Uncovering Cryptographic Failures with Internet-Wide Measurement

Zakir Durumeric

University of Michigan

Who am I?

My research focuses on measurement-driven security.

① Developing tools for researchers to better measure the Internet

zmap

② Using this perspective to understand how systems are deployed in practice

censys

Neither Snow Nor Rain Nor MITM… An Empirical Analysis of Email Delivery Security

Zakir Durumeric, David Adrian, Ariana Mirian, James Kasten,
Kurt Thomas, Vijay Eranti, Nicholas Lidzborski,
Elie Bursztein, Michael Bailey, J. Alex Halderman

E-mail Security in Practice

As originally conceived, SMTP had no built-in security

We've extended with SMTP with new extensions to:

1. Encrypt e-mail in transit

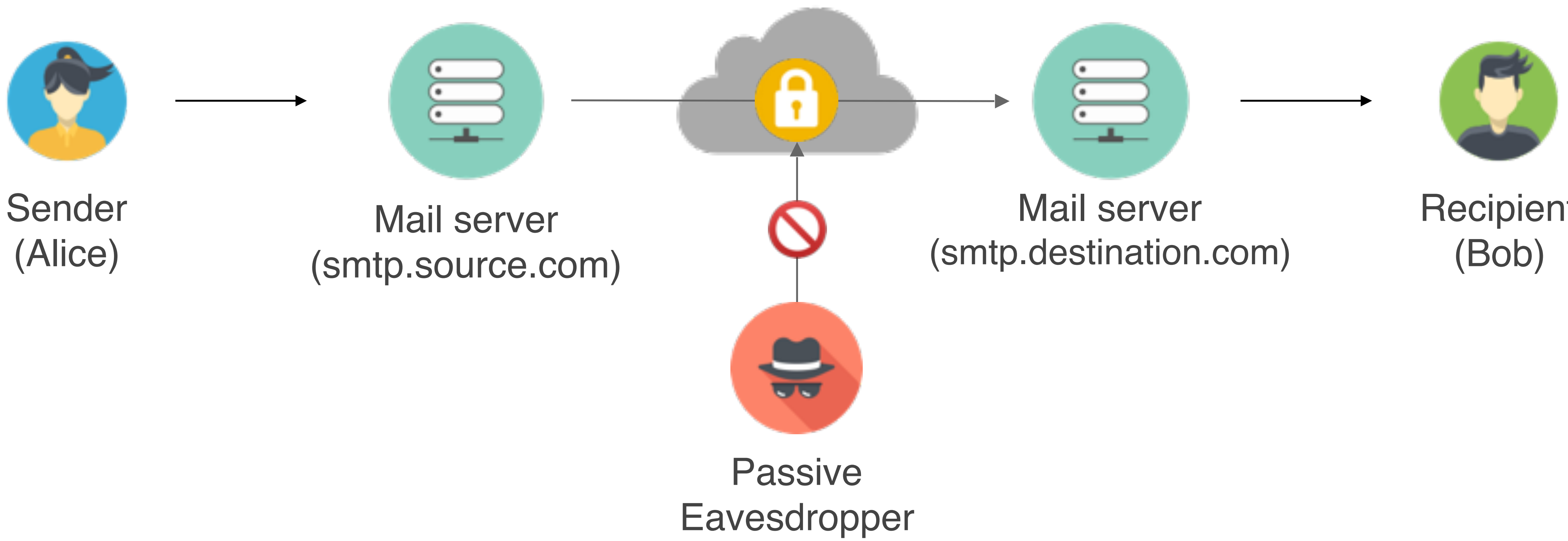
2. Authenticate email on receipt

However, deployment is voluntary and message security is hidden from the end user

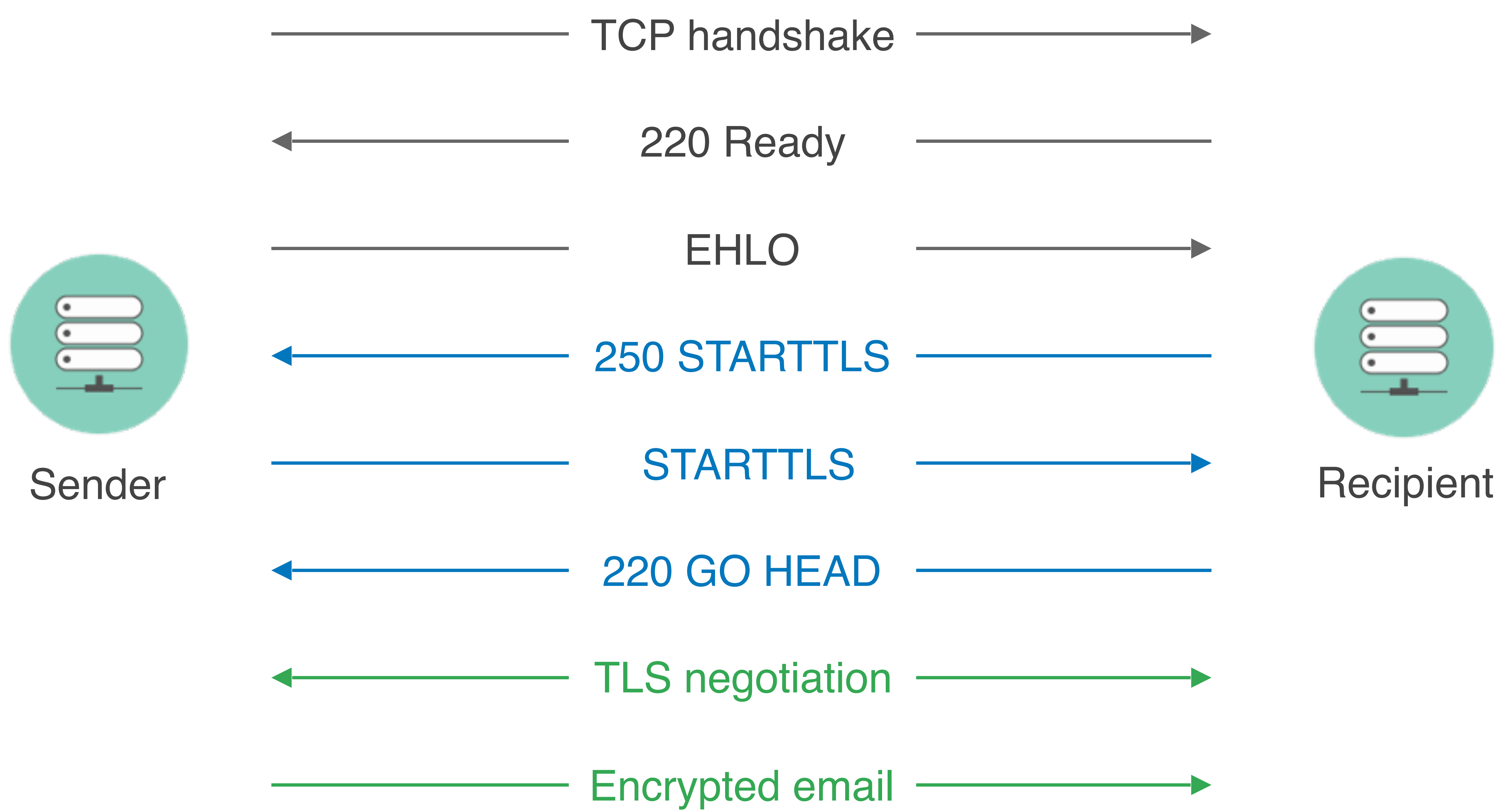
STARTTLS: TLS for SMTP

Allow TLS session to be started during an SMTP connection

Mail is transferred over the encrypted session



STARTTLS Protocol



Opportunistic Encryption Only

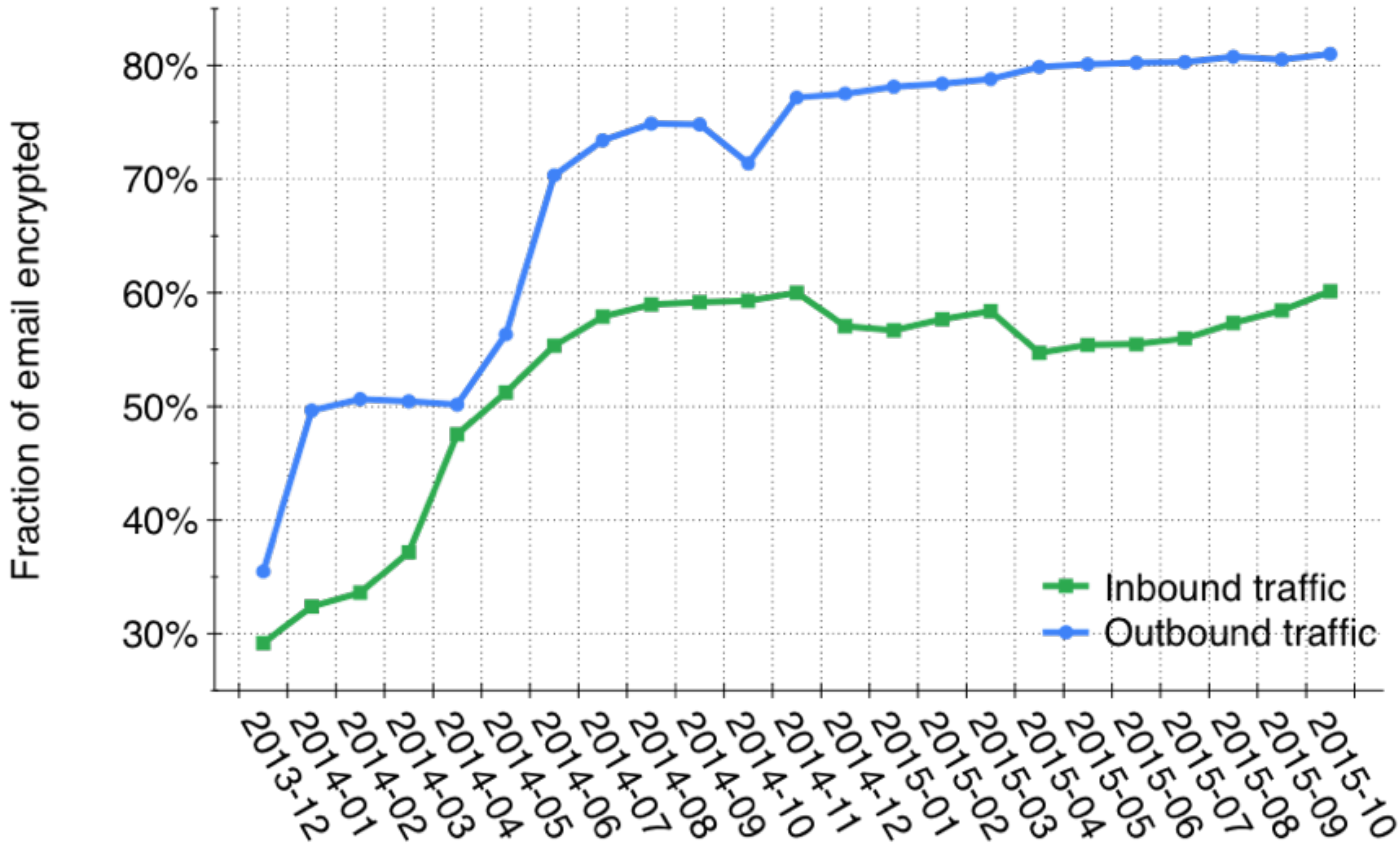
Unlike HTTPS, STARTTLS is used opportunistically

Senders do not validate destination servers — the alternative is cleartext

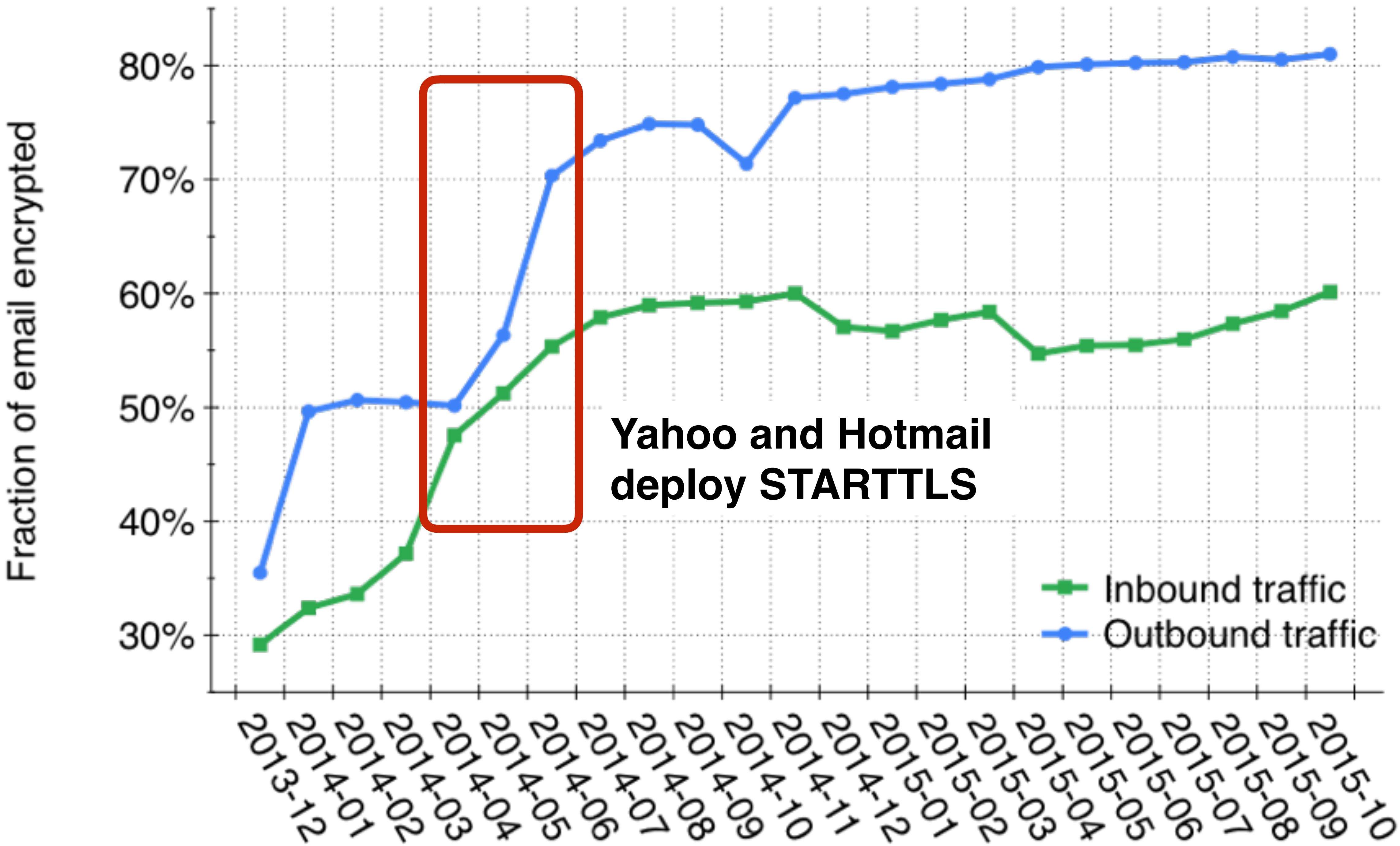
Many servers do not support STARTTLS

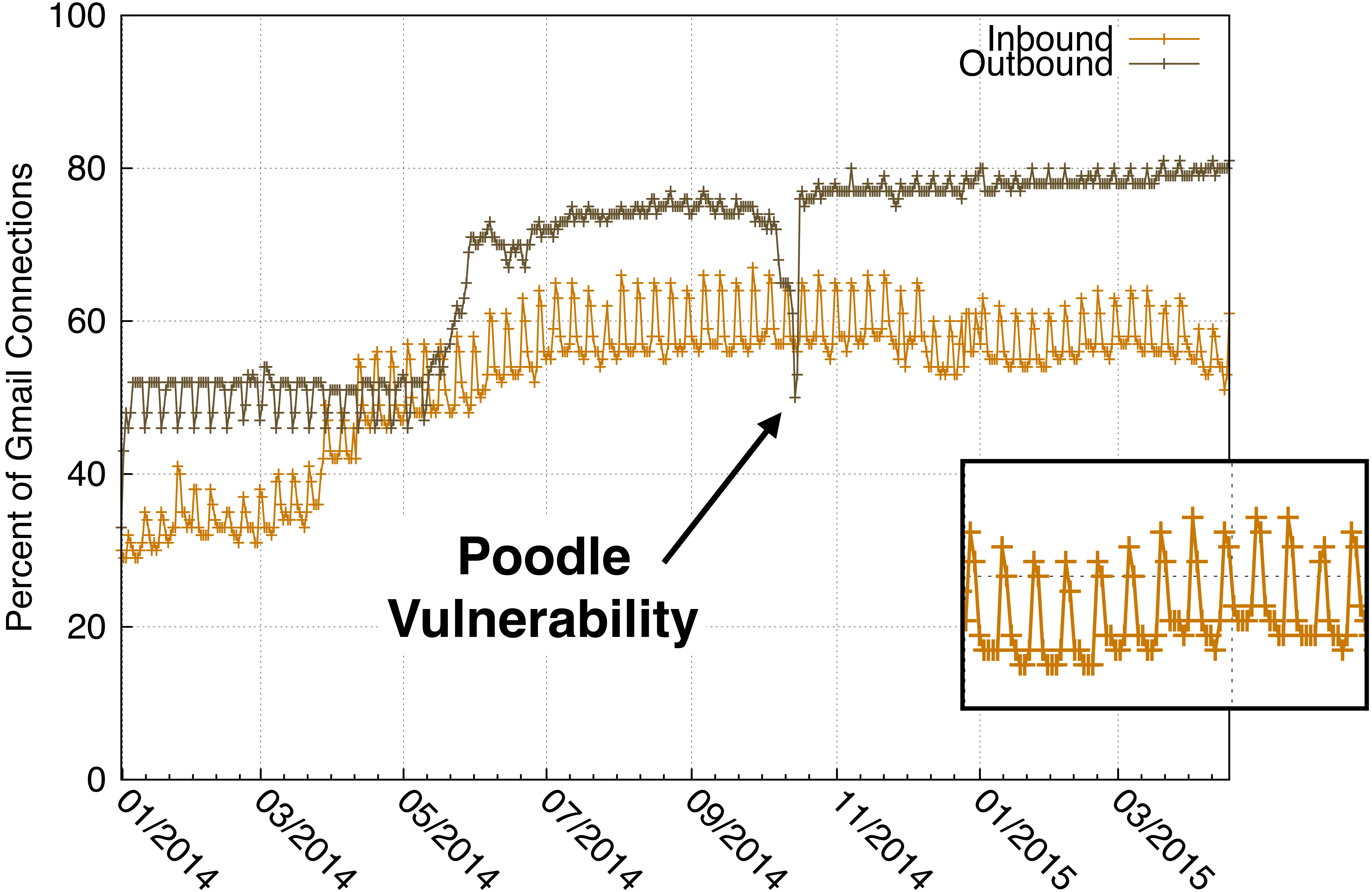
“A publicly-referenced SMTP server MUST NOT require use of the STARTTLS extension in order to deliver mail locally. This rule prevents the STARTTLS extension from damaging the interoperability of the Internet's SMTP infrastructure.” (RFC3207)

STARTTLS Usage as seen by Gmail



STARTTLS Usage as seen by Gmail





Percent of Gmail Connections
100
80
60
40
20
0
Inbound
Outbound
Poodle
Vulnerability
01/2014
03/2014
05/2014
07/2014
09/2014
11/2014
01/2015
03/2015

Long Tail of Mail Operators

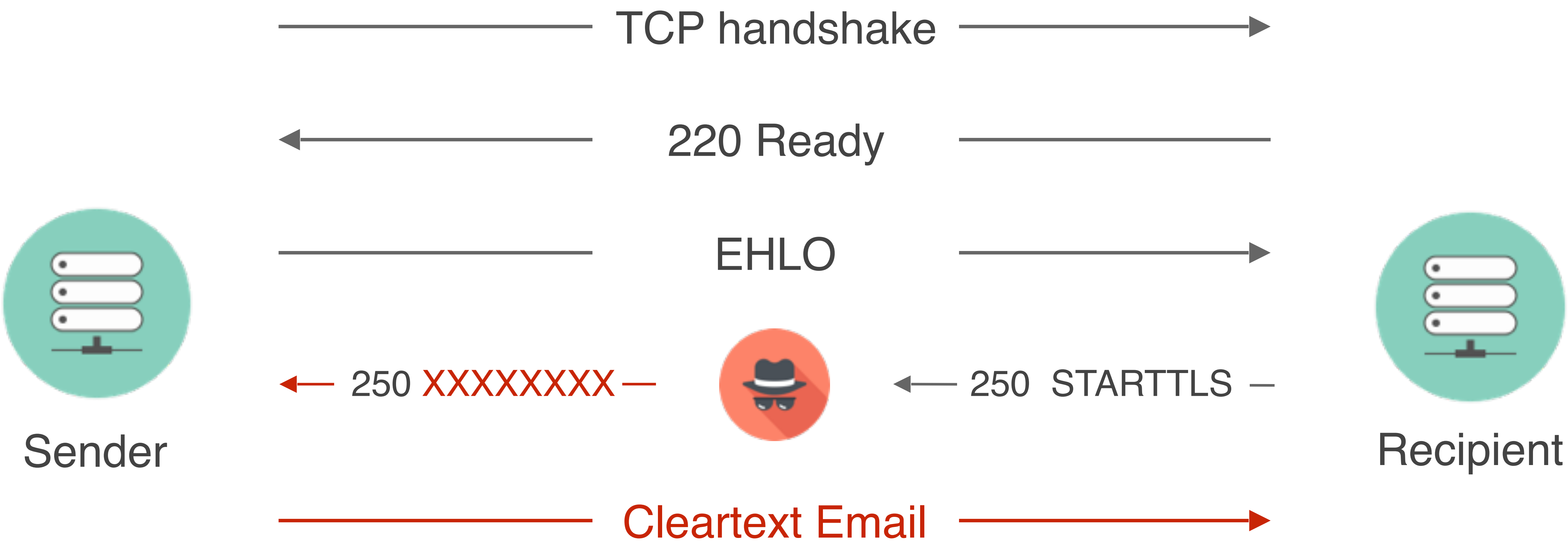
These numbers are dominated by a few large providers.

Of the Alexa Top 1M with Mail Servers:

- 81.8% support STARTTLS
- 34% have certificates that match MX server
- 0.6% have certificates that match domain (which would allow true authentication)

Not currently feasible to require STARTTLS

Attack 1: STARTTLS Stripping



STARTTLS Stripping in the Wild

Country	
Tunisia	96.1%
Iraq	25.6%
Papua New Guinea	25.0%
Nepal	24.3%
Kenya	24.1%
Uganda	23.3%
Lesotho	20.3%
Sierra Leone	13.4%
New Caledonia	10.1%
Zambia	10.0%

Authenticating Email

Authenticating Email

DomainKeys Identified Mail (DKIM)

Sender signs messages with cryptographic key

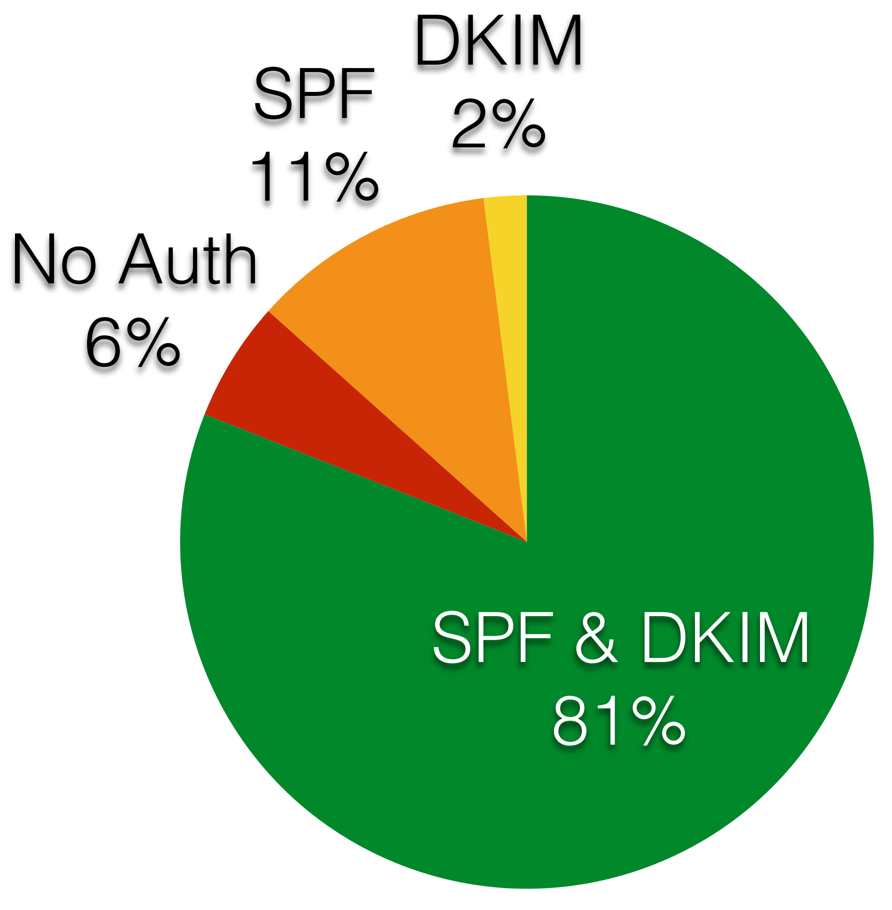
Sender Policy Framework (SPF)

Sender publishes list of IPs authorized to send mail

Domain Message Authentication, Reporting and Conformance (DMARC)

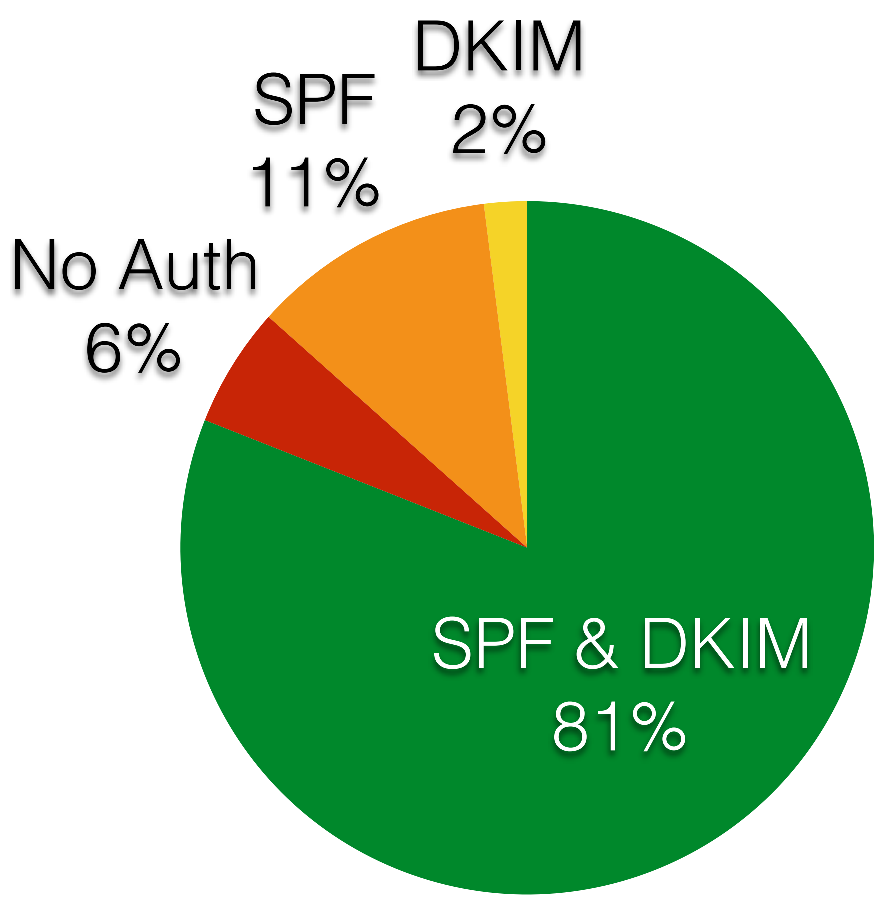
Sender publishes policy in DNS that specifies what to do if DKIM or SPF validation fails

E-mail Authentication in Practice



Gmail Authentication

E-mail Authentication in Practice



Gmail Authentication

Technology	Top 1M
SFP Enabled	47%
DMARC Policy	1%

DMARC Policy	Top 1M
Reject	20%
Quarantine	8%
Empty	72%

Top Million Domains

Moving Forward

Two IETF proposals to solve real world issues:

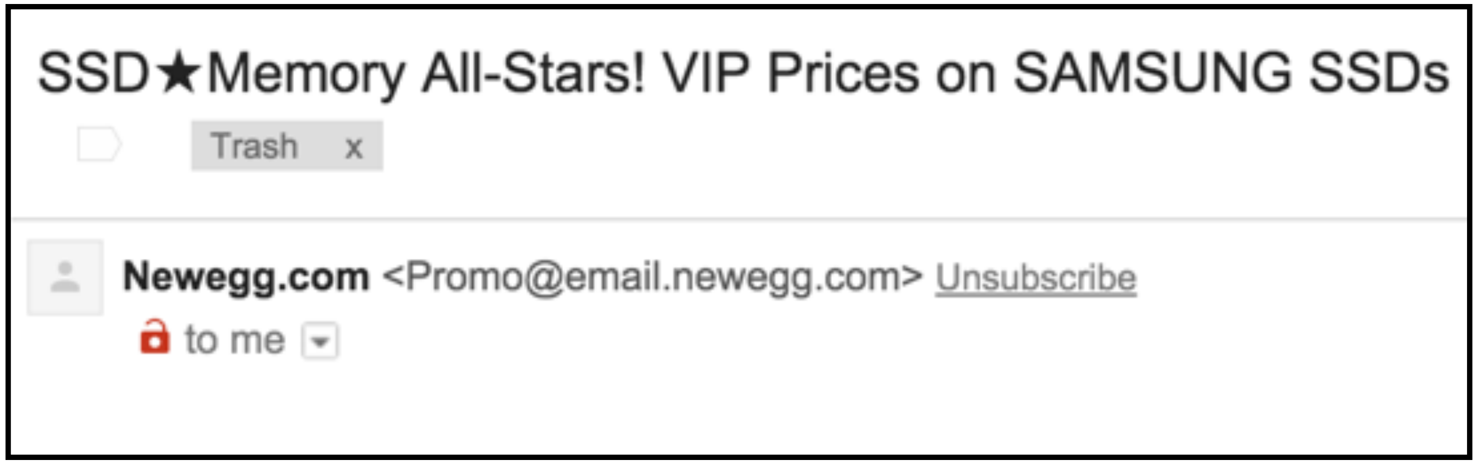
SMTP Strict Transport Security

Equivalent to HTTPS HSTS (key pinning)

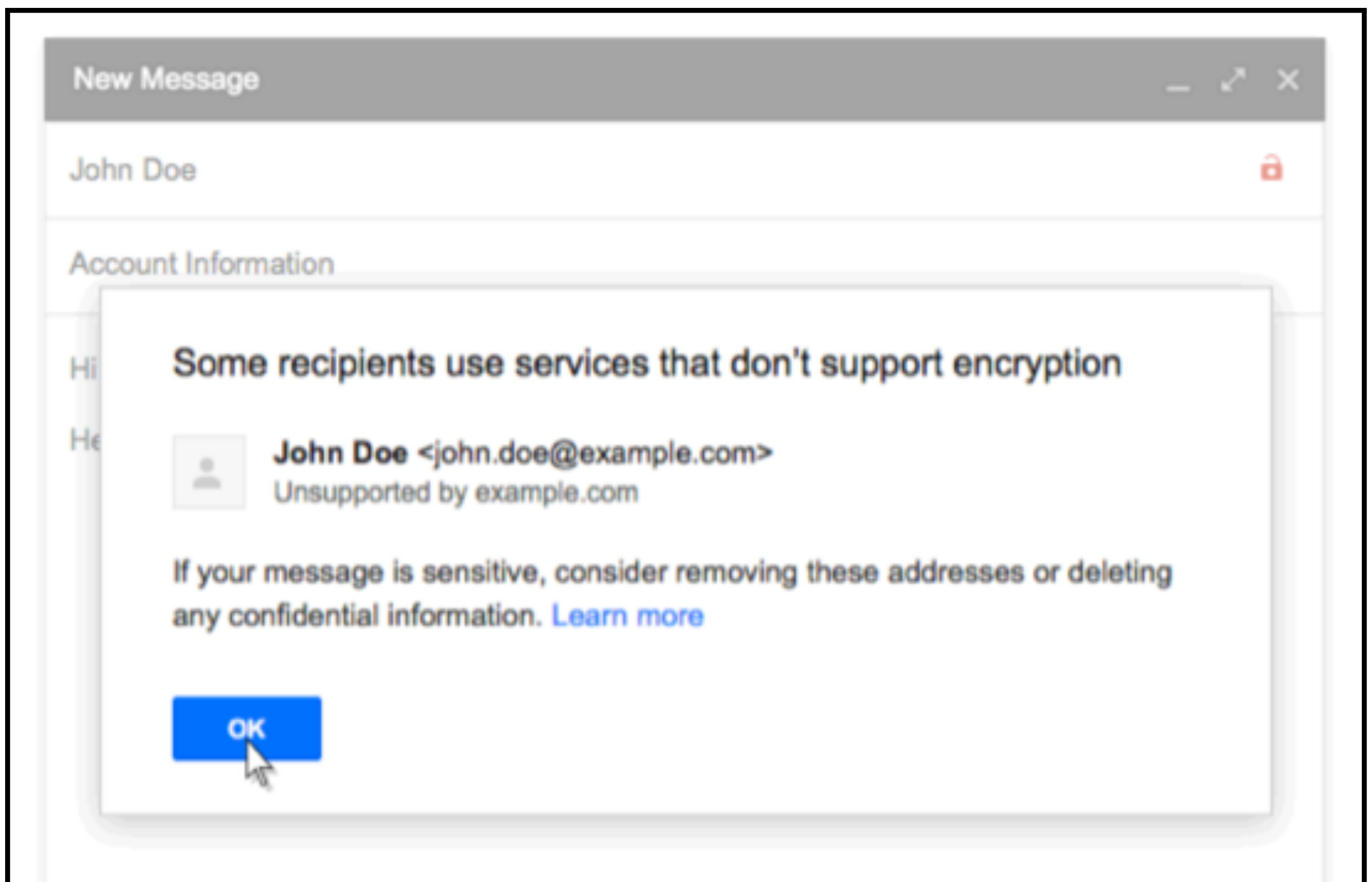
Authenticated Received Chain (ARC)

DKIM replacement that handles mailing lists

Gmail STARTTLS Indication

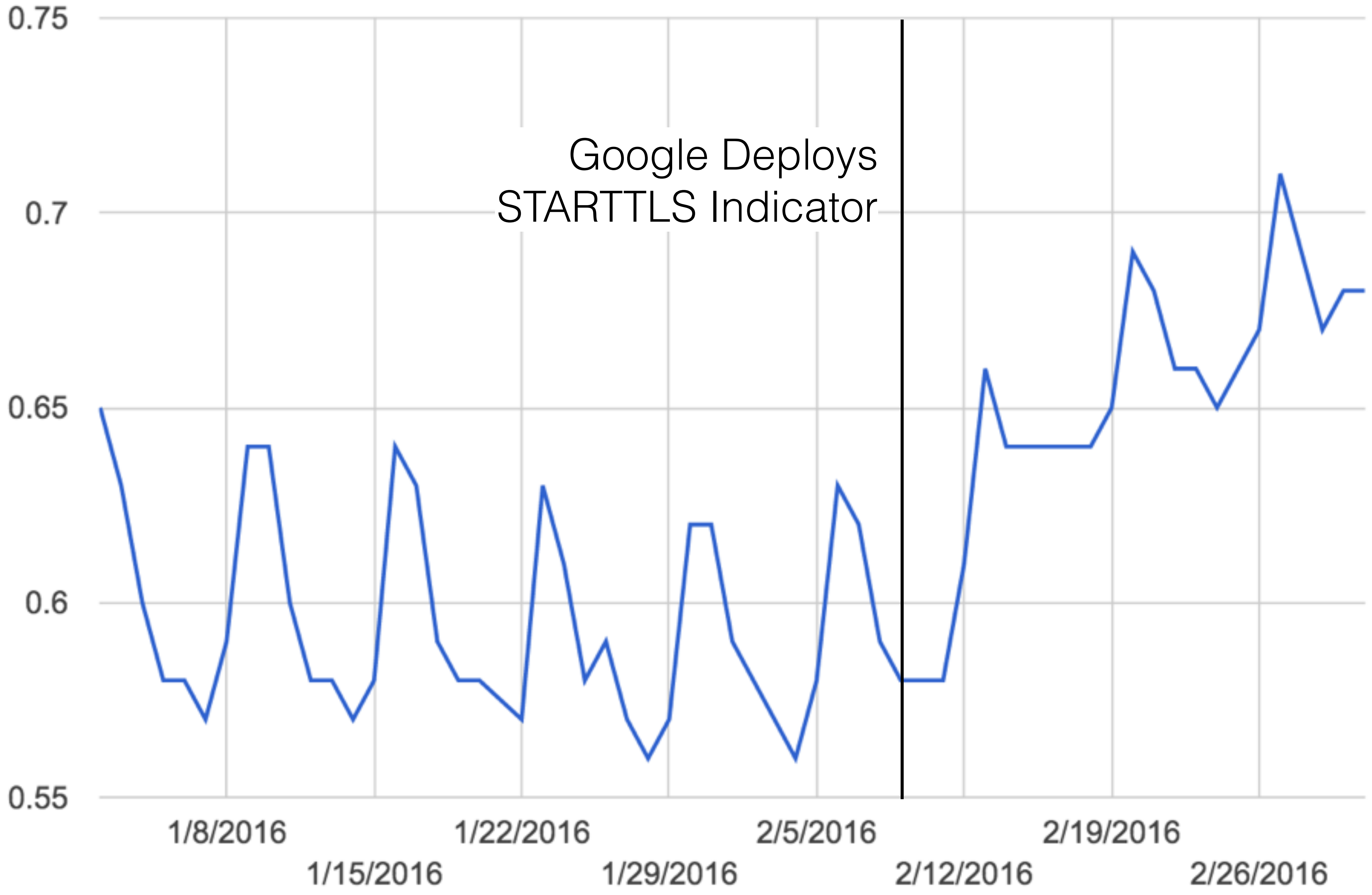


Insecure Received



Insecure Sending

Inbound Gmail Protected by STARTLES



Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

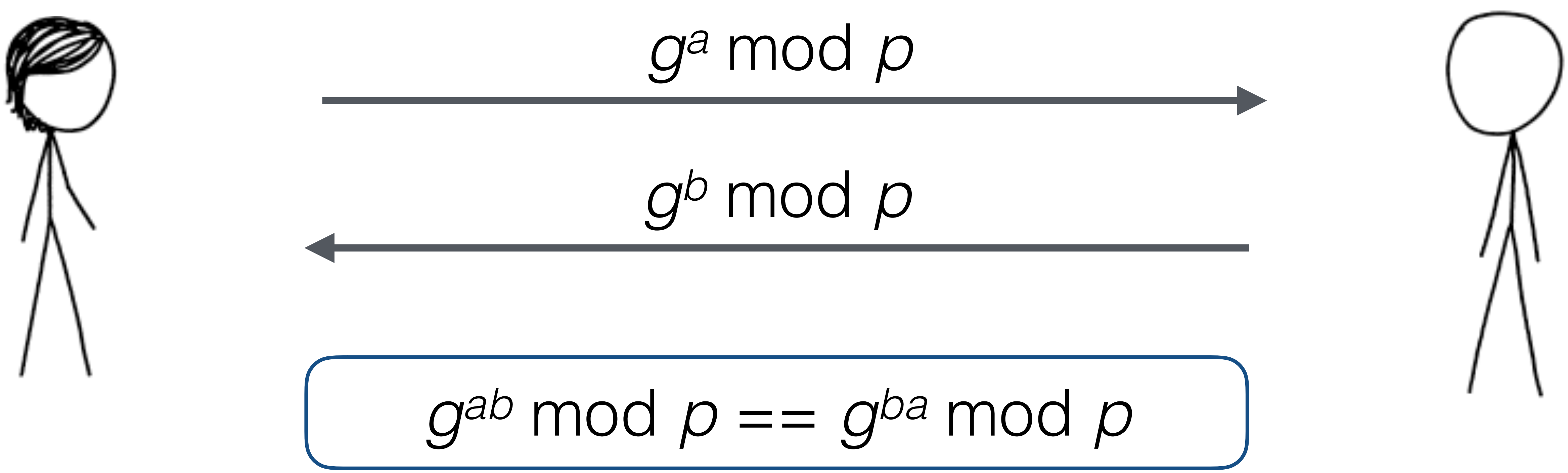
David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J . Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Beguelin, and Paul Zimmermann

Diffie-Hellman Key Exchange

First published key exchange algorithm

Public Parameters

- *p* (a large prime)
- *g* (generator for group *p*)



Diffie-Hellman on the Internet

Diffie-Hellman is pervasive on the Internet today

Primary Key Exchange

- SSH
- IPSEC VPNs

Ephemeral Key Exchange

- HTTPS
- SMTP, IMAP, POP3
- all other protocols that use TLS

"Sites that use perfect forward secrecy can provide better security to users in cases where the encrypted data is being monitored and recorded by a third party."

"Ideally the DH group would match or exceed the RSA key size but 1024-bit DHE is arguably better than straight 2048-bit RSA so you can get away with that if you want to."

"With Perfect Forward Secrecy, anyone possessing the private key and a wiretap of Internet activity can decrypt nothing."

2015 Diffie-Hellman Support

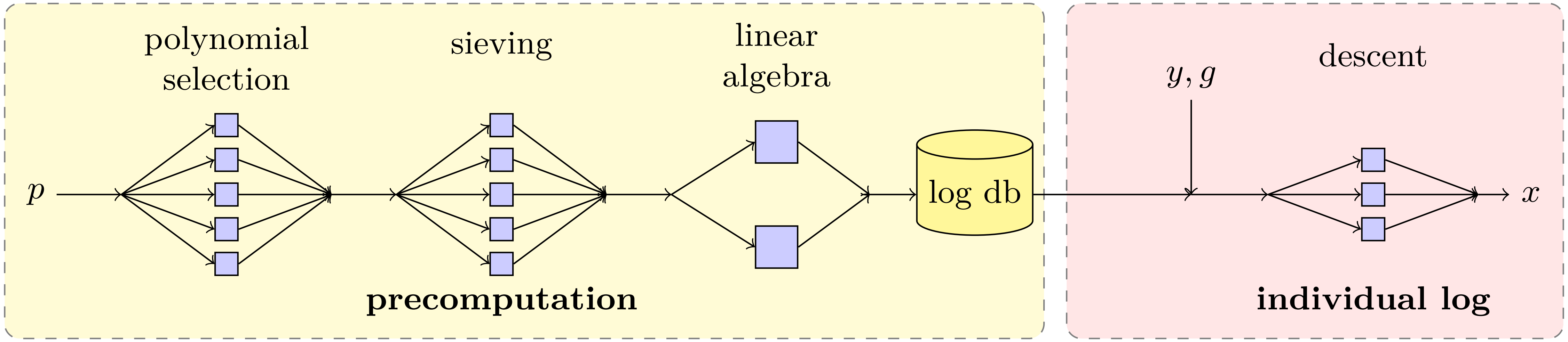
Protocol	Support
HTTPS (Top Million Websites)	68%
HTTPS (IPv4, Browser Trusted)	24%
SMTP + STARTTLS	41%
IMAPS	75%
POP3S	75%
SSH	100%
IPSec VPNs	100%

Breaking Diffie-Hellman

Computing discrete log is best known attack against DH

In other words, Given $g^x \equiv y \bmod p$, compute x

Number Field Sieve

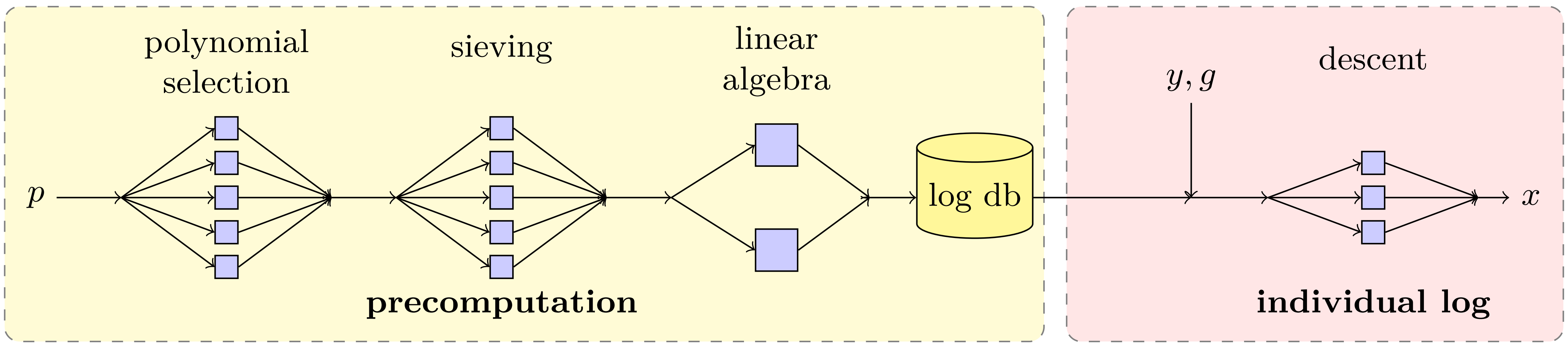


Breaking Diffie-Hellman

Computing discrete log is best known attack against DH

In other words, Given $g^x \equiv y \bmod p$, compute x

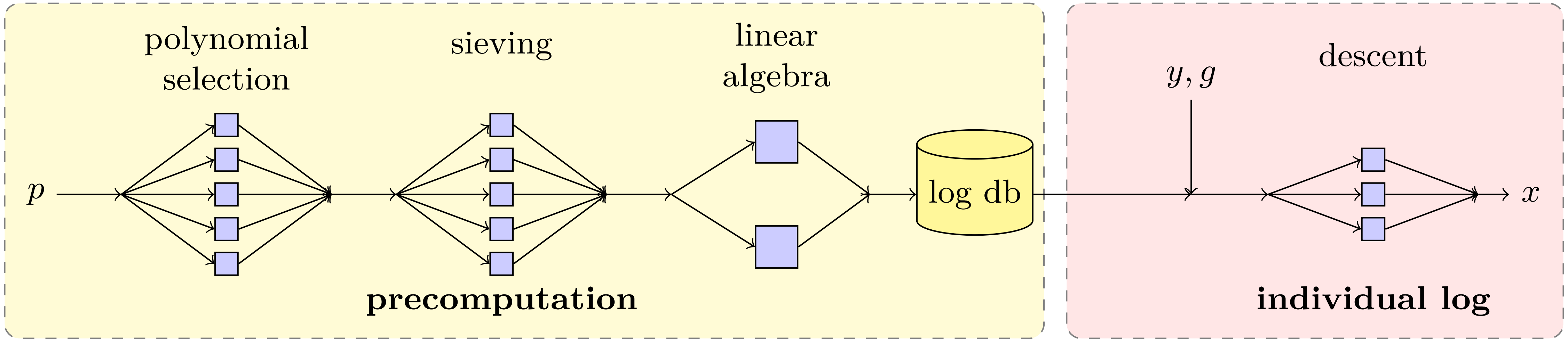
Number Field Sieve



Pre-computation is only dependent on *p*!

Breaking Diffie-Hellman

Number Field Sieve



	Sieving	Linear Algebra	Descent
DH-512	2.5 core years	7.7 core years	10 core min.

Lost in Translation

This was known within the cryptographic community

However, not within the systems community

66% of IPSec VPNs use a single 1024-bit prime

Lost in Translation

This was known within the cryptographic community

However, not within the systems community

66% of IPSec VPNs use a single 1024-bit prime

Are the groups used in practice still secure given this “new” information?

512-bit Keys and the Logjam Attack on TLS

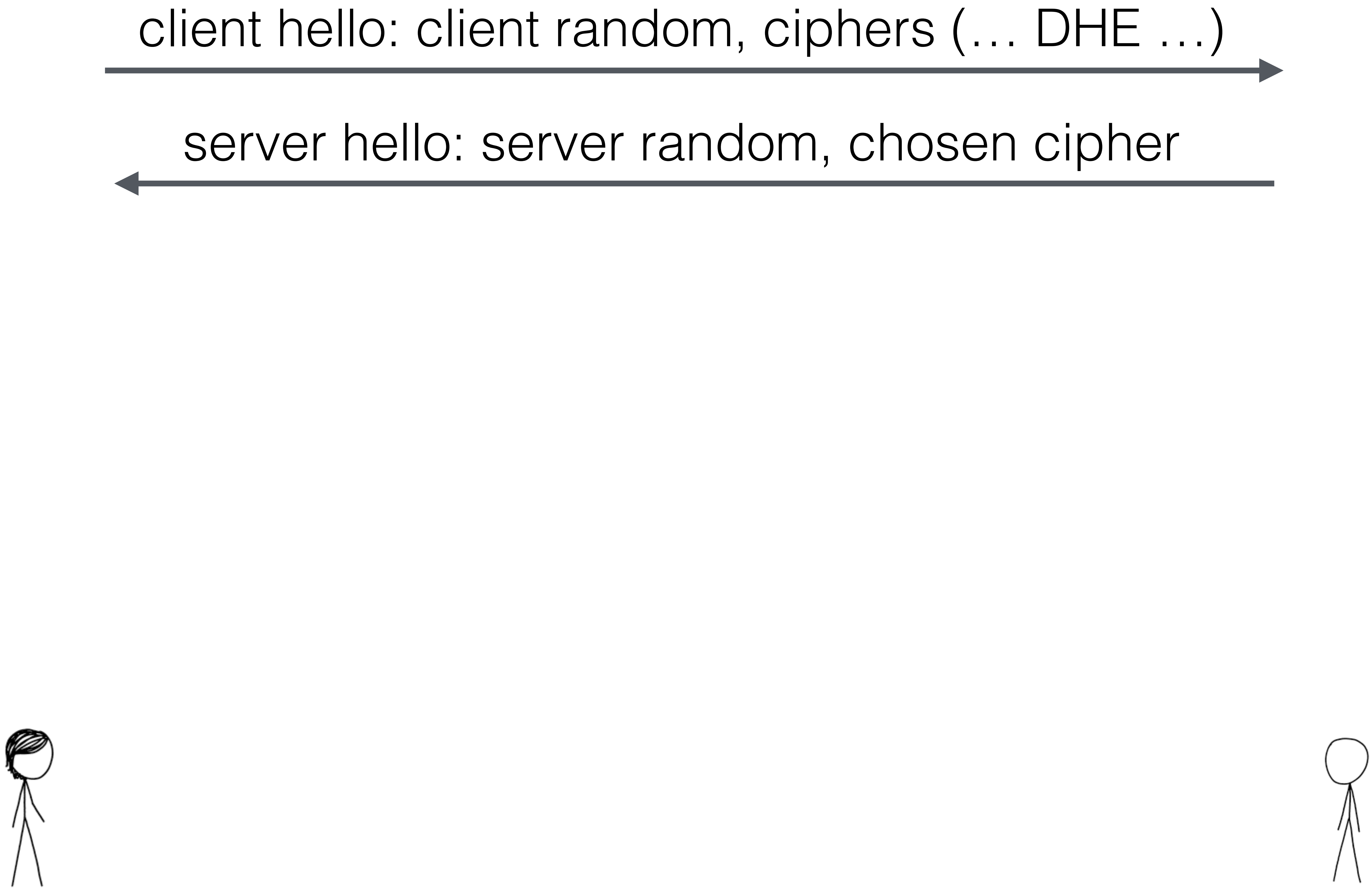
Diffie-Hellman in TLS

The majority of HTTPS websites use 1024-bit DH keys

However, nearly 8.5% of Top 1M still support *Export DHE*

Source	Popularity
Apache	82%
mod_ssl	10%
Other (463 distinct primes)	8%

Normal TLS Handshake



Normal TLS Handshake

client hello: client random, ciphers (… DHE …) →

← server hello: server random, chosen cipher

← certificate, p, g, g^a, $\text{Sign}_{\text{CertKey}}(p, g, g^a)$

g^b →

K_{ms}: KDF(g^{ab}, *client random, server random)*

Normal TLS Handshake

client hello: client random, ciphers (… DHE …) →

← server hello: server random, chosen cipher

← certificate, p, g, g^a, $\text{Sign}_{\text{CertKey}}$(p, g, g^a)

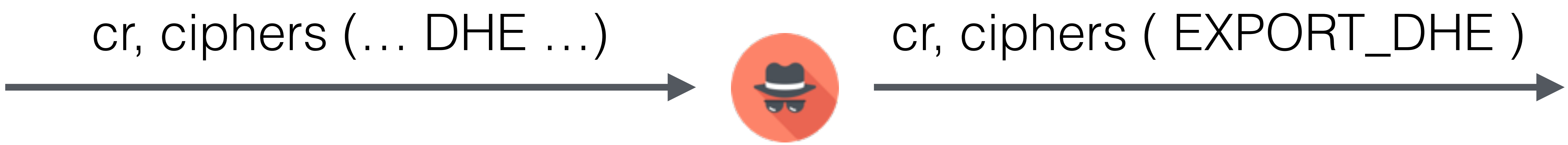
g^b →

K_{ms}: KDF(g^{ab}, *client random, server random*)

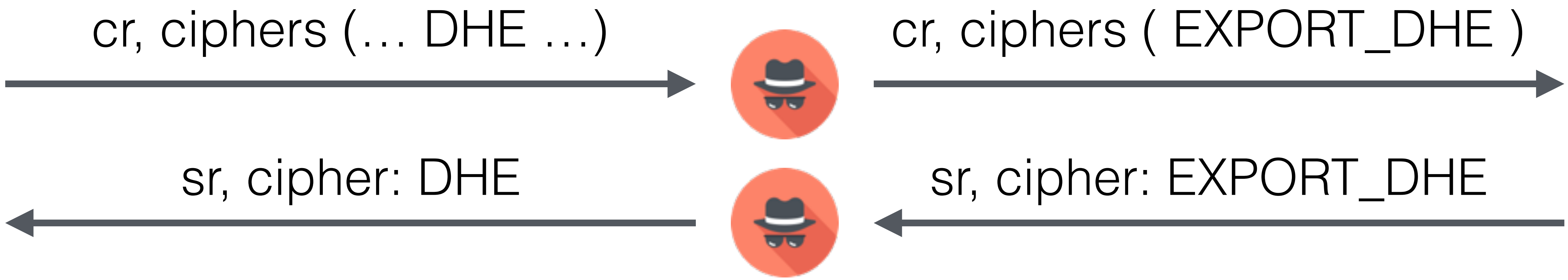
client finished: $\text{Sign}_{K_{ms}}$(Hash(m1 | m2 | …)) →

← server finished: $\text{Sign}_{K_{ms}}$(Hash(m1 | m2 | …))

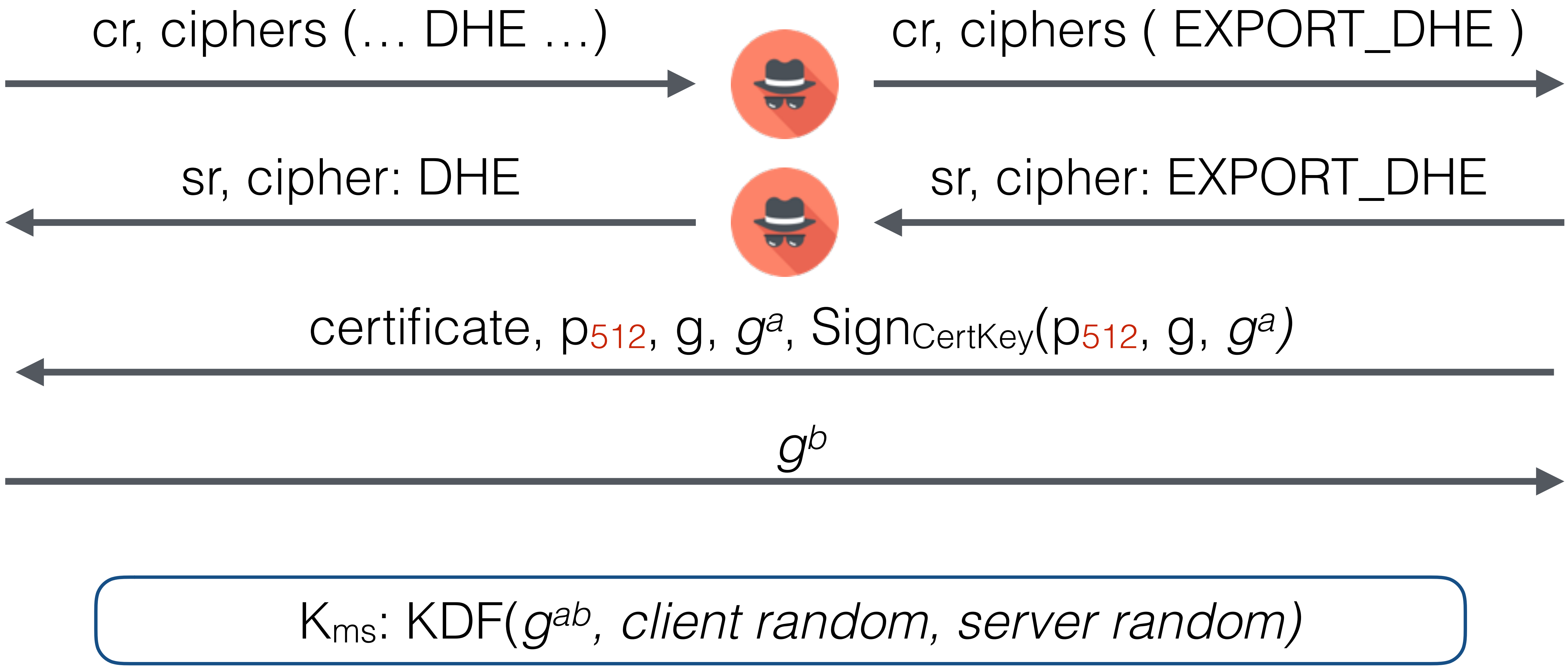
Logjam Attack



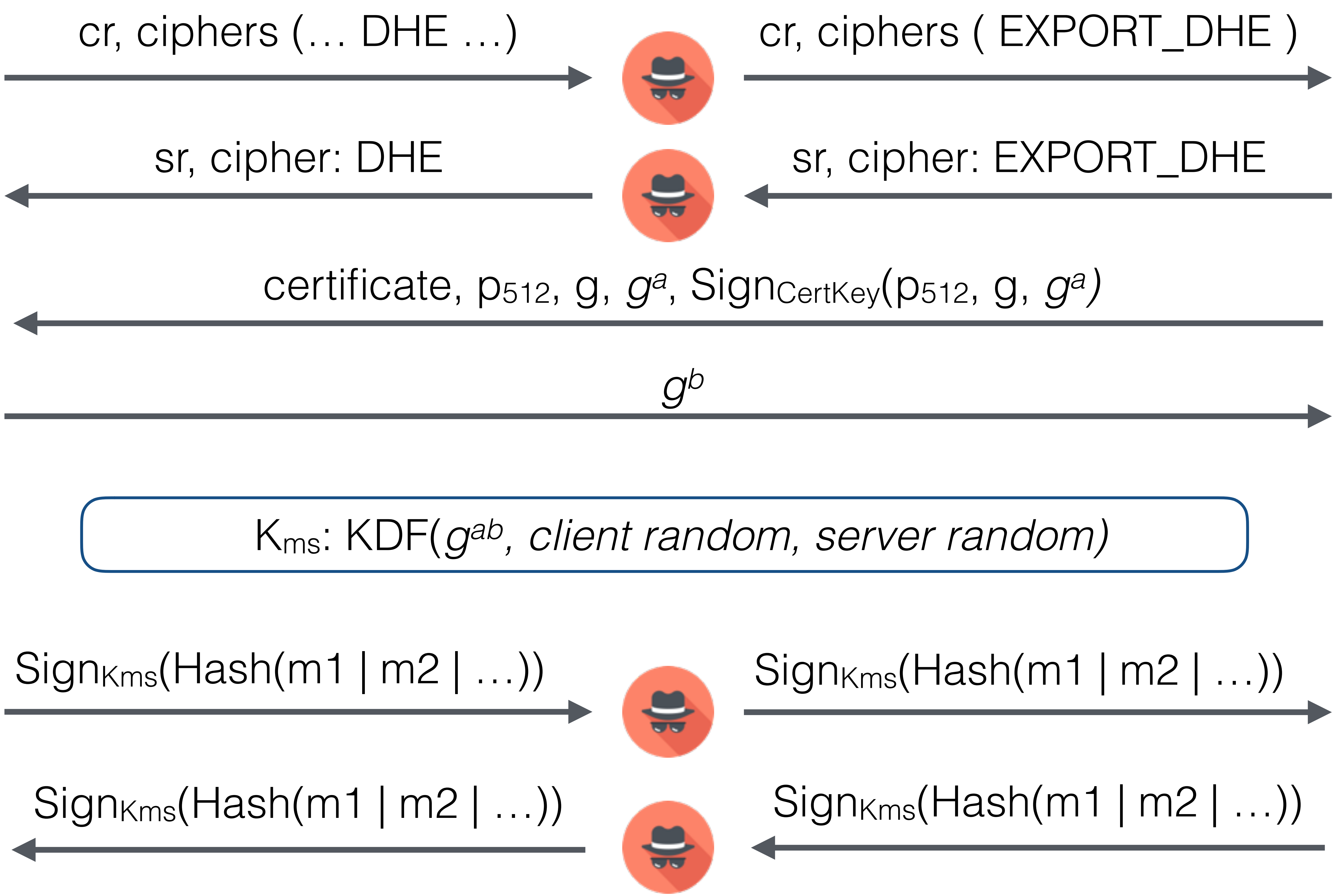
Logjam Attack



Logjam Attack



Logjam Attack



Computing 512-bit Discrete Logs

We modified CADO-NFS to compute two common primes

1 week pre-computation, individual log ~70 seconds

	polysel	sieving	linalg	descent
	2000-3000 cores		288 cores	36 cores
DH-512	3 hours	15 hours	120 hours	70 seconds

Logjam Mitigation

Browsers

- have raised minimum size to 768-bits
- ~~plan to move to 1024-bit in the future~~
- plan to drop all support for DHE

Server Operators

- Disable export ciphers!!
- ~~Use a 2048-bit or larger DHE key~~
- ~~If stuck using 1024-bit, generate a unique prime~~
- Moving to ECDHE

768- and 1024-bit Keys

Breaking One 1024-bit DH Key

Estimation process is convoluted due to the number of parameters that can be tuned.

Crude estimations based on asymptotic complexity:

	Sieving core-years	Linear Algebra core-years	Descent core-time
RSA-512	0.5	0.33	
DH-512	2.5	7.7	10 mins
RSA-768	800	100	
DH-768	8,000	28,500	2 days
RSA-1024	1,000,000	120,000	
DH-1024	10,000,000	35,000,000	30 days

Custom Hardware

If you went down this route, you would build ASICs

Prior work from Geiselmann and Steinwandt (2007) estimates ~80x speed up from custom hardware.

≈$100Ms of HW precomputes one 1024-bit prime/year

Custom Hardware

If you went down this route, you would build ASICs

Prior work from Geiselmann and Steinwandt (2007) estimates ~80x speed up from custom hardware.

≈$100Ms of HW precomputes one 1024-bit prime/year

For context… annual budgets for the U.S.

- Consolidated Cryptographic Program: 10.5B
- Cryptanalyic IT Services: 247M
- Cryptanalytic and exploitation services: 360M

Impact of Breaking a 1024-bit Key

Impact of Breaking Popular Keys

Computing one 1024-bit key (Oakley Group 2) would allow passively decrypting connections with:

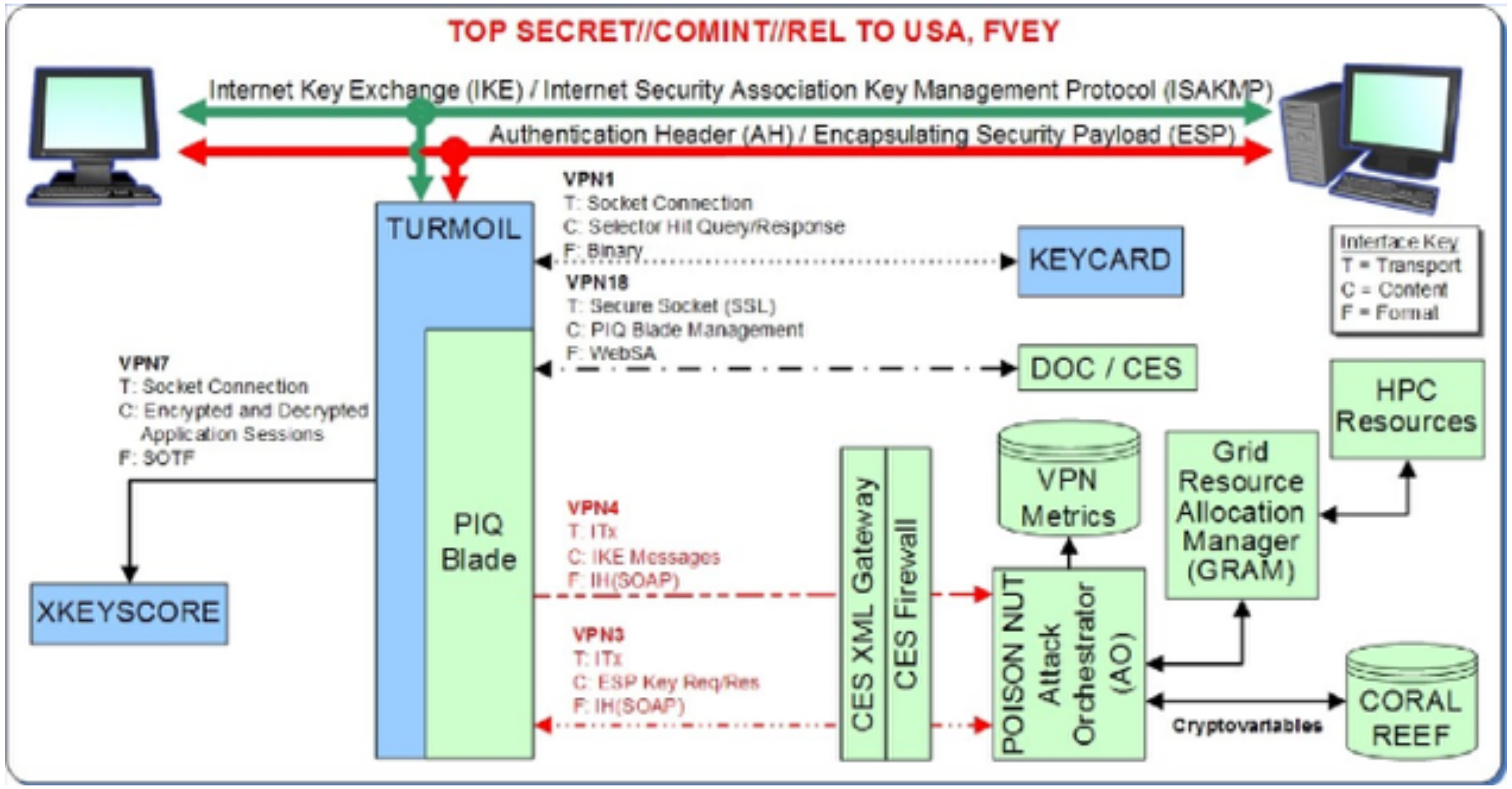
- 66% of IPSEC VPN servers
- 26% of SSH servers

The second most common prime (Apache):

- 18% of top 1 million websites
- 6.6% of all browser trusted websites

Is the NSA breaking DH Connections?

Plausibly. Our findings are consistent with the Snowden leaks on decrypting VPN traffic and within the NSA budget. However… speculative.



Uncovering Cryptographic Failures with Internet-Wide Measurement

Zakir Durumeric
University of Michigan
zakir@umich.edu