How to defend from an attacker armed with a mathematician

Antonio Sanso (@asanso)

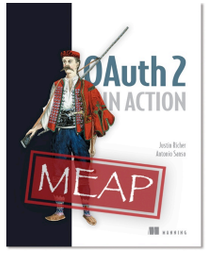
Senior Software Engineer

Adobe Research Switzerland

Who is this guy, BTW?

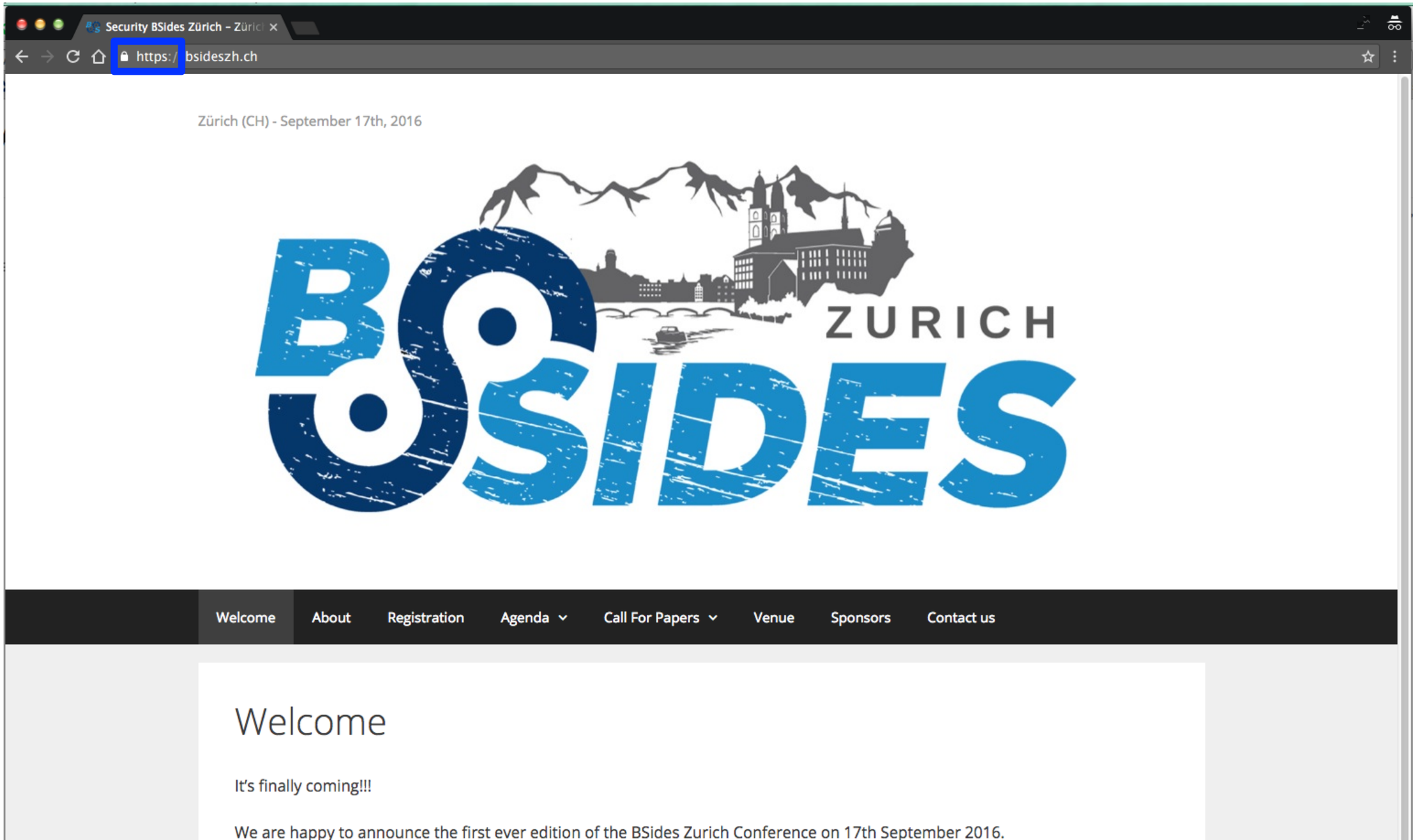
- Senior Software Engineer Adobe Research Switzerland
- Internet Bug Bounty, Google Security Hall of Fame, Facebook Security Whitehat, GitHub Security Bug Bounty, Microsoft Honor Roll

- Found vulnerabilities in OpenSSL , Google Chrome, Apple Safari



- Co-Author of “OAuth 2 in Action”
- 2 3 5 7 11 Obsessed by prime numbers

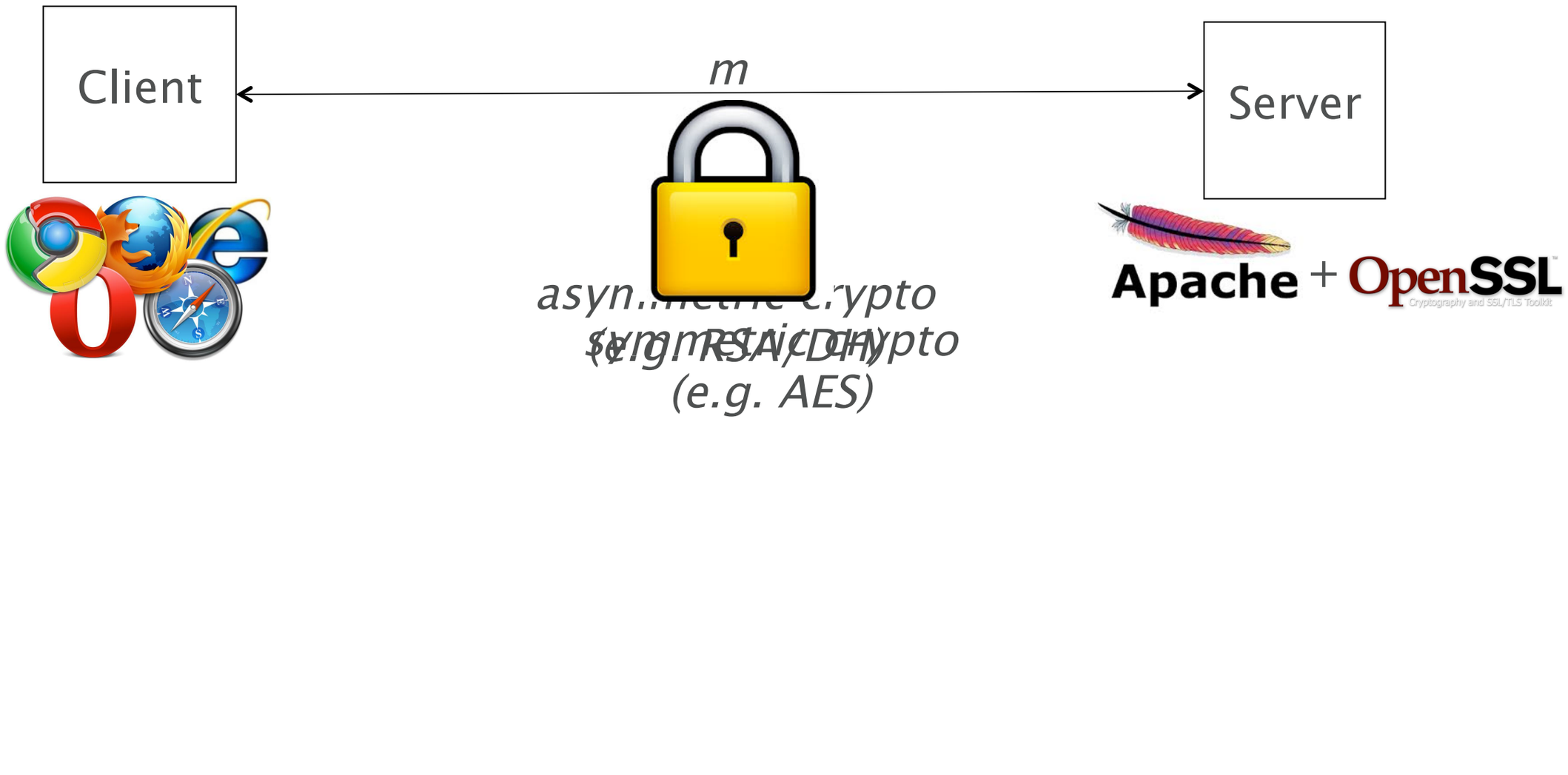
TLS (Transport Layer Security)



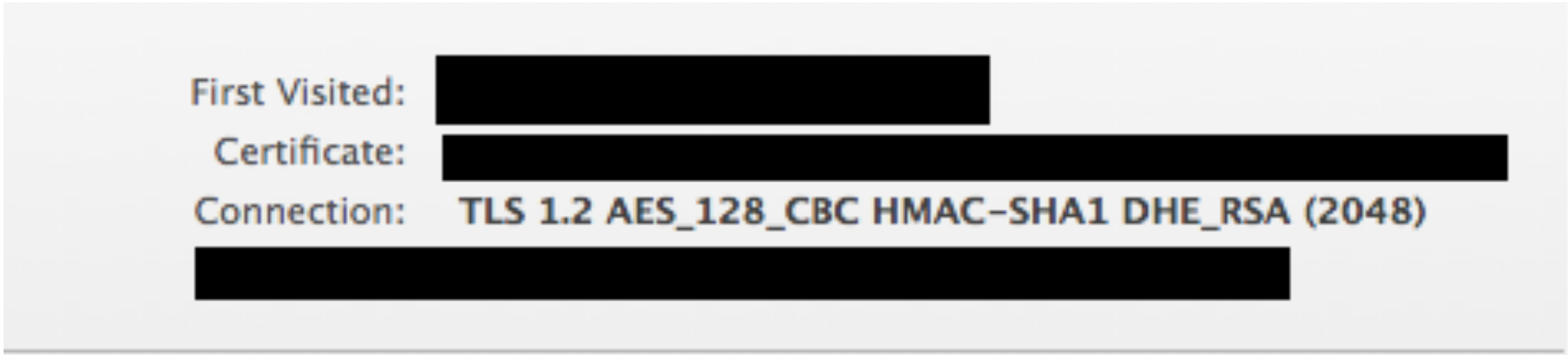
TLS

TLS Flow (highly simplified)

m = premaster key



TLS

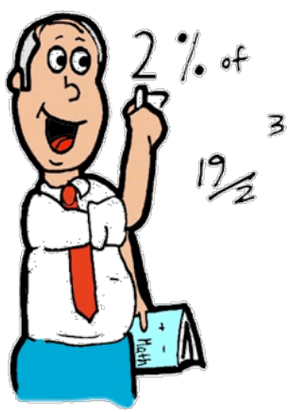

First Visited:
Certificate:
Connection: TLS 1.2 AES_128_CBC HMAC-SHA1 DHE_RSA (2048)

Key Exchange

m = premaster key

Client

Server




CAUTION
MATH
AHEAD

Math Ahead !!

- Addition
- Multiplication
- Exponentiation
- Modular arithmetic

Public key cryptography

New direction in cryptography

New Directions in Cryptography

Invited Paper

Whitfield Diffie and Martin E. Hellman

Abstract Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

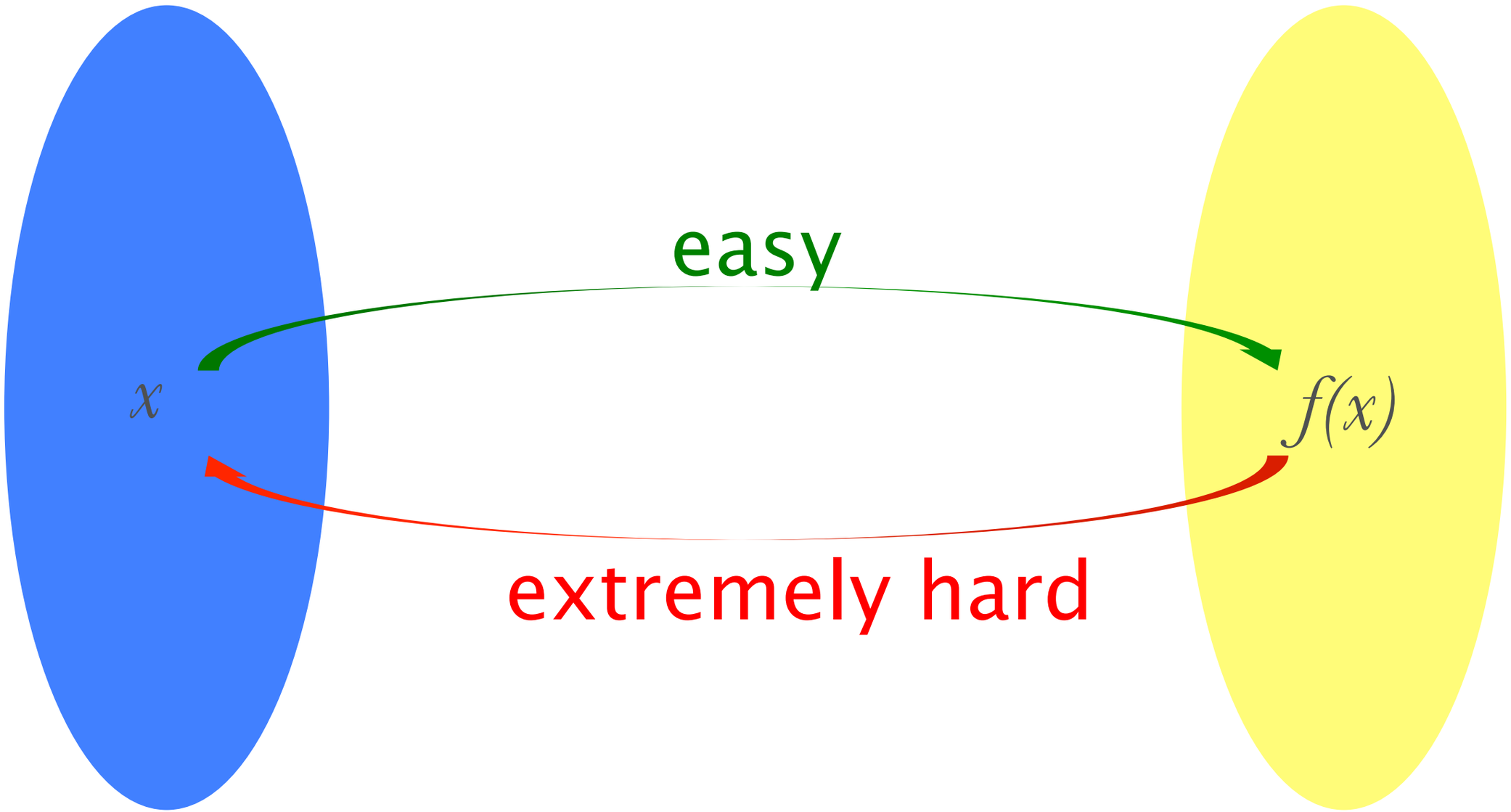
1 INTRODUCTION

We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where

communications over an insecure channel order to use cryptography to insure privacy, however, it currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such a private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channel without compromising the security of the system. In *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infeasi-

One way functions



Discrete log problem

$y = g^x \pmod p$ → EASY!

Example

p = 17

g = 3

x = 4

$y = 3^4 \pmod{17} = ?$

$y = 3^4 = 81 \pmod{17} = 13$

Discrete log problem

$\text{dlog}_g\ y$ ⟸ HARD!

How much hard? Trivial alg. O(p)

We need do find $0 < x < p-1$ s.t. $g^x = y \pmod p$

Example

$p = 17$

$g = 3$

$y = \boxed{10}$ $x = ??$

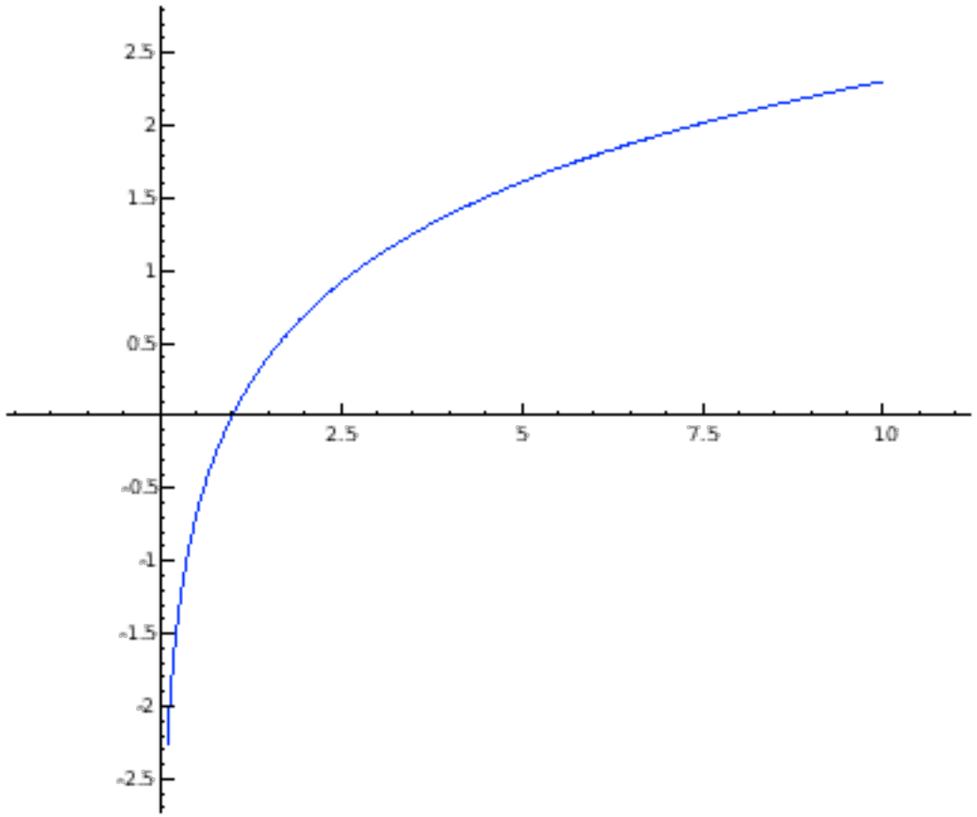
$x = 0$	$y = 3^0 = 1 \pmod{17}$
$x = 1$	$y = 3^1 = 3 \pmod{17}$
$x = 2$	$y = 3^2 = 9 \pmod{17}$
$\mathbf{x = \boxed{3}}$	$\mathbf{y = 3^3 = \boxed{10} \pmod{17}}$

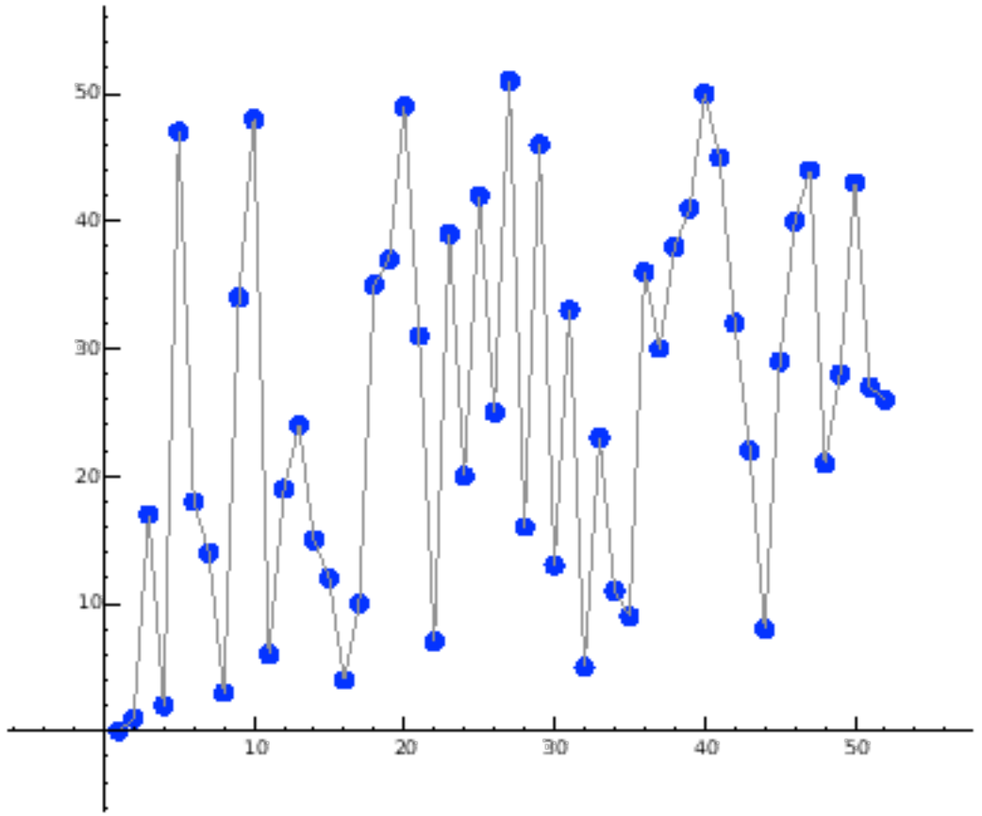
Discrete log problem

p =

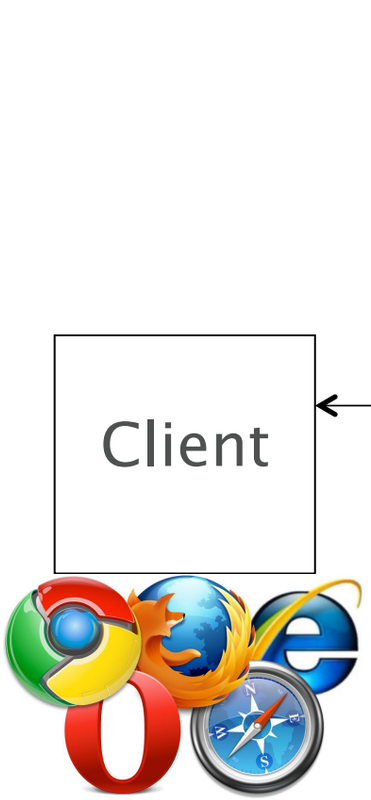
2184735958988820847550672491716226506357140198532537036763136178111402965302595681515760532819041114104416068981574131938119653297987150003897986230915873825094511855496162682415230753660587261650288428887806246705277760522784670978185061479274845883895134220481260183811293780537178260038010602052288440645282381882445568398204288292818343119459318917143106637113851025297964851355307876258459614742745683728962300887936482947770518363614930412099894865427813387402671118849431177088351488936335138006452041345960269614135394940797181007184835412786872593405781105228551172607095195482862576198479783107980185782843 1

Discrete log problem





Key Exchange



Client

y_b g, p

Ephemeral

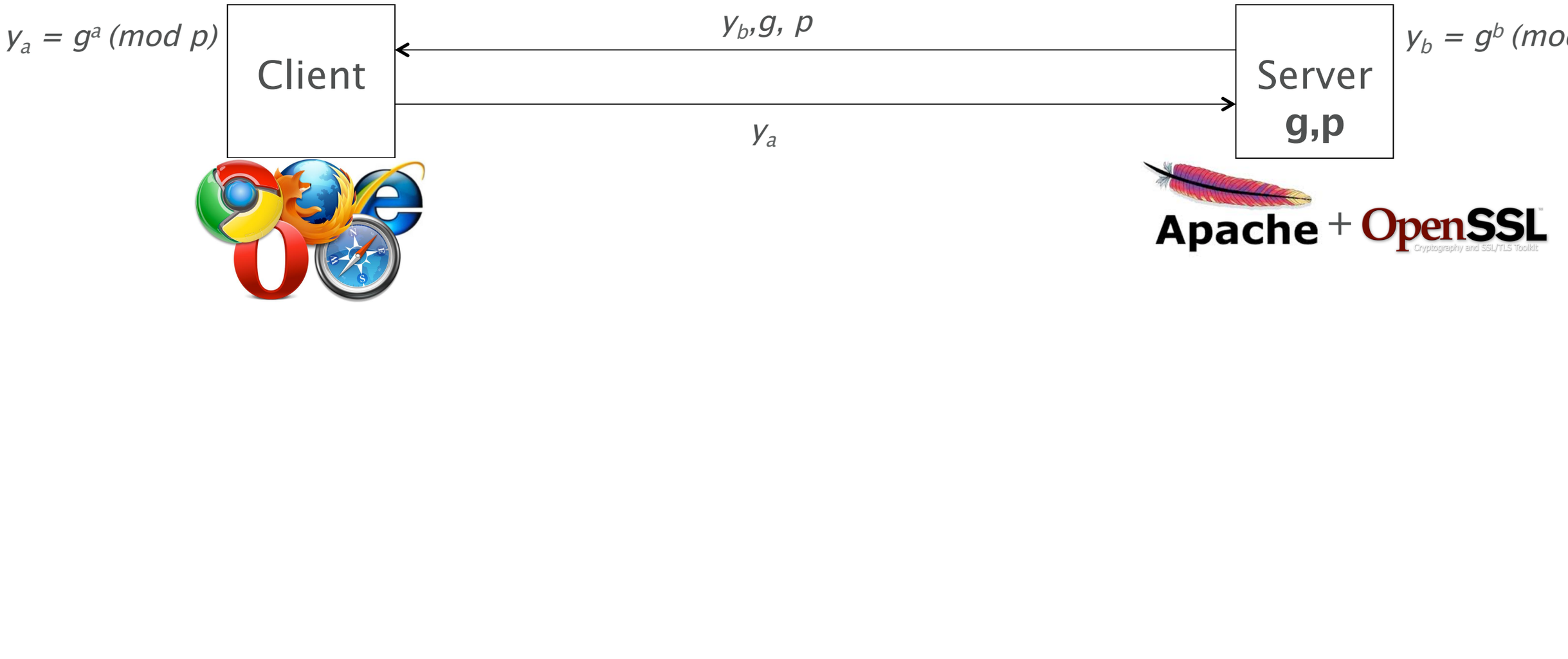
TLS_DHE_RSA_WITH_AES_128….

Random b

$y_b = g^b \pmod p$

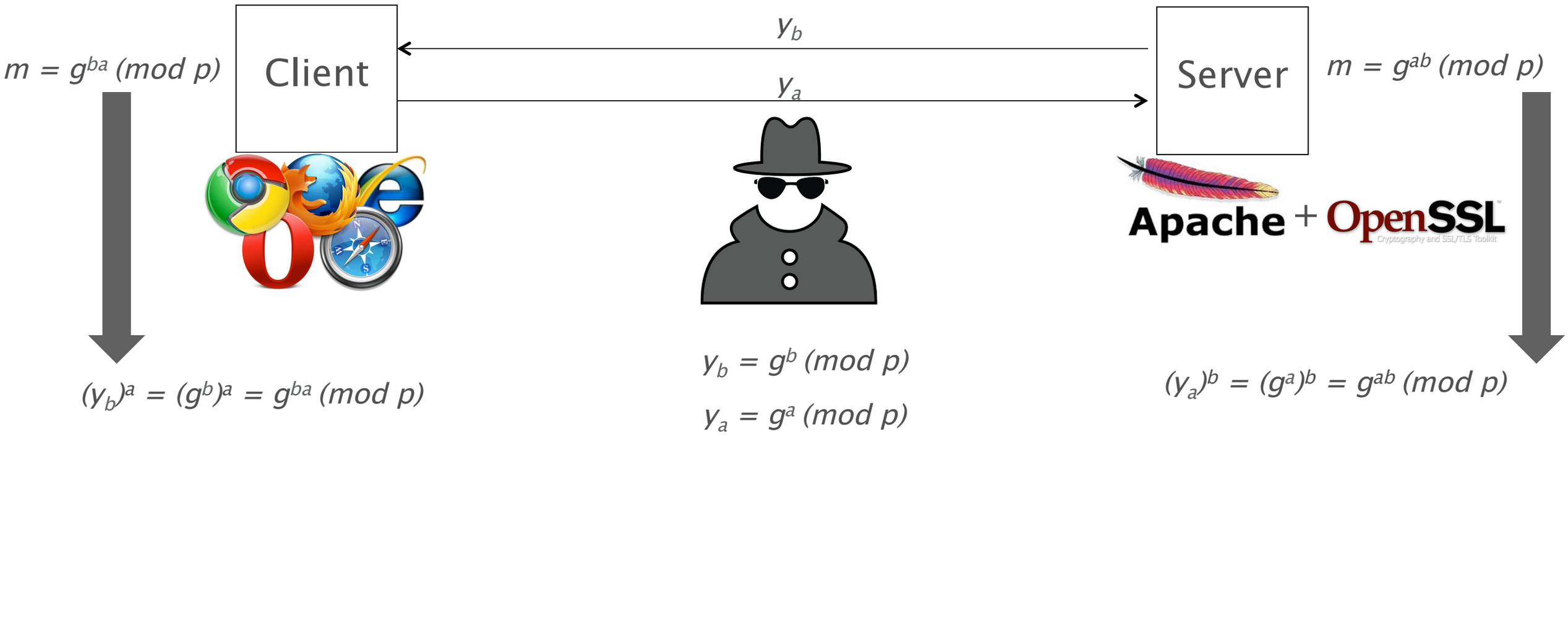
Server
g,p

Key Exchange



Key Exchange

m = premaster key



OpenSSL Key Recovery Attack on DH small subgroups (CVE-2016-0701)

High-severity bug in OpenSSL allows attackers to decrypt HTTPS traffic

OpenSSL maintainers release update that fixes key-recovery bug. Patch now.

DAN GOODIN - 1/28/2016, 7:42 PM

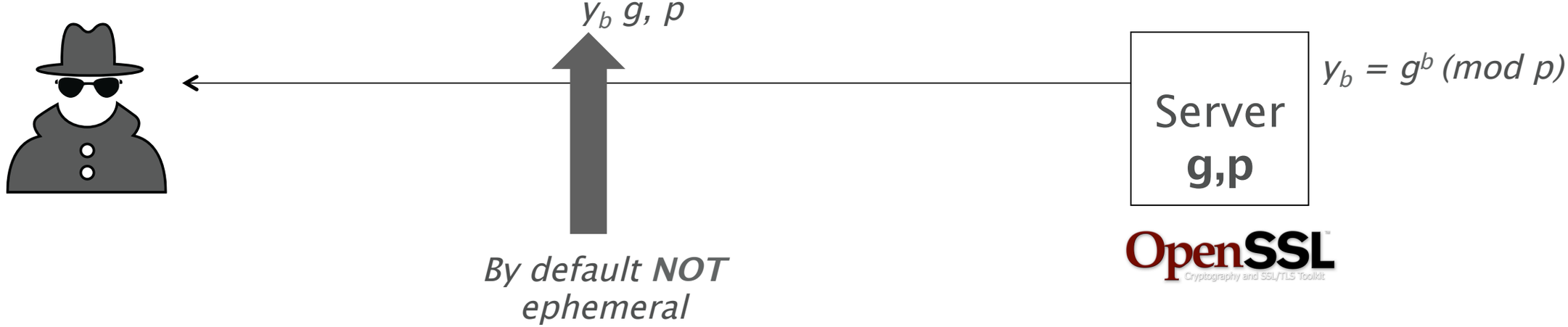
Hudson

OpenSSL Key Recovery Attack on DH small subgroups (CVE-2016-0701)

Attack Requirement #1

(Static ephemeral keys)

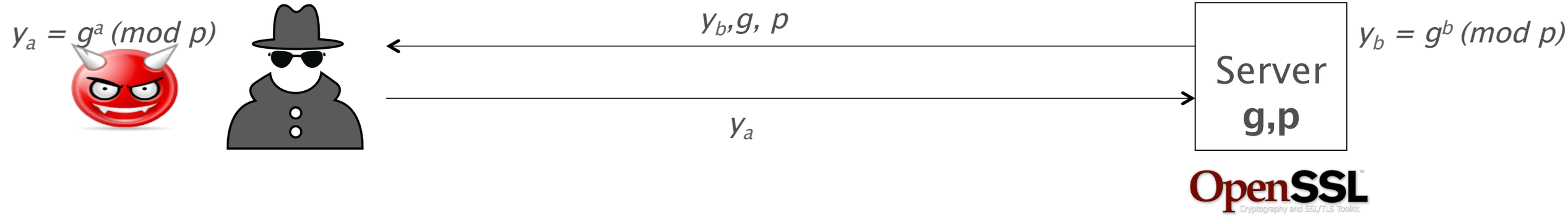
m = premaster key



SSL_OP_SINGLE_DH_USE *was*n't set

Attack Requirement #2

(Not verifying the incoming value)



OpenSSL
Cryptography and SSL/TLS Toolkit

$m = g^{ab} \pmod p$

$(y_a)^b = (g^a)^b = g^{ab} \pmod p$

Discrete log problem

What about p? Which prime number should I choose?

OBSERVATION: (p-1) is always even

{ **Safe prime** ==> (p -1)/2 is also prime (many RFCs)

{ **Or at least** == > (p -1)/2 should not be a product of small primes

RFC 5114 - O(224)

p = 2048 bit

g = >>2

q = 224

p -1 /2 = 2 * 3 * 3 * 5 * 43 * 73 * 157 * 387493 * 605921 * 5213881177 * 3528910760717 * 835018070204734293 49 * C489 (where C489 is a 489 digit composite number with no small factors).

Attack Requirement #3

(Product of small primes)

m = premaster key

y_b *g, p*

RFC 5114

$y_b = g^a \pmod p$

OpenSSL
Cryptography and SSL/TLS Toolkit

Attack succeeded!!

m = premaster key

$y_a = g^a \pmod p$

y_b g, p

y_a

By default **NOT** ephemeral

RFC 5114

$y_b = g^b \pmod p$

OpenSSL Cryptography and SSL/TLS Toolkit

So we can compute:

{ *b mod (2)*

{ *b mod (3)*

{ *…*

{ *b mod (*5213881177*)*

Finally we can combine the result using the Chinese Remainder Theorem **(CRT)!!**

History of RFC 5114

*...a **semi-mysterious** RFC 5114 – Additional Diffie-Hellman Groups document. It introduces new MODP groups not with higher sizes, but just with different primes.*

*... no one really wanted these, but no one really objected to it either, so the document (**originating from Defense contractor BBN**) made it to RFC status.*

References

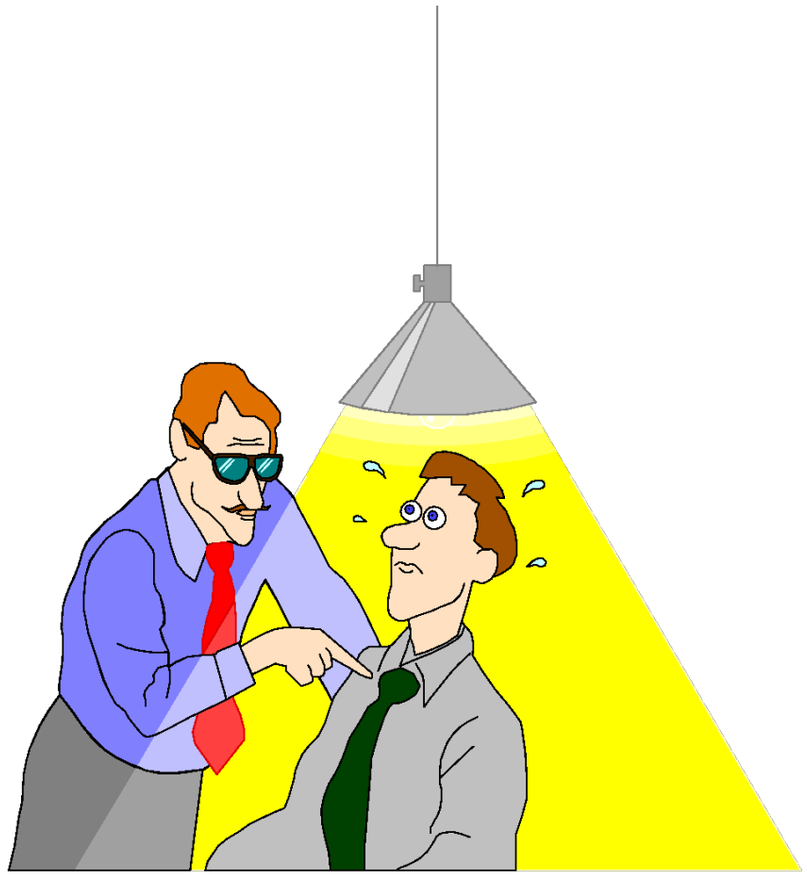
{ https://www.openssl.org/news/secadv/20160503.txt

{ http://blog.intothesymmetry.com/2016/01/openssl-key-recovery-attack-on-dh-small.htmlMultiplication

{ http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.5296

{ http://arstechnica.com/security/2016/01/high-severity-bug-in-openssl-allows-attackers-to-decrypt-https-traffic/

Questions?



Additional material

Discrete log function

Primitive root (generator)

Example

p = 17

Generator g = 3

$\{3^0,3^1,3^2,3^3,3^4,3^5,3^6, \ldots\} ==> \{1, 3, 9, 10, 13, 5, 15, \ldots\}$

OpenSSL Key Recovery Attack on DH small subgroups (CVE-2016-0701)

y_b g, p

y_a

Server
g,p

$y_b = g^a \pmod p$

OpenSSL
Cryptography and SSL/TLS Toolkit

Choose B s.t. ord(B) is small

$y_a = g^a B \pmod p$

$(y_b)^a B^j = g^{ba} B^j$

$0 < j < ord(B)$ == > *small*

$B = j \bmod (ord(B))$

$(y_a)^b = (g^a B)^b = g^{ab} B^b \pmod p$

RFC 5114

p -1 /2 = 2 * 3 * 3 * 5 * 43 * 73 * 157 * 387493 * 605921 * 5213881177 * 352891076071 7 * 8350180702047342934 9 * C489 (where C489 is a 489 digit composite number with no small factors).

- $\exists\ g_i$ s.t. q = 2
- $\exists\ g_i$ s.t. q = 3
- $\exists\ g_i$ s.t. q = 5
- ...

from O(224) to O(44)

★ OpenSSL Key Recovery Attack on DH small subgroups – Validation

Choose B s.t. ord(B) is small
$y_a = g^a B \pmod p$

y_b g, p

y_a

Server
g,p

$y_b = g^a \pmod p$

OpenSSL
Cryptography and SSL/TLS Toolkit

VALIDATION

Check that
$y_a^q \pmod p = 1$

★ Chinese Remainder Theorem (CRT)

$b = l \pmod{ord(B_1)}$

$b = m \pmod{ord(B_2)}$

$b = n \pmod{ord(B_3)}$

We can find a unique solution

$mod(ord(B_1) * ord(B_2) * ord(B_3))$

“Trap door” one way functions