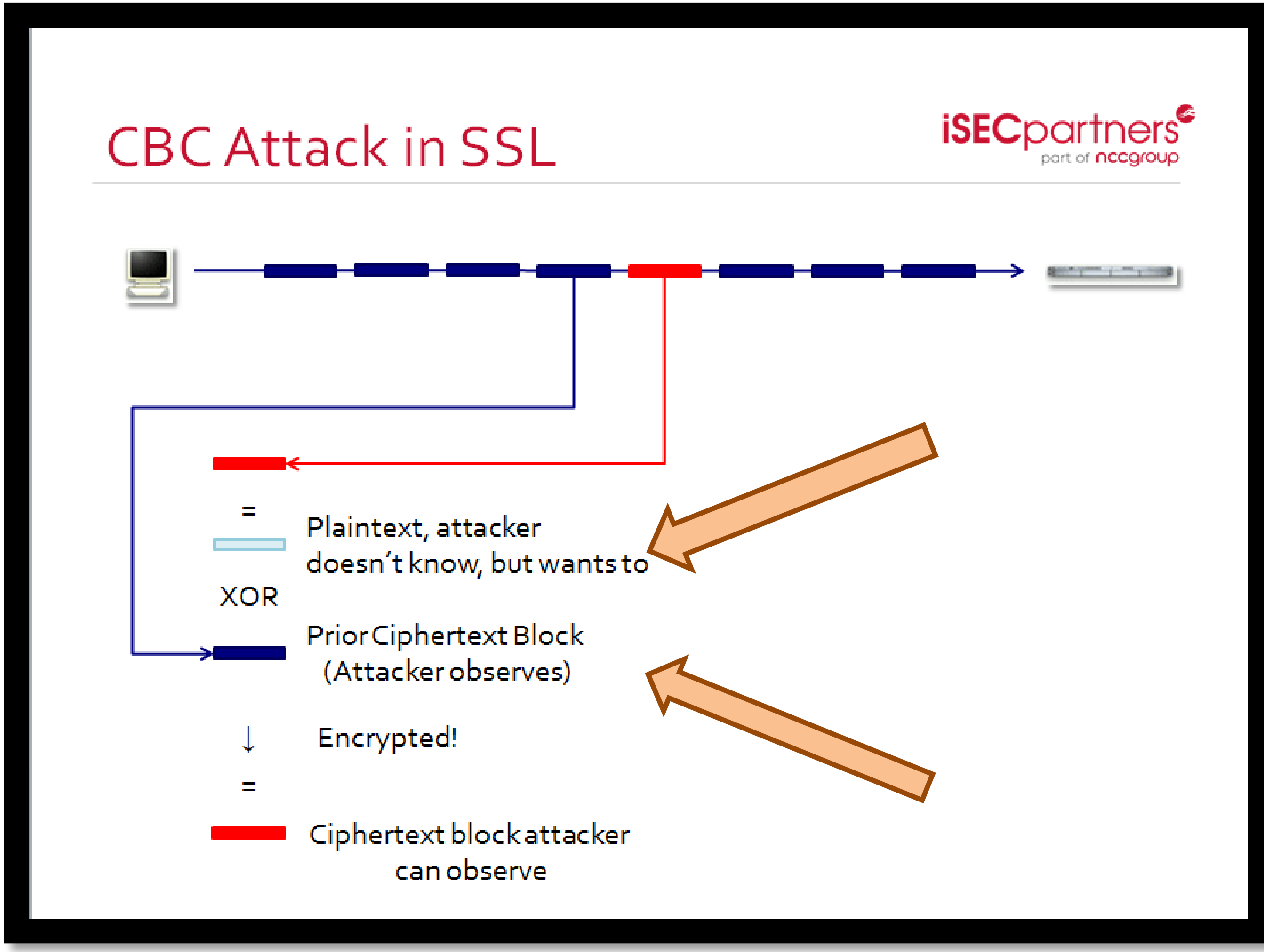
CBC Attack in SSL

- Remember This Slide?

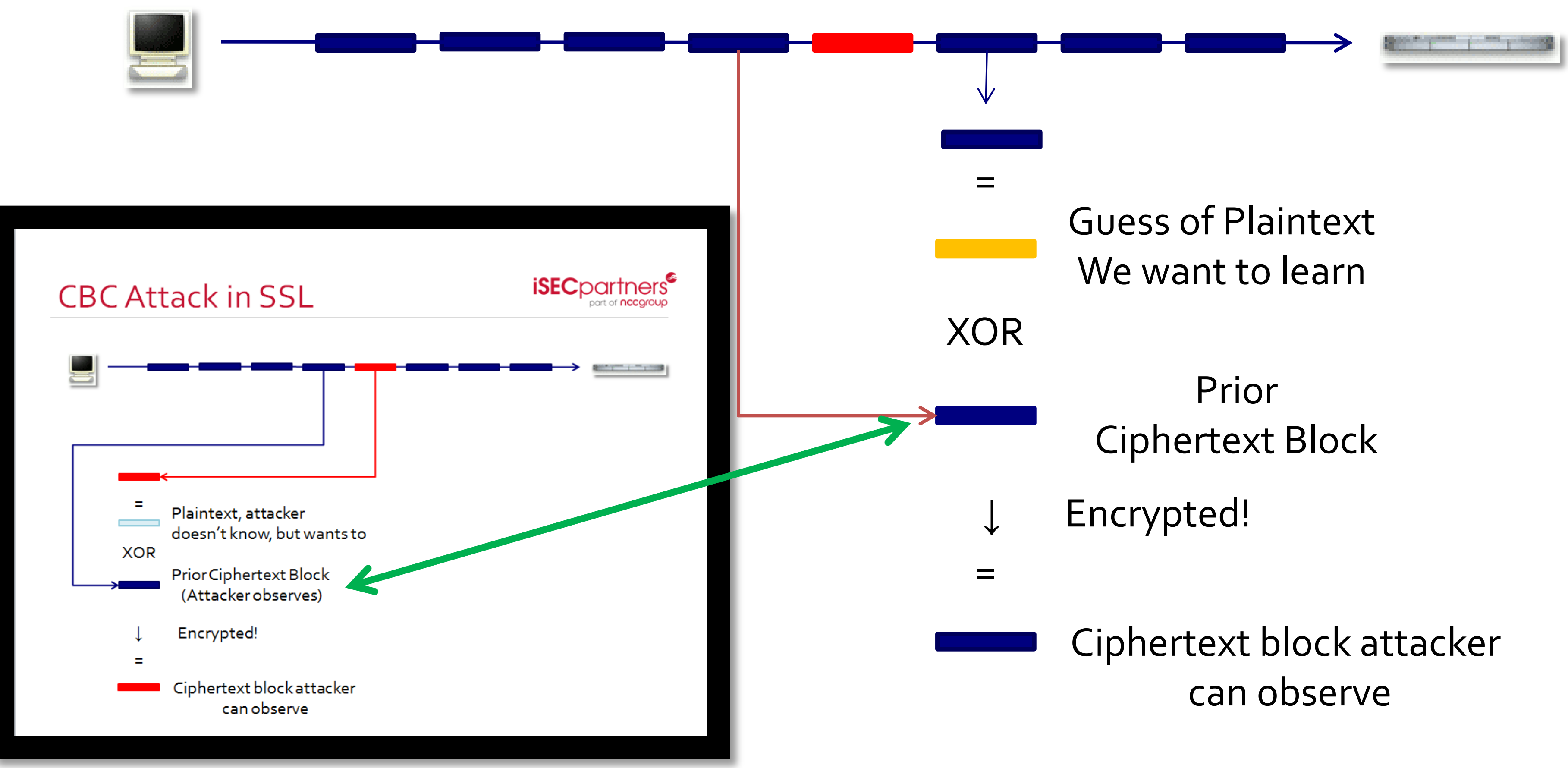


CBC Attack in SSL

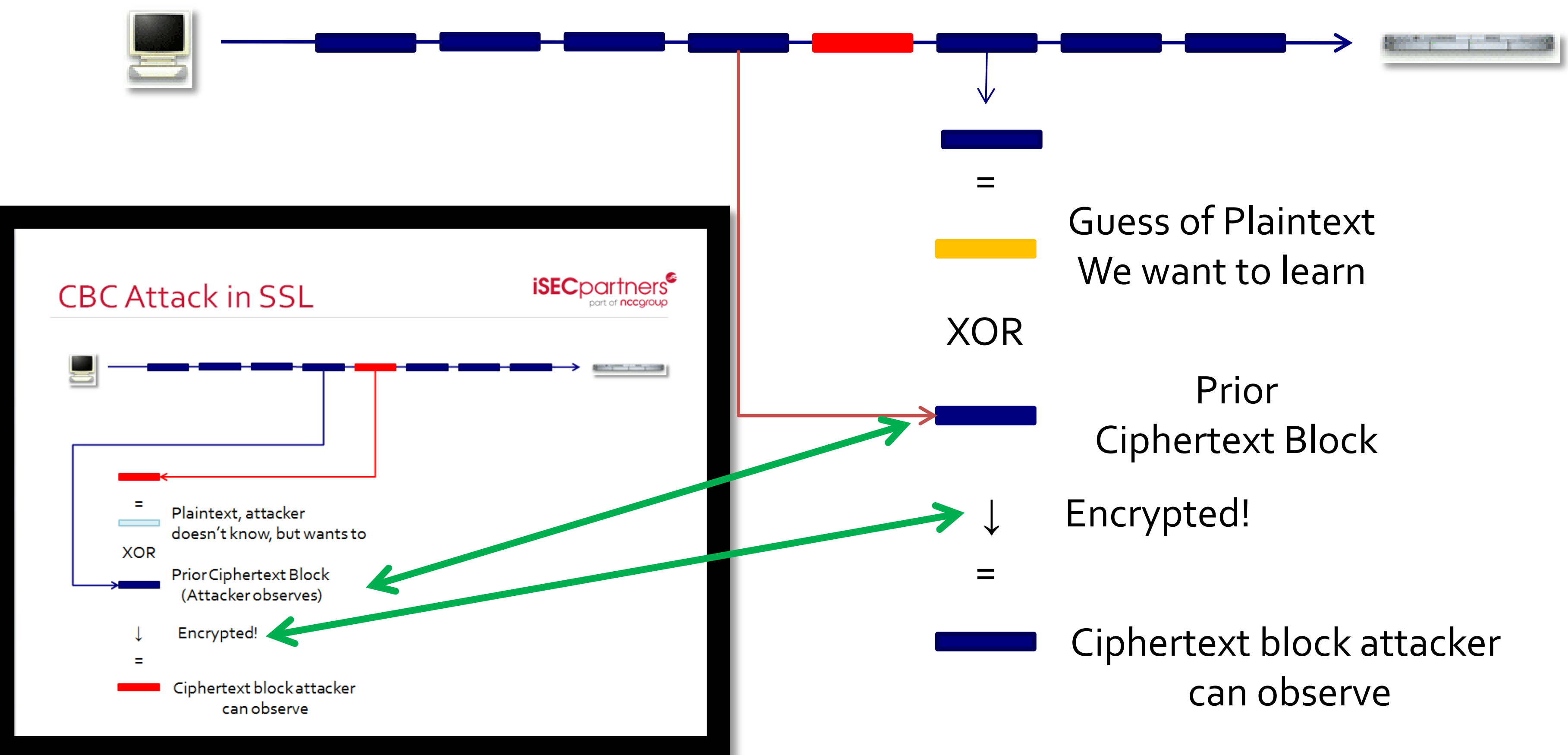
- Remember This Slide?



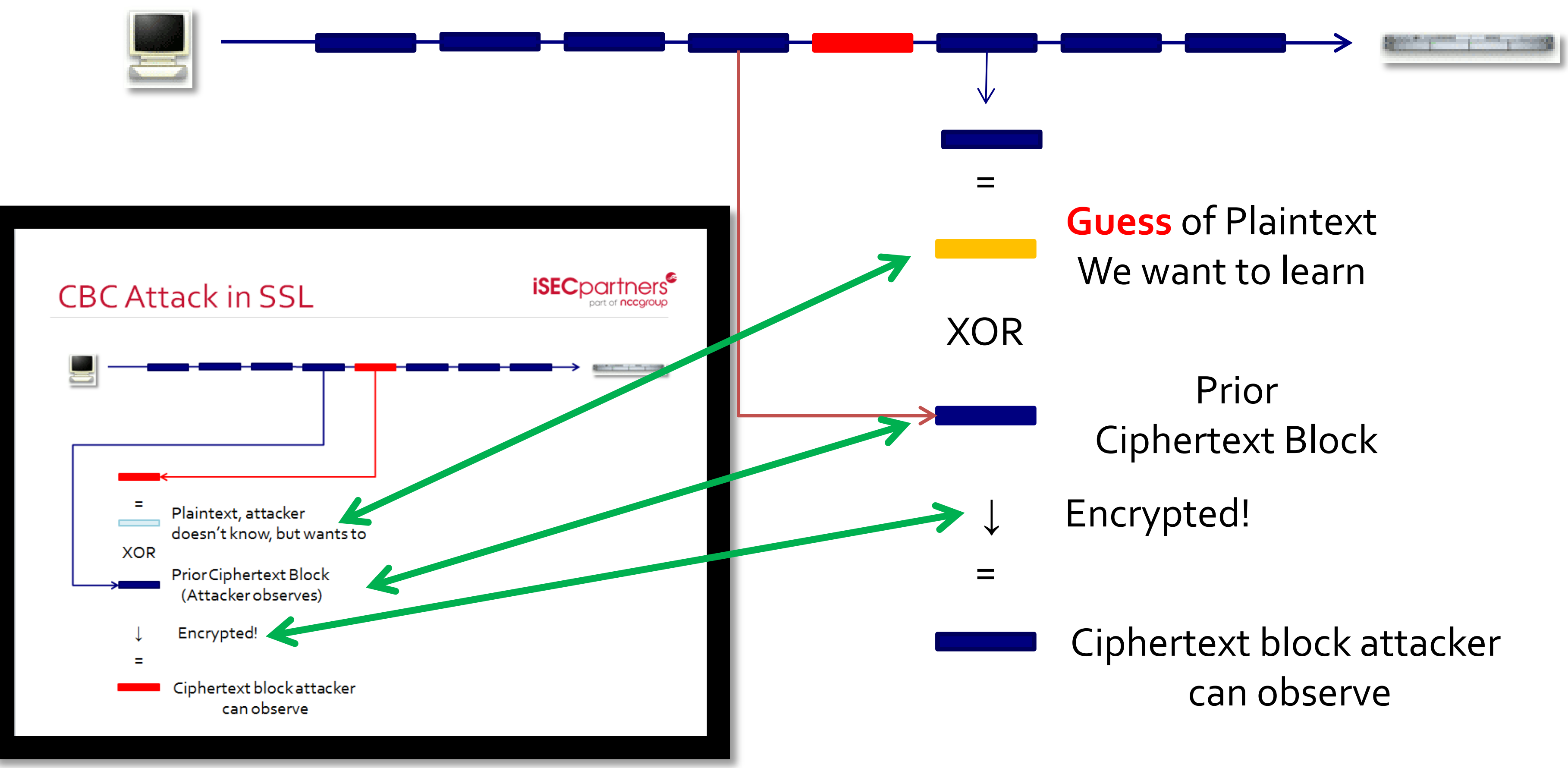
CBC Attack in SSL



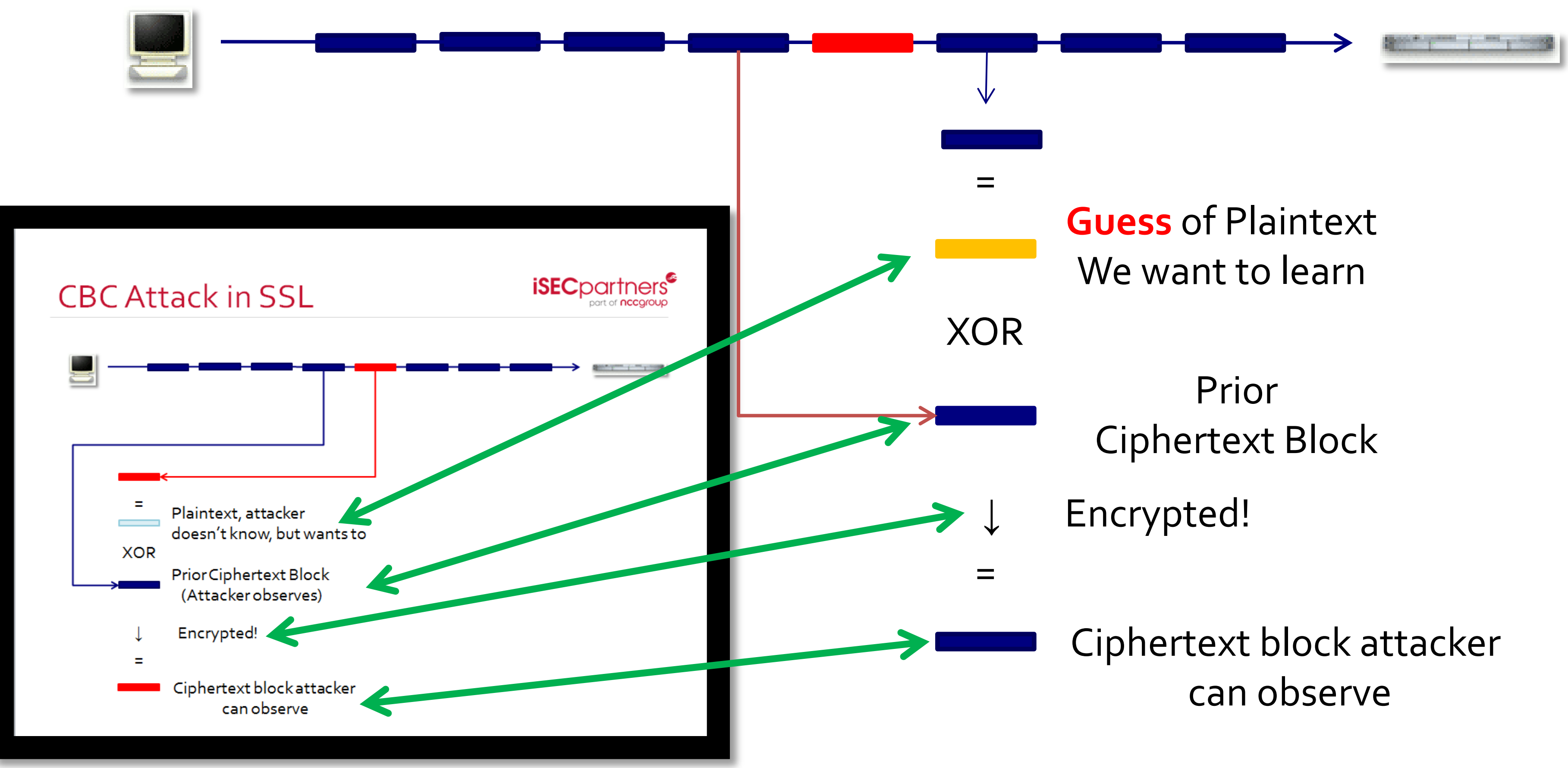
CBC Attack in SSL



CBC Attack in SSL

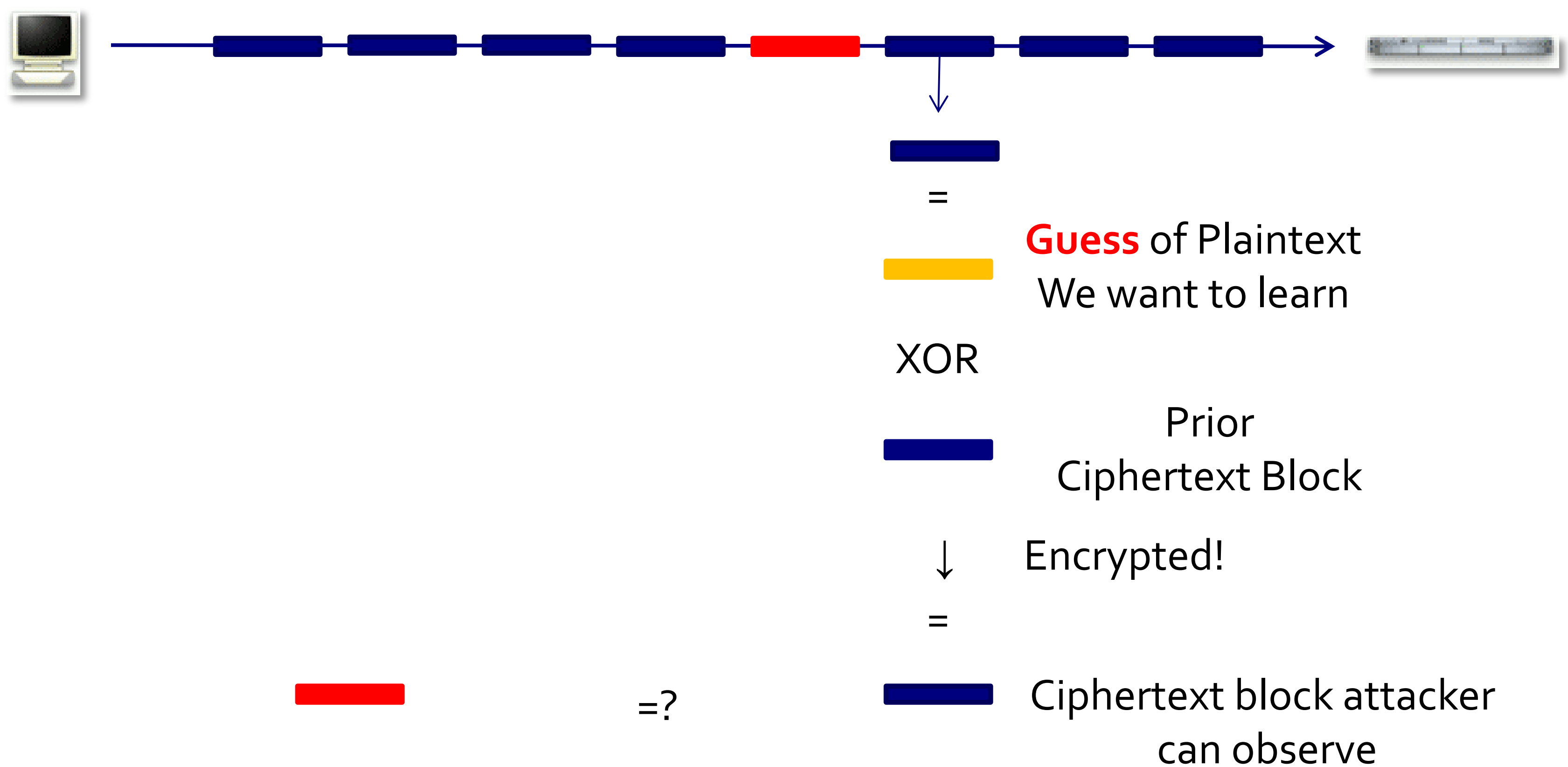


CBC Attack in SSL



If we guessed right, we will see an output that MATCHES a cipher text block we saw previously!

CBC Attack in SSL



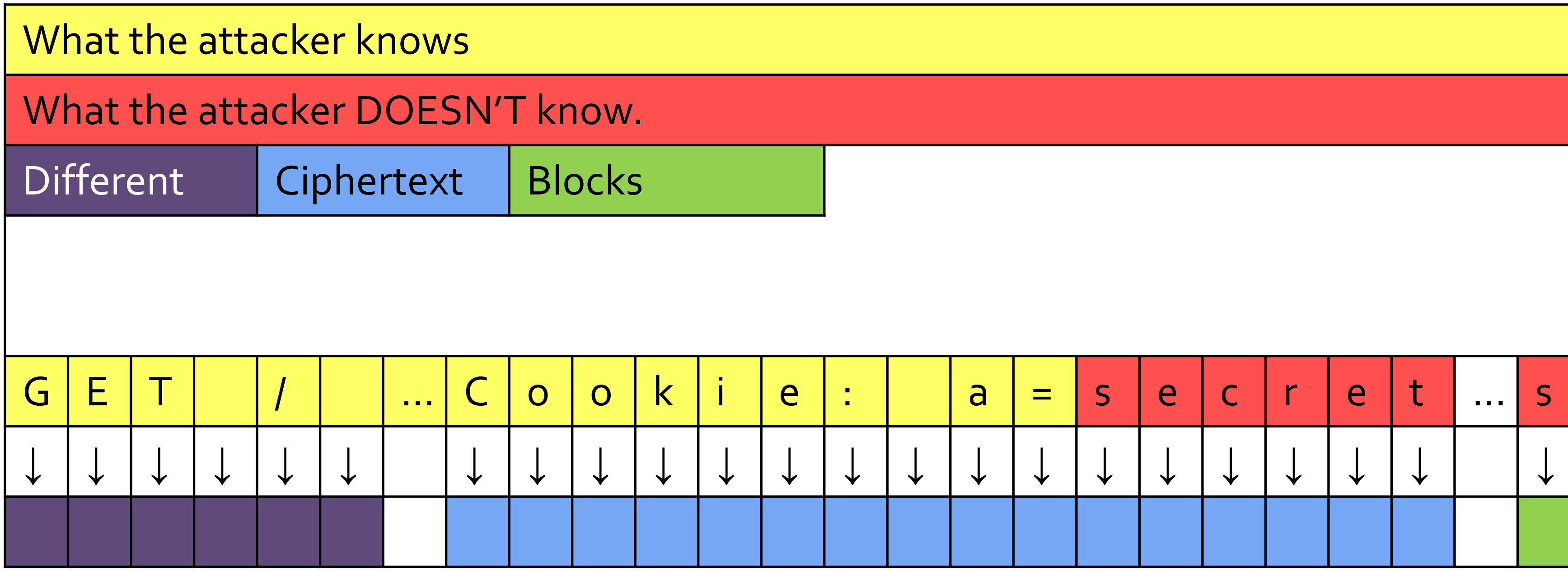
If we guessed right, we will see an output that MATCHES a cipher text block we saw previously!

HTTP Request – What Do I Know?

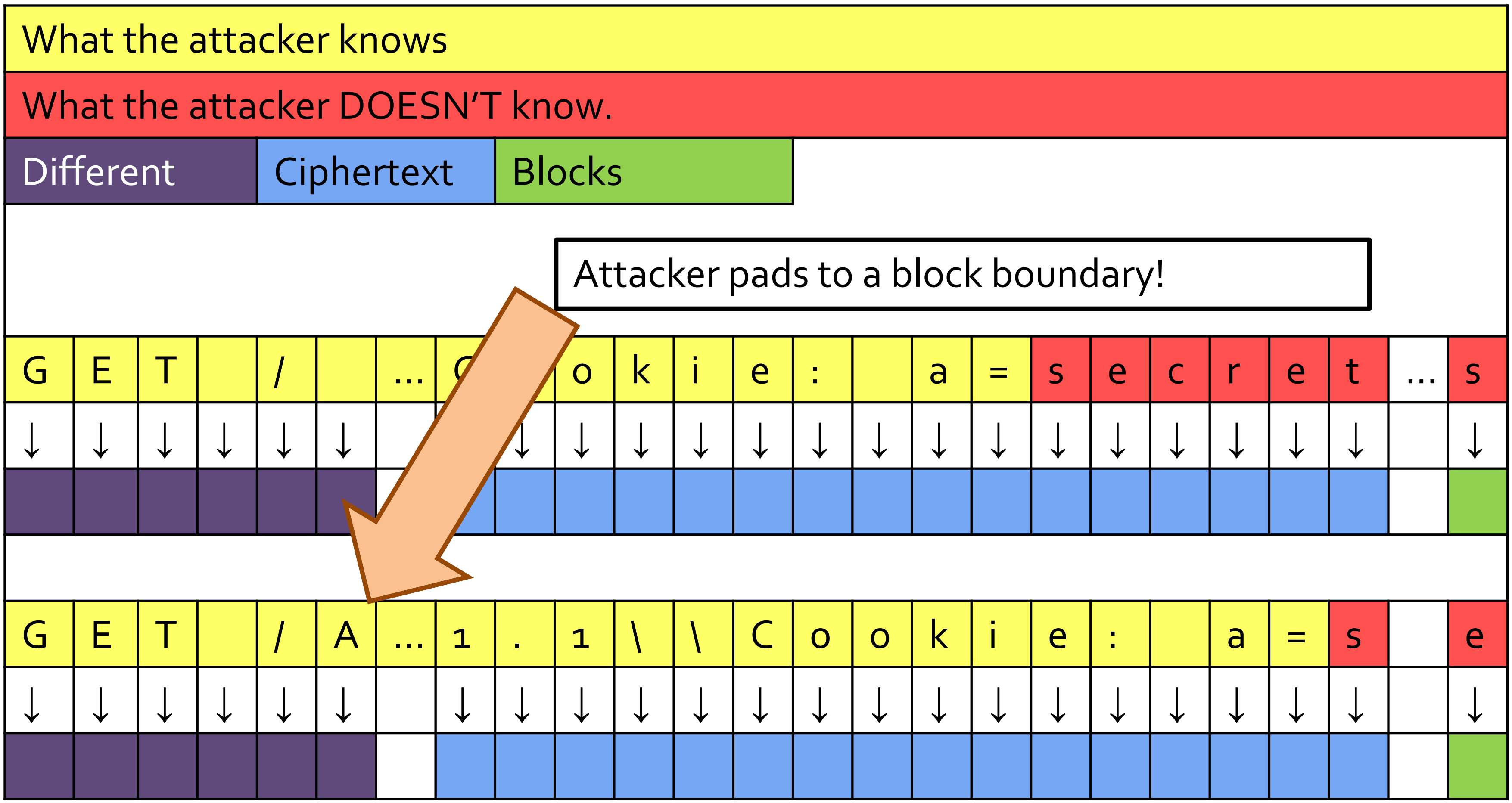
```
POST /login HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/20100101 Firefox/16.0
Cookie: a=secrets298fc1c149afbf4c8996fb924
```

I know everything but the cookie!

Chosen Boundary in a Slide



Chosen Boundary in a Slide



Put It Together: BEAST

- Attacks SSL 3.0 TLS 1.0 with CBC Cipher suites
- Steals Cookies
- Works on HTTPS-only sites
  - Sorry Paypal

BEAST – Feasibility

- Ability to eavesdrop on the network
- Force victim to visit attackers page
- Ability to inject plaintext in an active SSL/TLS session

BEAST – Counter Measures

- Upgrade browsers
- Enable TLS 1.1, preferably 1.2
- Use RC4

Interlude: Protocol Downgrades

- We've mentioned TLS 1.1 and 1.2
- They're great!

- There's a problem:

Interlude: Protocol Downgrades

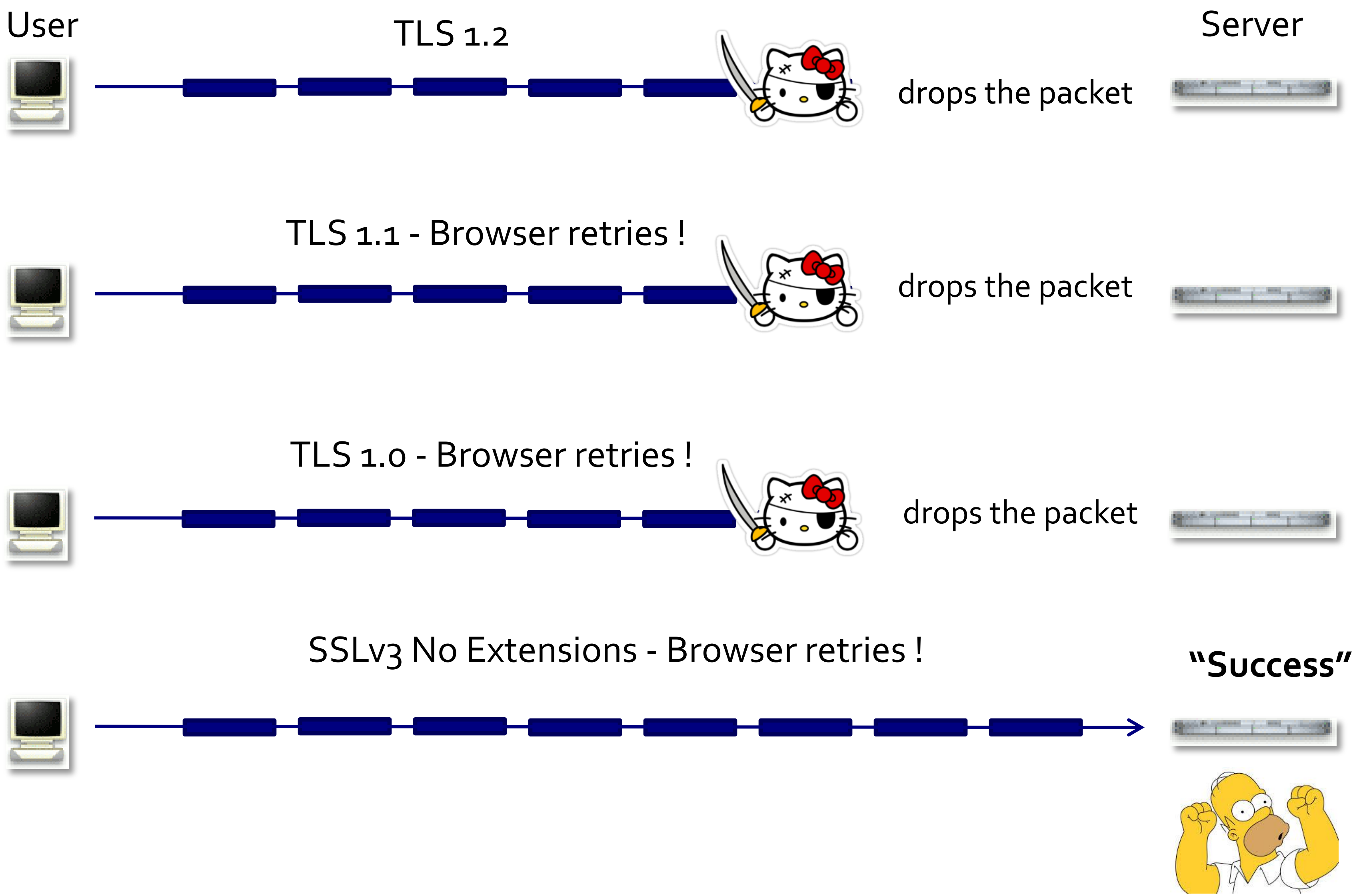
iSECpartners part of nccgroup

- We've mentioned TLS 1.1 and 1.2
- They're great!

- There's a problem:

They provide no security at all against an active attacker

Interlude: Protocol Downgrades



Interlude: Protocol Downgrades

Why Do Browsers Support Fallback?

- Networks Are Hostile to TLS 1.1+
  - Middleboxes don’t recognize it and choke
- Sites Can’t Speak TLS 1.1+
  - Sometimes an error (not so bad)
  - Sometimes they just hang (quite bad)

Interlude: Protocol Downgrades

- Until Browsers Remove Fallback to TLS 1/SSLv3 we cannot fully rely on TLS 1.1+
- Until sites stop breaking for TLS 1.1+ Browsers can't Remove the Fallback

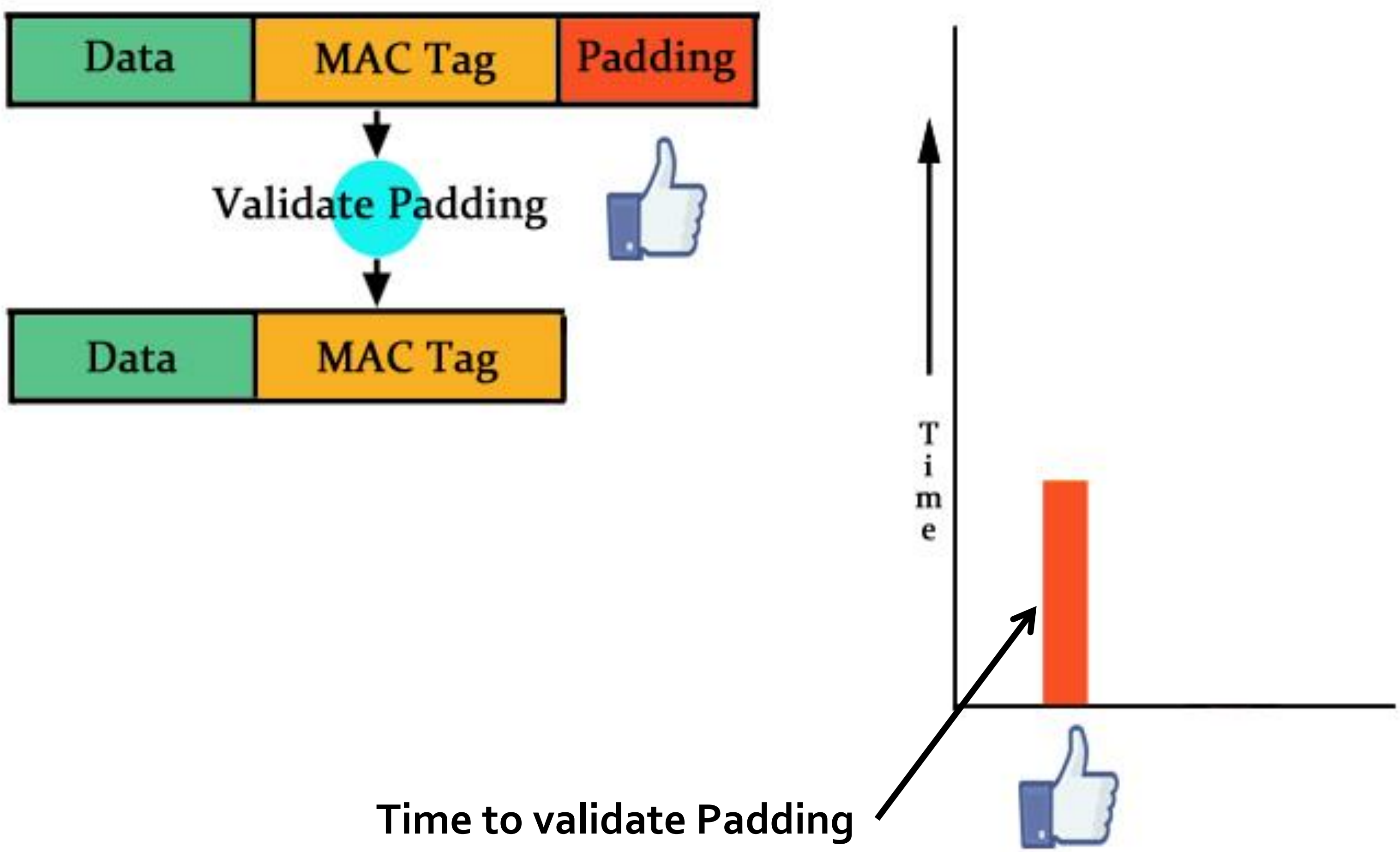
- Not to call anyone out.... But....
  - https://www.imperialviolet.org/2013/10/07/f5update.html

Lucky 13

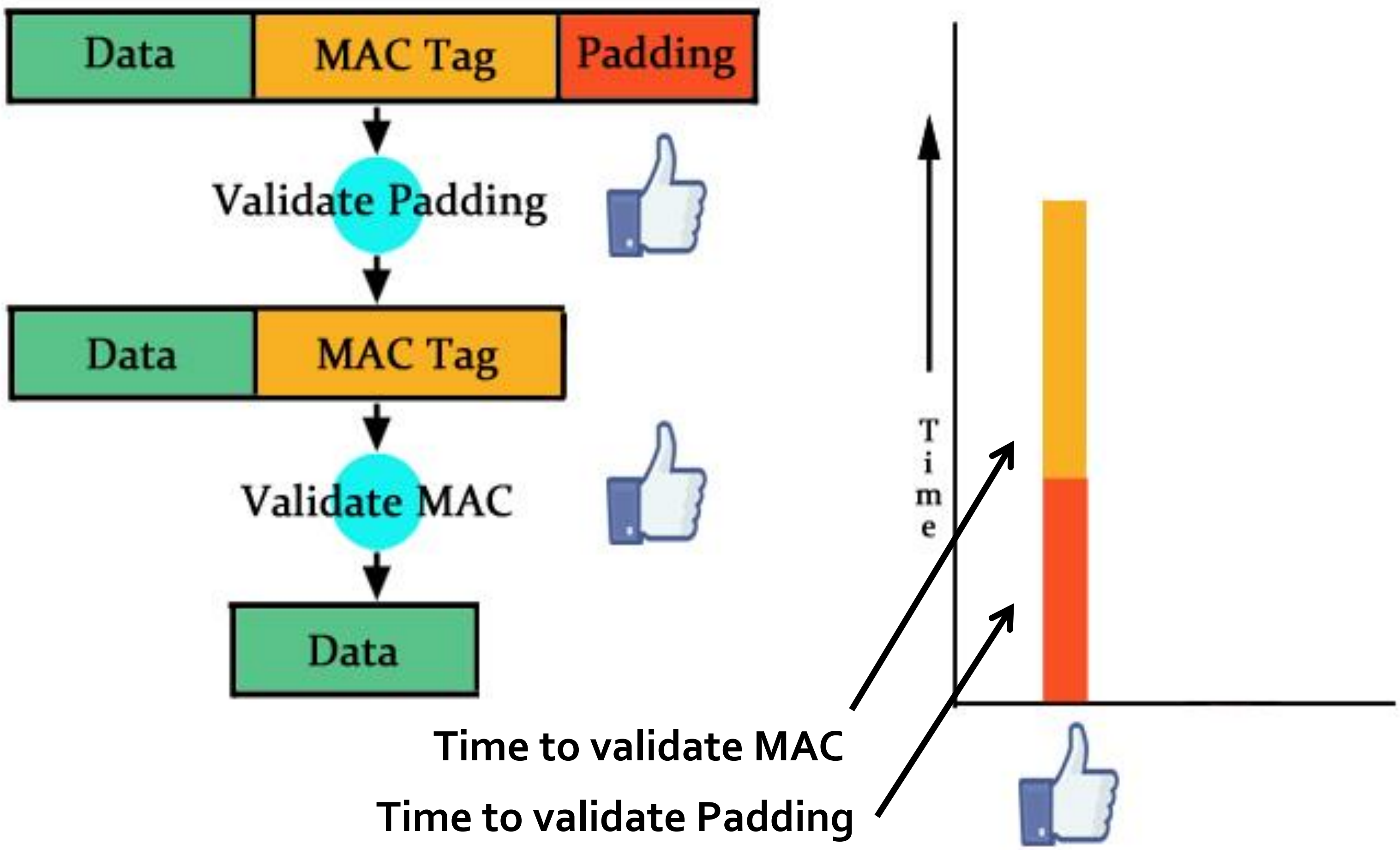
Lucky 13 - What is it ?

- Successor of Padding Oracle Attack
- Timing attack on CBC encryption mode
- 13 bytes of header information in TLS MAC calculation leaks timing information during decryption.

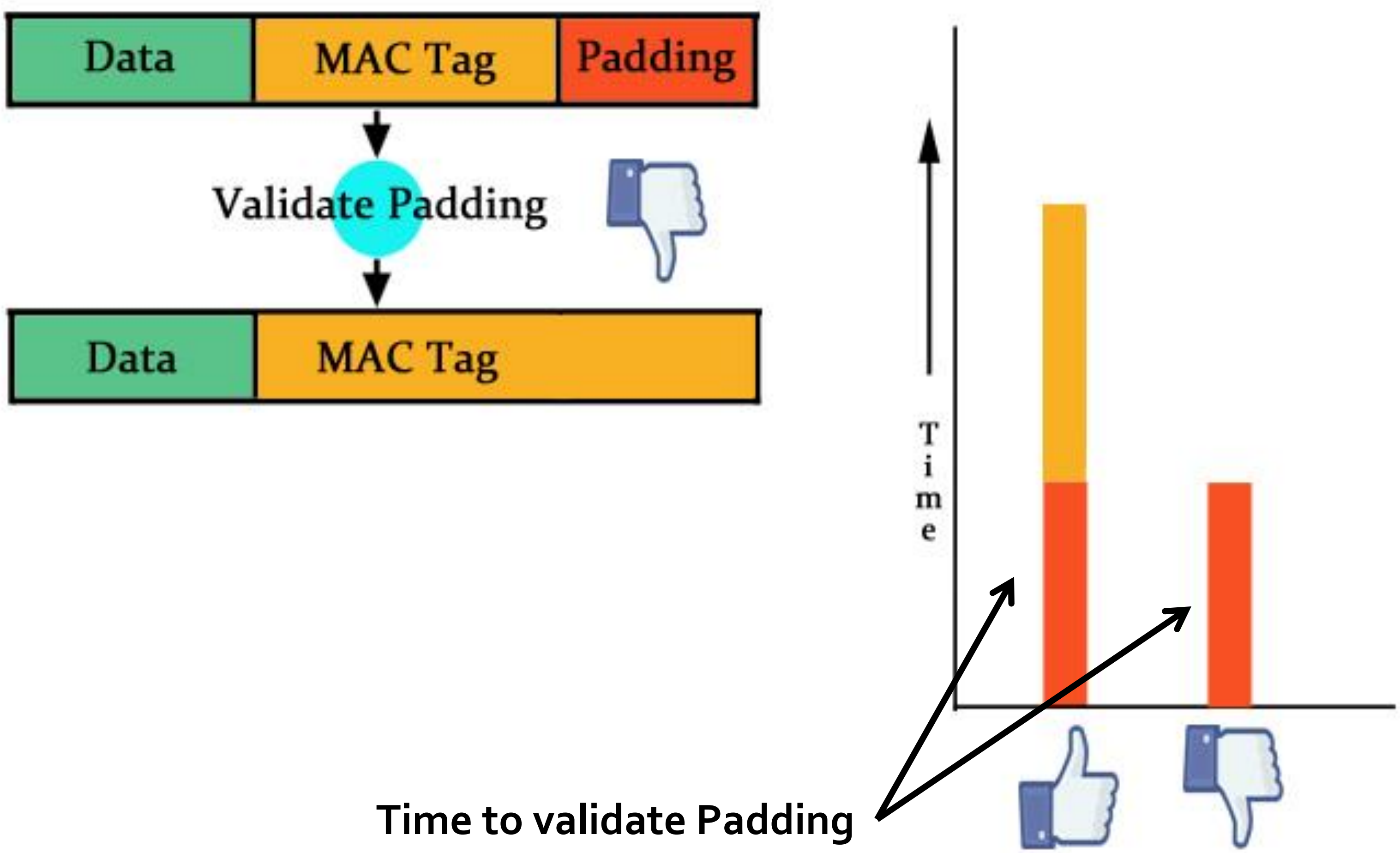
Lucky 13 – How it works



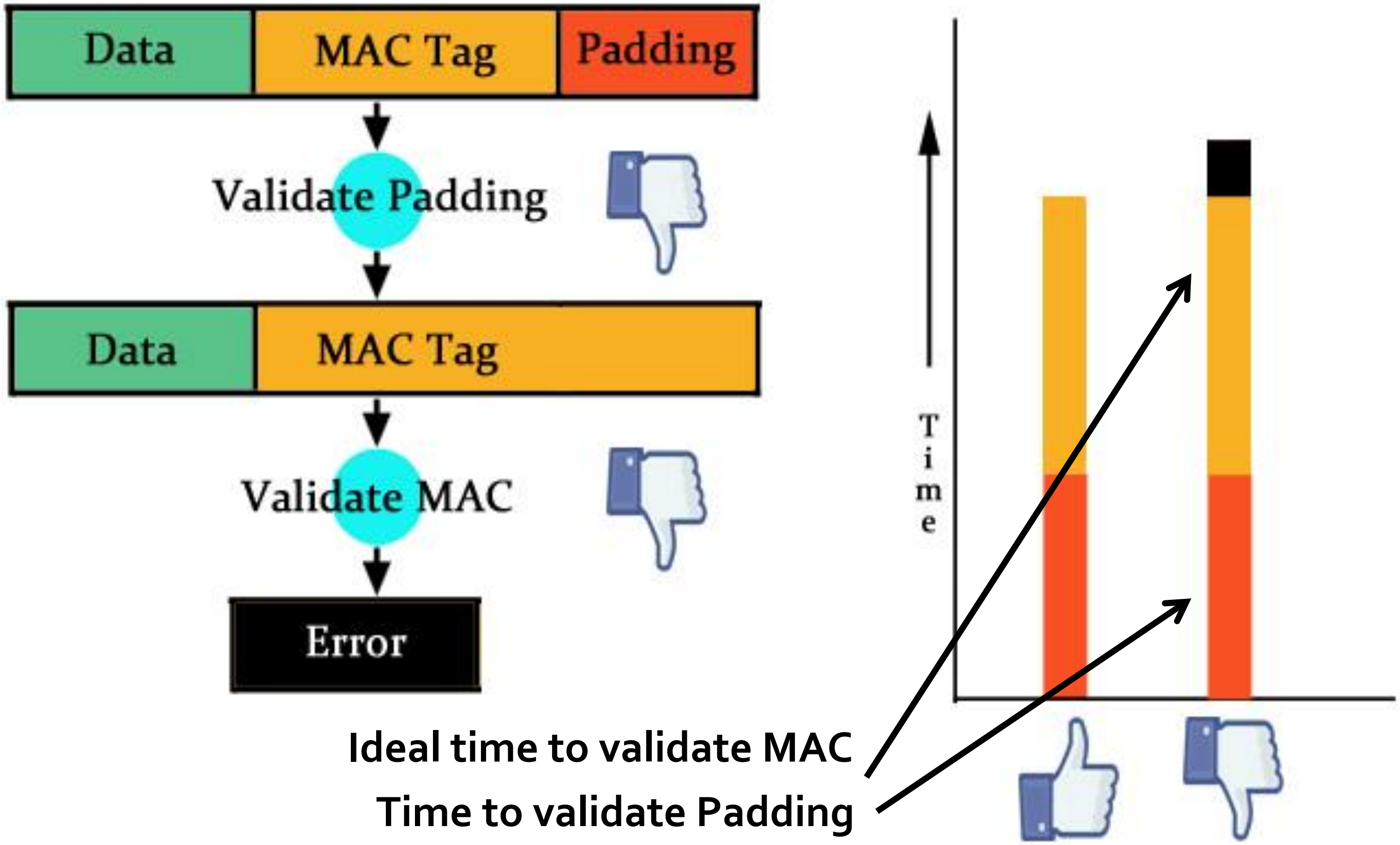
Lucky 13 – How it works



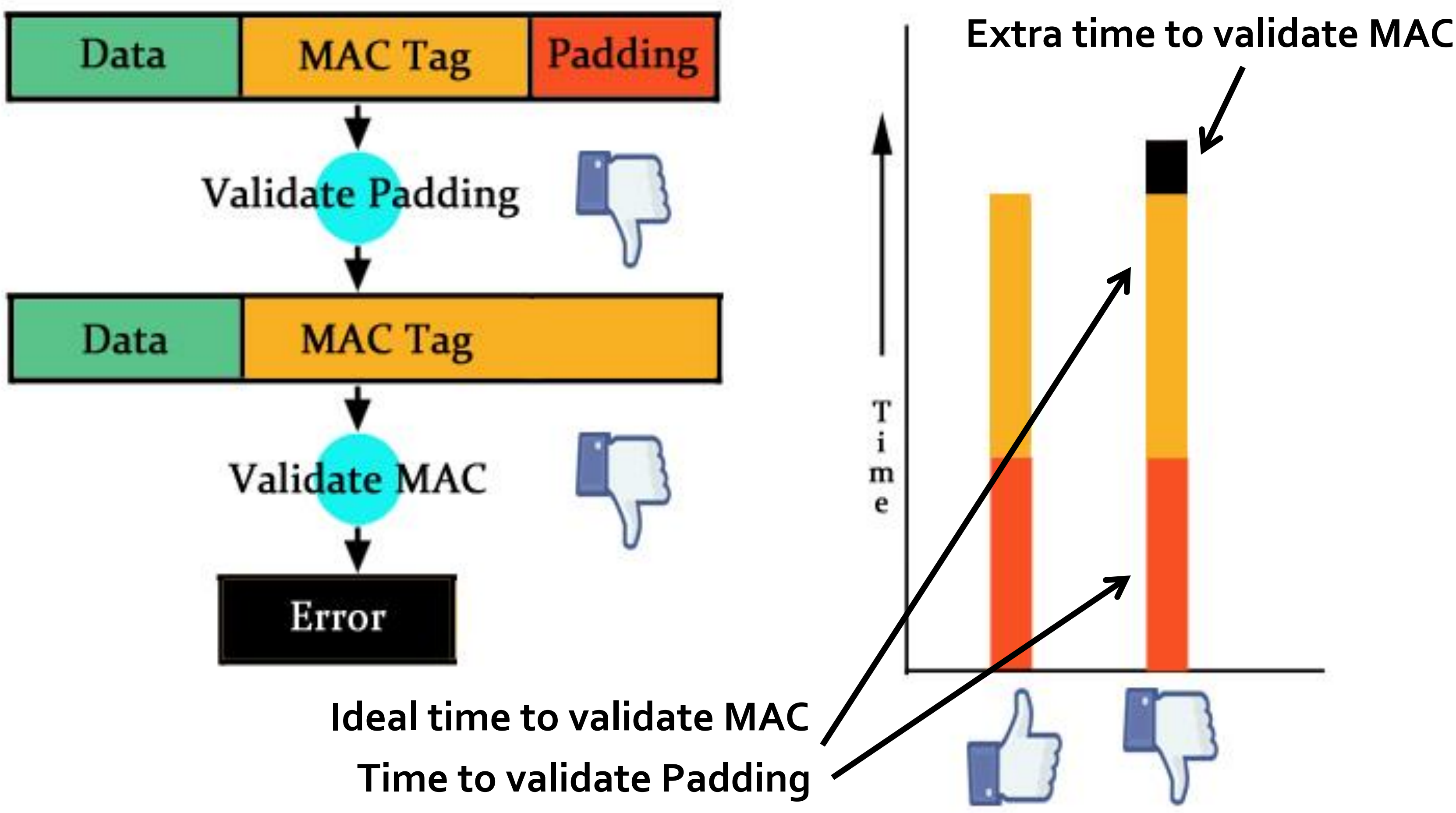
Lucky 13 – How it works



Lucky 13 – How it works



Lucky 13 – How it works



Lucky 13 – Feasibility

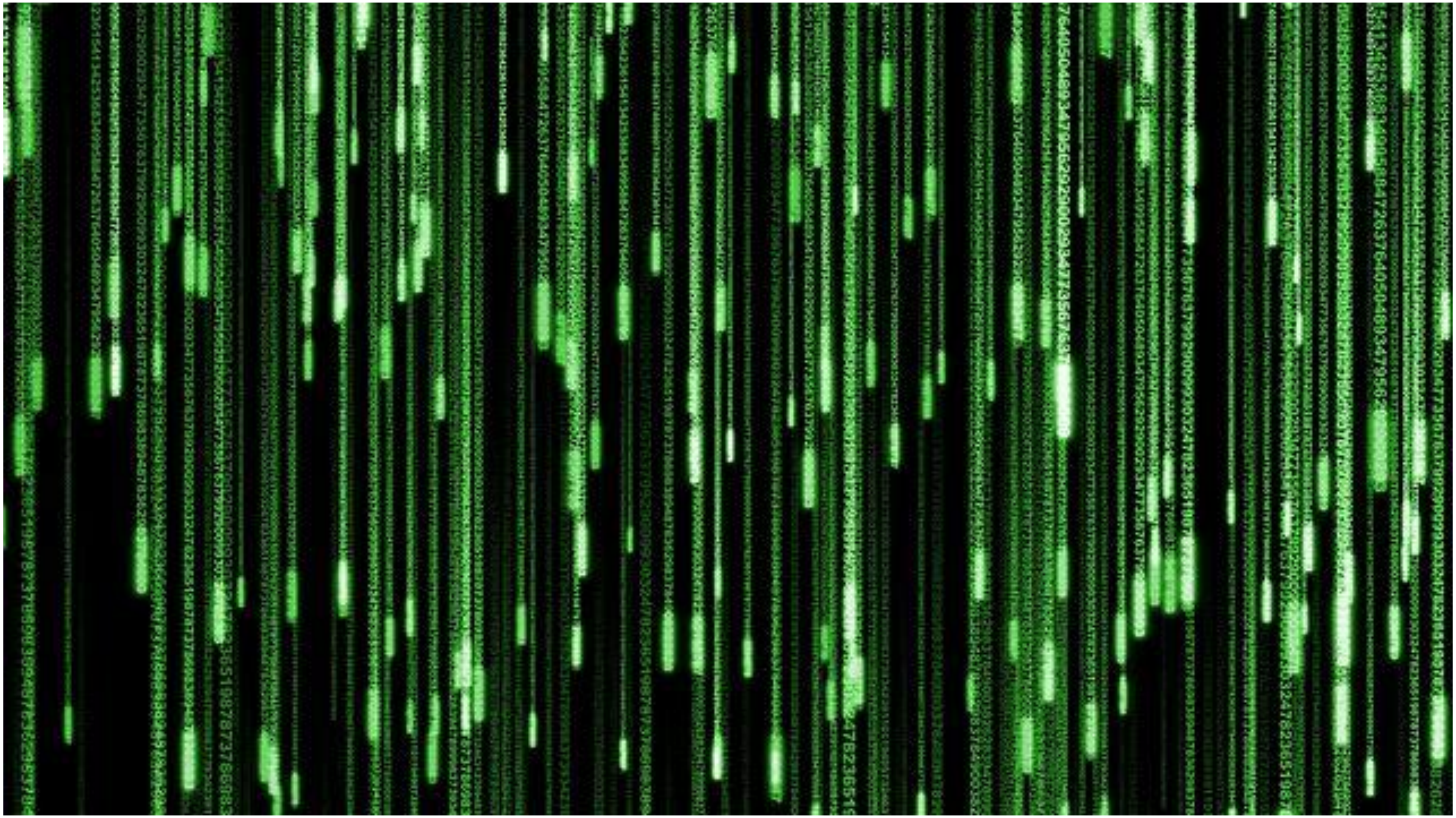
- Needs Man-in-the-middle
- CBC-mode encryption in versions of TLS are potentially vulnerable.
- Requires huge number of request
- Requires no Network jitter

Lucky 13 – Counter Measures

iSECpartners part of nccgroup

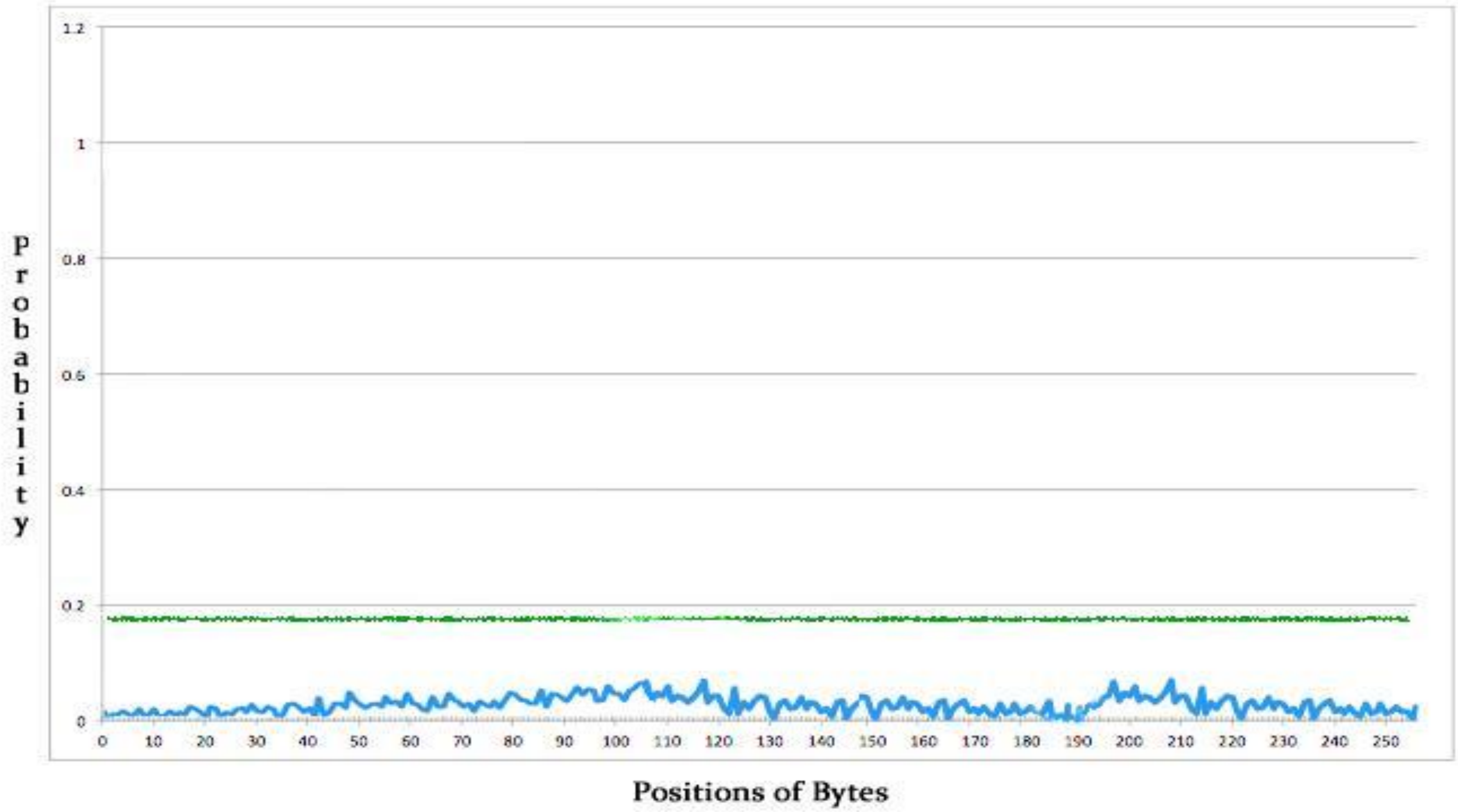
- Uniform processing time to decrypt ciphertexts
- Add random timing delays to the decryption for any timing attack
- Using stream cipher like RC4
- Using an authenticated encryption algorithm, such as AES-GCM

RC4



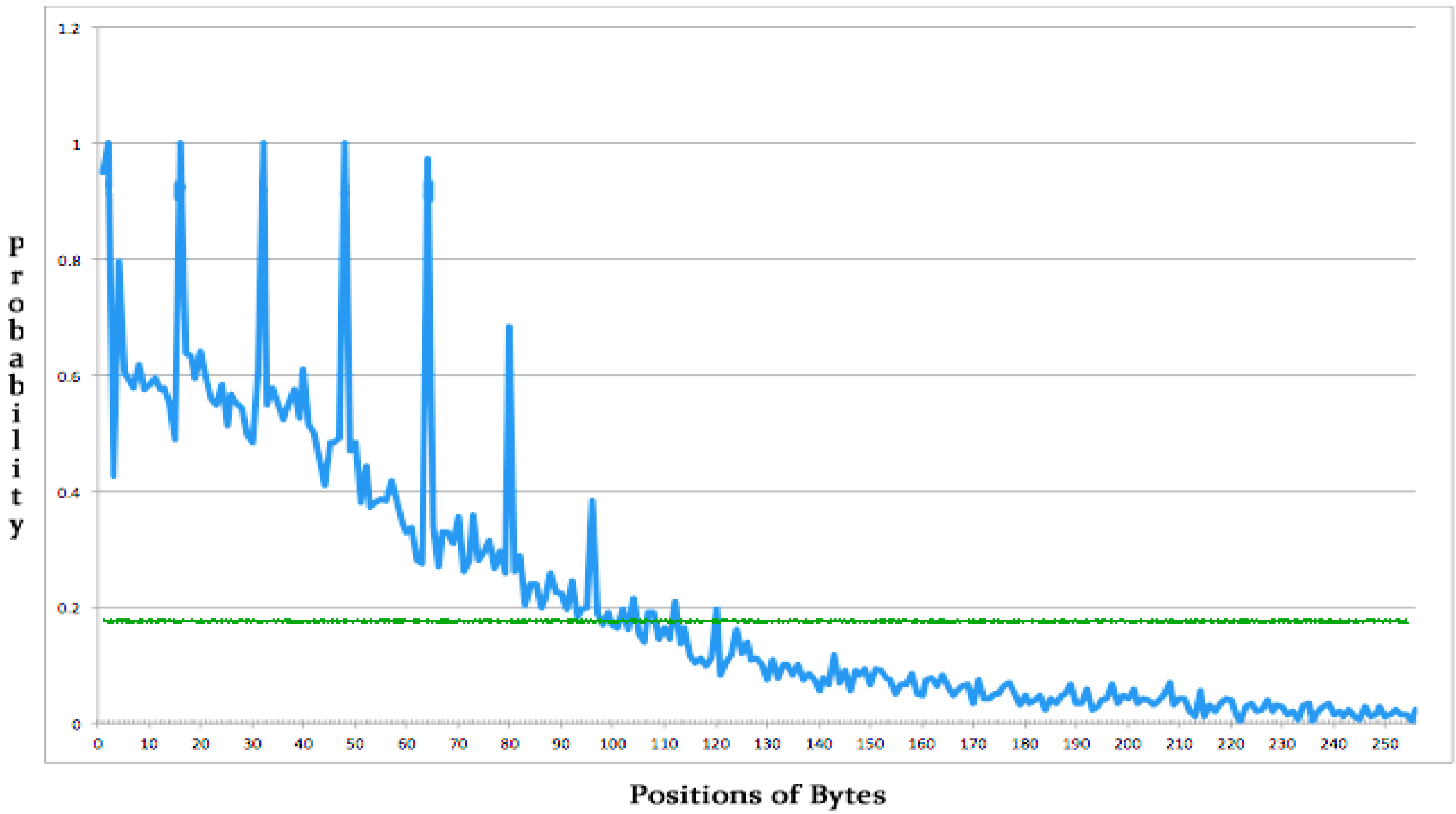
RC4 Biases - What is it ?

iSECpartners part of nccgroup



RC4 Biases - What is it ?

iSECpartners part of nccgroup



RC4 Biases – Feasibility

- Force victim to renegotiate.
- This attack will require over **4 billion** SSL connections or re-negotiations for an individual HTTP session.

RC4 Biases – Counter Measures

- Researchers still working on finding mitigations of this issue.

- Temporary mitigations
  - Throttle client initiated re-negotiations and connections from individual IP addresses
  - If possible use block ciphers with mitigations of timing and CBC mode encryption attack mitigated

Compression

How compression works?

- DEFLATE compression mechanism

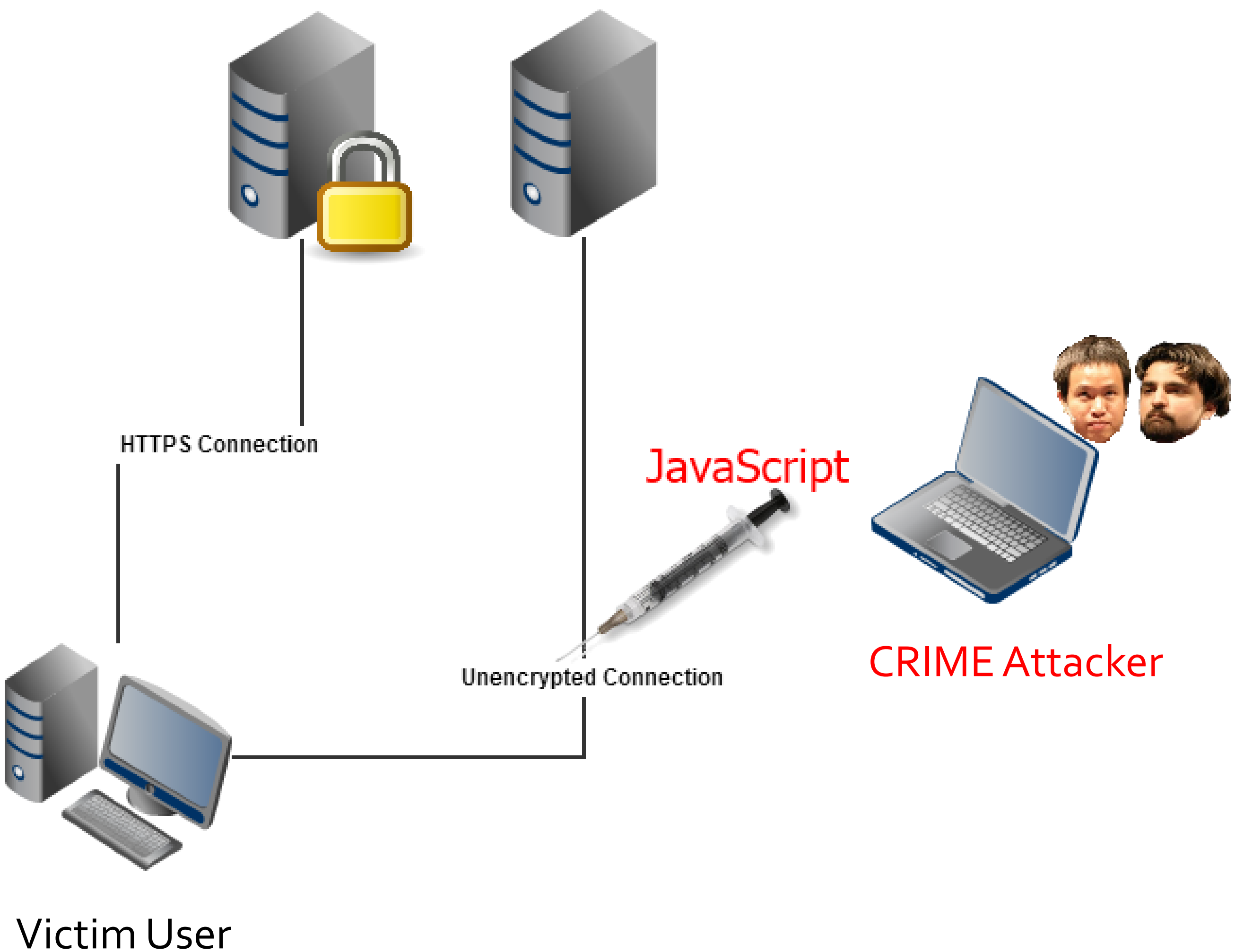
Etiam euismod. Fusce facilisis lacinia dui. Sus
pendisse potenti. In mi erat, cursus id, nonumm
y sed, ullamcorper eget, sapien. Praesent preti
um, magna in eleifend egestas, pede pede pretiu
m lorem, quis consectetuer tortor sapien facili
sis magna.

CRIME

CRIME - What is it ?

- **C**ompression **R**atio **I**nfo-leak **M**ade **E**asy
- Chosen plaintext attack on HTTP request
- Uses size information in TLS compression to recover plaintext cookies

CRIME – How it works



CRIME – How it works

iSECpartners part of nccgroup

```
GET /evil_request_path HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0)
Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

Attacker doesn't control entire request, but can see its cipher text on the wire

CRIME – How it works

iSECpartners part of nccgroup

```
GET /evil_request_path HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

Attacker fully controls request path

CRIME – How it works

```
GET /evil_request_path HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0)
Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

Attacker does not see, but can infer these values

CRIME – How it works

iSECpartners part of nccgroup

```
GET /evil_request_path HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0)
Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

Attacker cannot see/control, wants to steal

CRIME – How it works

iSECpartners part of nccgroup

```
GET /sessionid=a HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0)
Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

=> Compressed Length = 12,494 bytes – Not a match

```
GET /sessionid=d HTTP/1.1
Host: bank.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0)
Gecko/20100101 Firefox/16.0
Cookie: sessionid=d3b0c44298fc1c149afbf4c8996fb924
```

=> Compressed Length = 12,493 bytes – **Possible match**

CRIME – Feasibility

- The attacker can intercept the victim's network traffic.
- Victim authenticates to a website over HTTPS and negotiates TLS Compression with the server.
- Victim accesses a non-HTTPS website.
- Browser supporting TLS Compression

CRIME – Counter Measures

- Disabling TLS compression on both Browser and Server side.
- Updated Browser versions:
  - Chrome: 21.0.1180.89 and above
  - Firefox: 15.0.1 and above
  - Opera: 12.01 and above
  - Safari: 5.1.7 and above
- Apache 2.2 using mod_SSL:

  SSLCompression flag is set to "*SSLCompression off*"
- Apache using mod_gnutls :

  GnuTLSPriorities flag = "*!COMP-DEFLATE*"

BREACH

BREACH – What is it ?

- **B**rowser **R**econnaissance and **E**xfiltration via **A**daptive **C**ompression of **H**ypertext
- Chosen plaintext attack on HTTP response
- Uses difference of response size information in due to varying sizes of HTTP compression to recover plaintext secret information
- Resurrection of CRIME

BREACH – How it works

```
<form target="https://example.com:443/
products/catalogue.aspx?id=12345&user=username" >
...
<td nowrap id="tdErrLgf">
<a href="logoff.aspx?CSRFtoken=4bd634cda846fd7cb4cb00
31ba249ca2">Log Off</a></td>
```

Attacker can control this value

BREACH – How it works

```
<form target="https://example.com:443/
products/catalogue.aspx?id=12345&user=username" >
...
<td nowrap id="tdErrLgf">
<a href="logoff.aspx?CSRFtoken=4bd634cda846fd7cb4cb00
31ba249ca2">Log Off</a></td>
```

Attacker cannot control this parameter, want to steal it

BREACH – How it works

```
GET /product/?id=12345&user=CSRFtoken=a HTTP /1.1
Host: example.com
```

```
<form target="https://example.com:443/
products/catalogue.aspx?id=12345&user=CSRFtoken=a" >
...
<td nowrap id="tdErrLgf">
<a href="logoff.aspx?CSRFtoken=4bd634cda846fd7cb4cb00
31ba249ca2">Log Off</a></td>
```

```
Size of response < Previous size = Match
Size of response >= Previous size = Mismatch
```

BREACH – Feasibility

- The application supports **HTTP compression**.
- The response should **reflect** back user's input.
- The response should have some **sensitive/ secret** information embedded in the body.

BREACH – Counter Measures

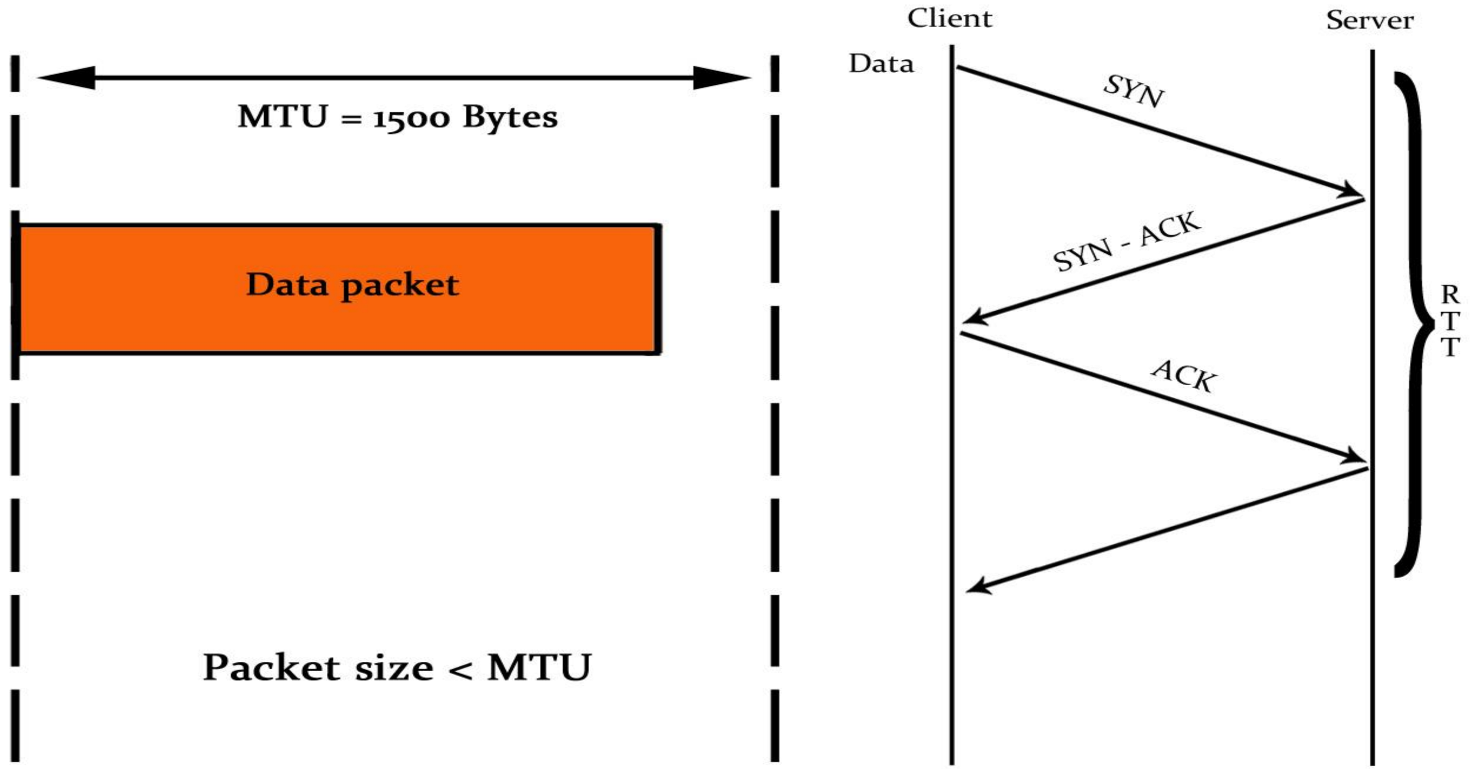
- Mask the secret:
  - new secret = random || (random $\oplus$ previous secret)
- Enable anti-automation techniques
- Monitor your traffic
- Separate secrets from user input
- ~~Disable HTTP compression~~

TIME

TIME - What is it ?

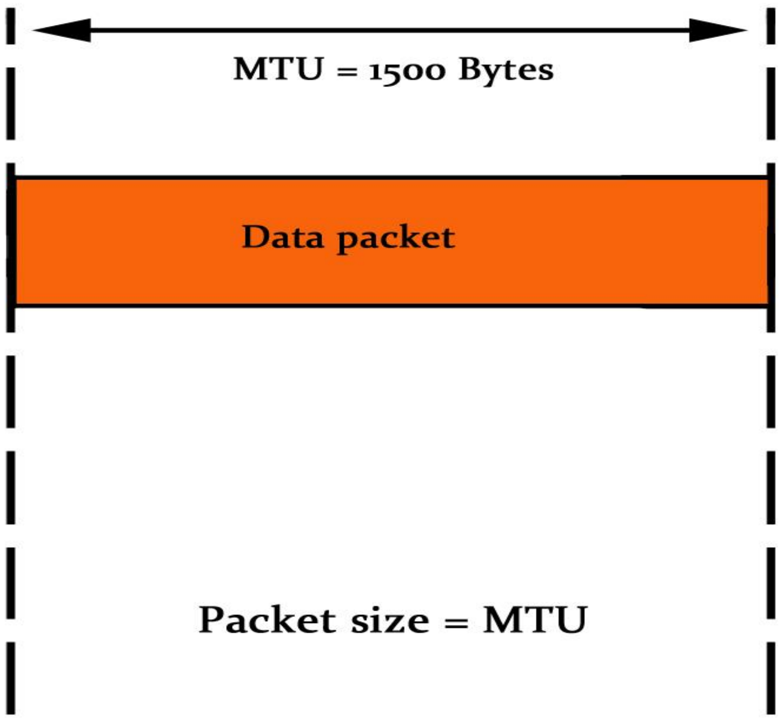
- **T**iming **I**nfo-leak **M**ade **E**asy
- Chosen plaintext attack on HTTP response
- Uses difference of response time information in due to varying sizes of HTTP compression to recover plaintext secret information
- Resurrection of CRIME

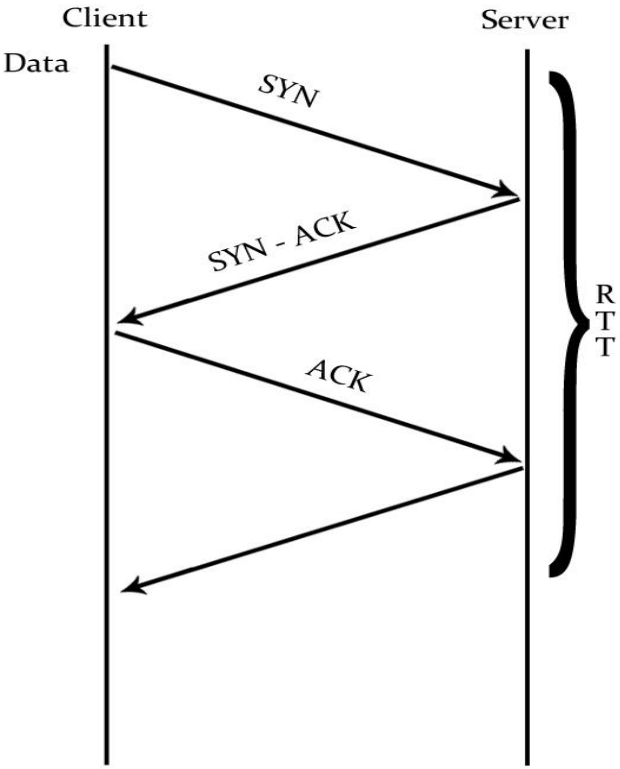
TIME – How it works



MTU : Maximum Transmission Unit
RTT: Rount Trip Time

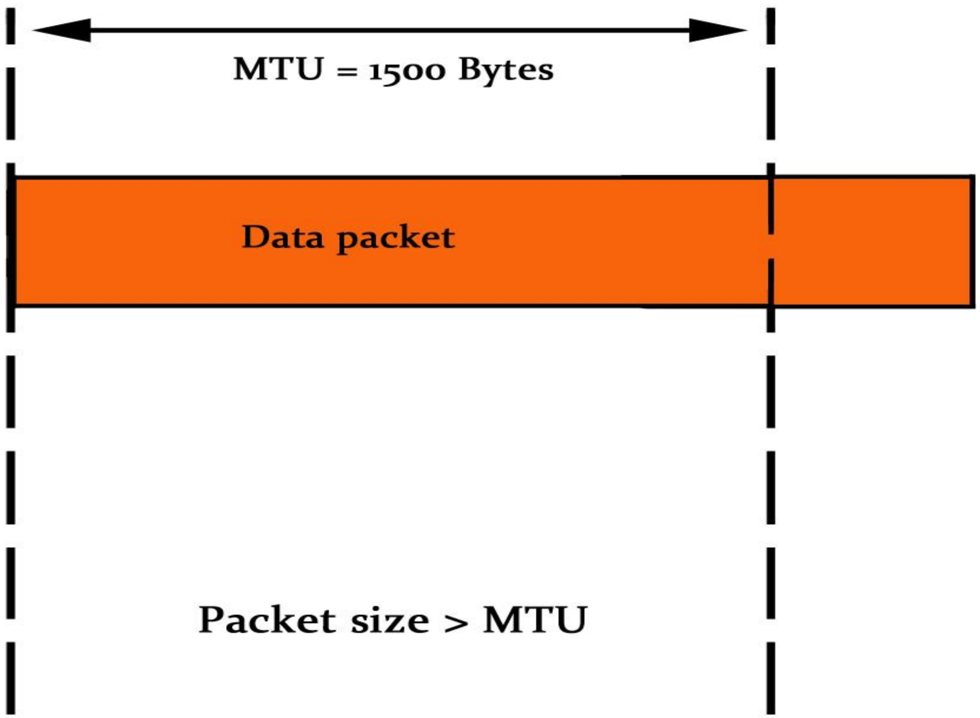
TIME – How it works





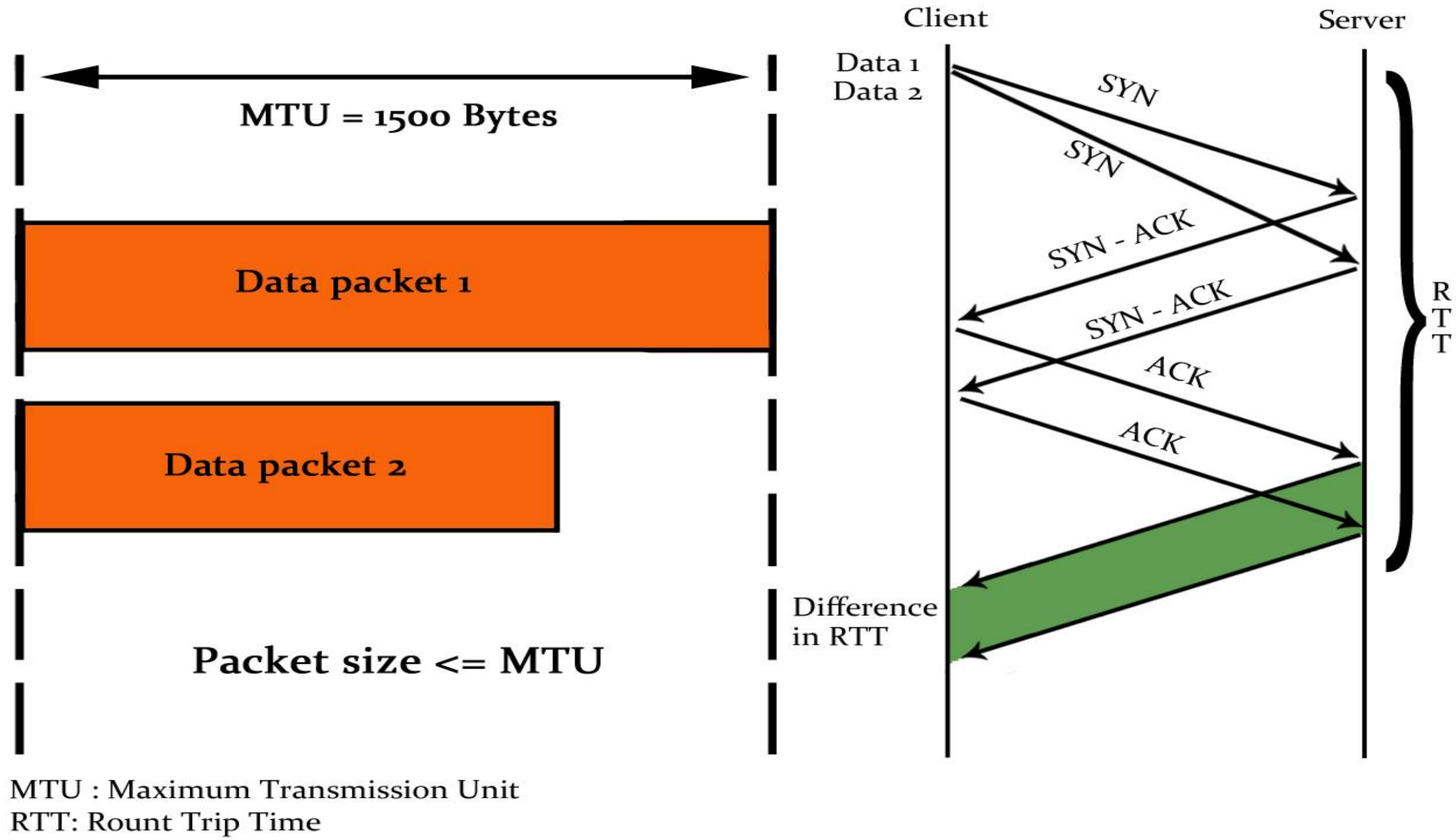
MTU : Maximum Transmission Unit
RTT: Rount Trip Time

TIME – How it works



MTU : Maximum Transmission Unit
RTT: Rount Trip Time

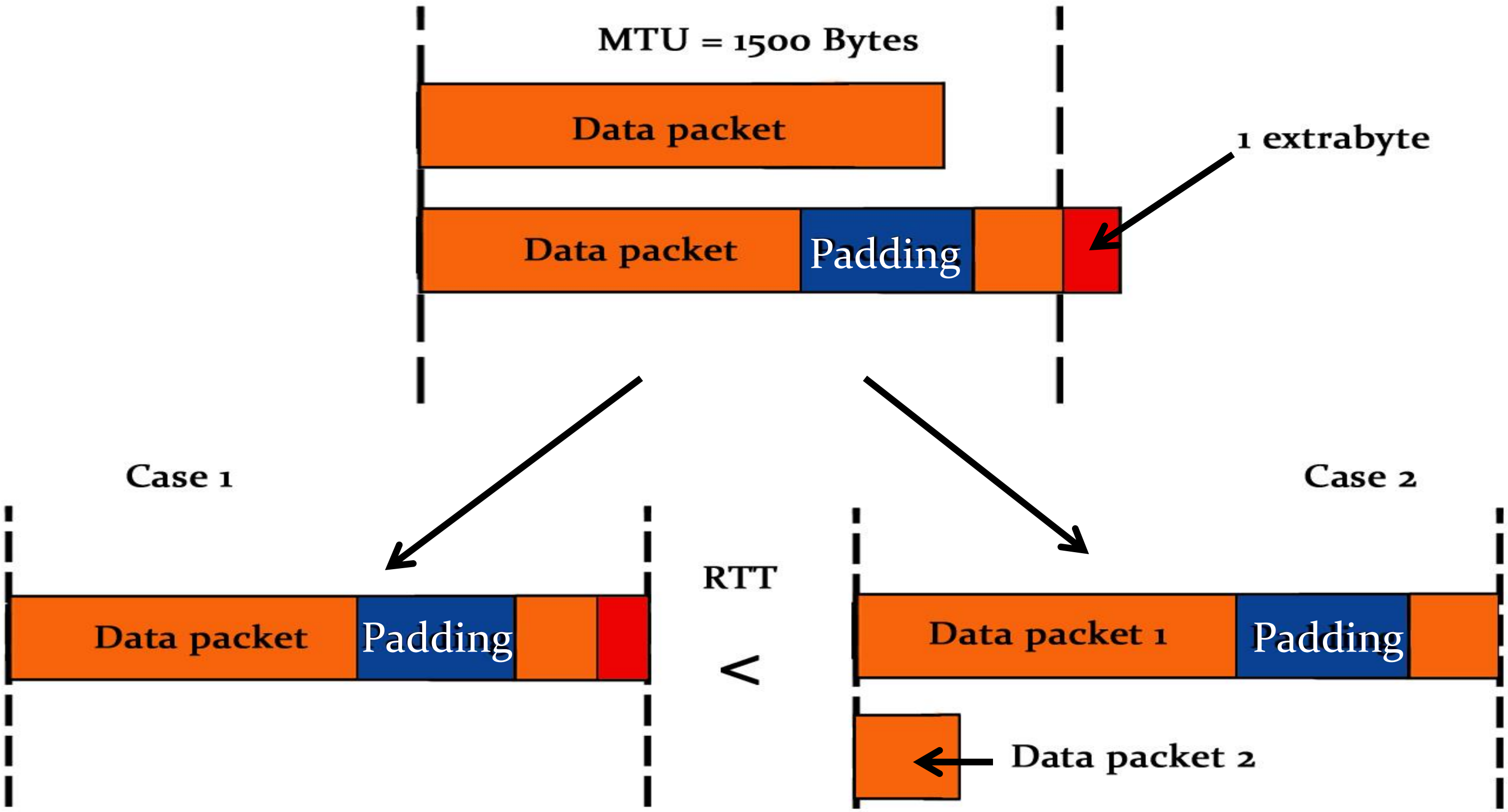
TIME – How it works



TIME – How it works

Screenshot credit - Tal Be'ery BH presentation

TIME – How it works



MTU : Maximum Transmission Unit
RTT: Rount Trip Time

TIME – Feasibility

- No requirements for Man-in-the-Middle
- Concentrate on HTTP responses
- The attacker creates HTTP request with JavaScript and response timing leaks the request size.
- Repeat for few times to void aberration due to network jitter.

TIME – Counter Measures

- Adding random timing delays to the decryption
- Browser should support and respect ``X-Frame-Options”
- Strict restriction on reflection of user input in the response.
- Enable anti-automation techniques like CAPTCHA, CSRF token

Anti-Automation Recommendations

iSECpartners part of nccgroup

- Rate limiting using HAProxy
- Rate limiting via various DDOS protection

Comparison of counter measures

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE		random timing delays	Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	Use RC4	Use RC4	Do not use RC4		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.2	Upgrade to TLS 1.2	Upgrade to TLS 1.2		Separate secrets from user input	Random timing delay

Comparison of counter measures

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE		~~random timing delays~~	Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	Use RC4	Use RC4	Do not use RC4		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.2	Upgrade to TLS 1.2	Upgrade to TLS 1.2		~~Separate secrets from user input~~	~~Random timing delay~~

Comparison of counter measures

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE			Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	Use RC4	Use RC4	Do not use RC4		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.2	Upgrade to TLS 1.2	Upgrade to TLS 1.2			

Comparison of counter measures

iSECpartners part of nccgroup

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE			Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	Use RC4	Use RC4	Do not use RC4		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.2	Upgrade to TLS 1.2	Upgrade to TLS 1.2			

Comparison of counter measures

iSECpartners part of nccgroup

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE			Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	~~Use RC4~~	~~Use RC4~~	~~Do not use RC4~~		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.2	Upgrade to TLS 1.2	Upgrade to TLS 1.2			

Summarizing counter measures

	BEAST	Lucky 13	RC4 Biases	CRIME	BREACH	TIME
CLIENT SIDE	Upgrade browsers			Upgrade browsers with no TLS compression support		Upgrade browsers with X-Frame-Options
SERVER SIDE			Throttle client initiated re-negotiations	Disable TLS compression	Mask the secret	Restrict reflection of user input
	~~Use RC4~~	~~Use RC4~~	~~Do not use RC4~~		anti-automation techniques	anti-automation techniques
	Upgrade to TLS 1.1 +	Upgrade to TLS 1.2	Upgrade to TLS 1.2			

Thank You

- Special Thanks to Shawn, Tom, Michael, Javed, Jonathan, Tim, Josh, Alban, Ryan, Aaron and everybody in **iSEC**

- Whitepaper: Attacks on SSL (bit.ly/1cAqL7o)

- **Pratik Guha Sarkar**
  - Security Consultant at iSEC Partners
  - *psarkar@isecpartners.com* **|** *@pragusa55*

- **Shawn Fitzgerald**
  - Principal Security Consultant at iSEC Partners
  - *shawn@isecpartners.com*