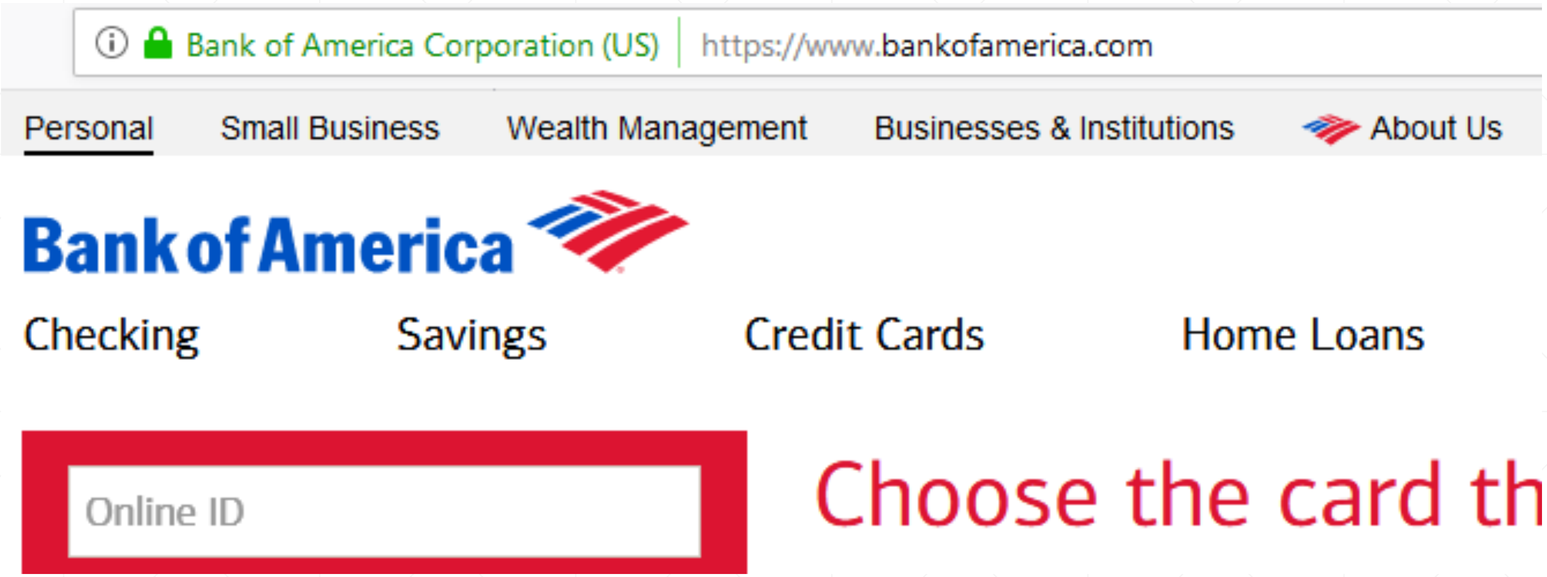
Coming of Age: A Longitudinal Study of TLS Deployment

Accepted at ACM Internet Measurement Conference (IMC) 2018, Boston, MA, USA

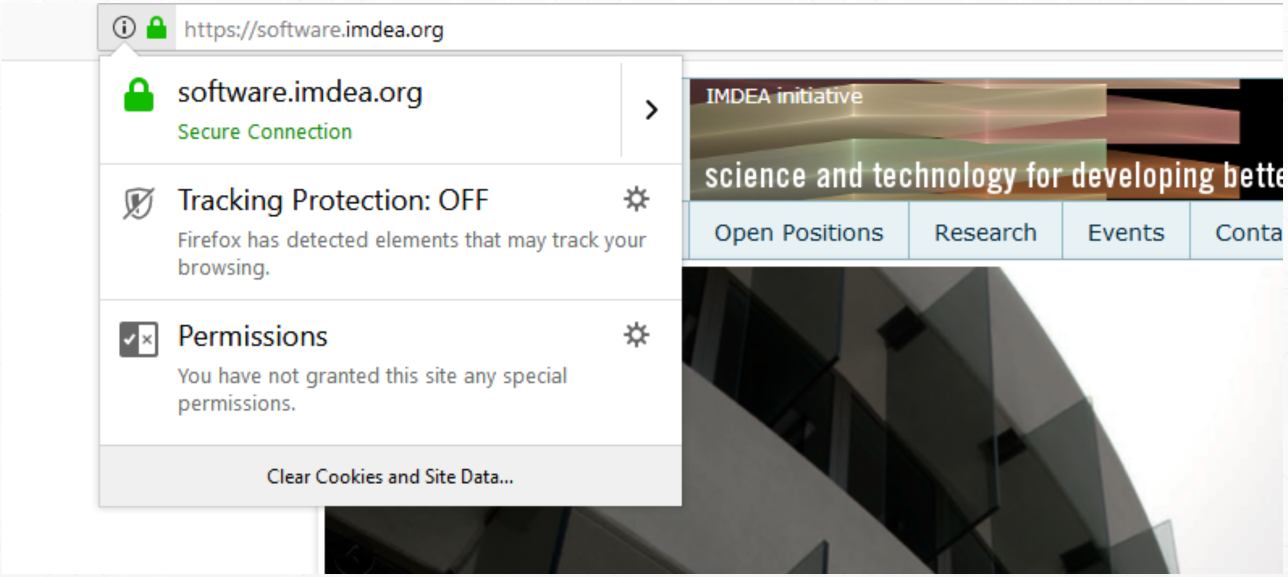
Platon Kotzias, Abbas Razaghpanah, Johanna Amann, Kenneth G. Paterson, Narseo Vallina-Rodriguez, Juan Caballero

TLS – The De Facto secure protocol



- ❑ Originally designed for secure e-commerce over HTTP

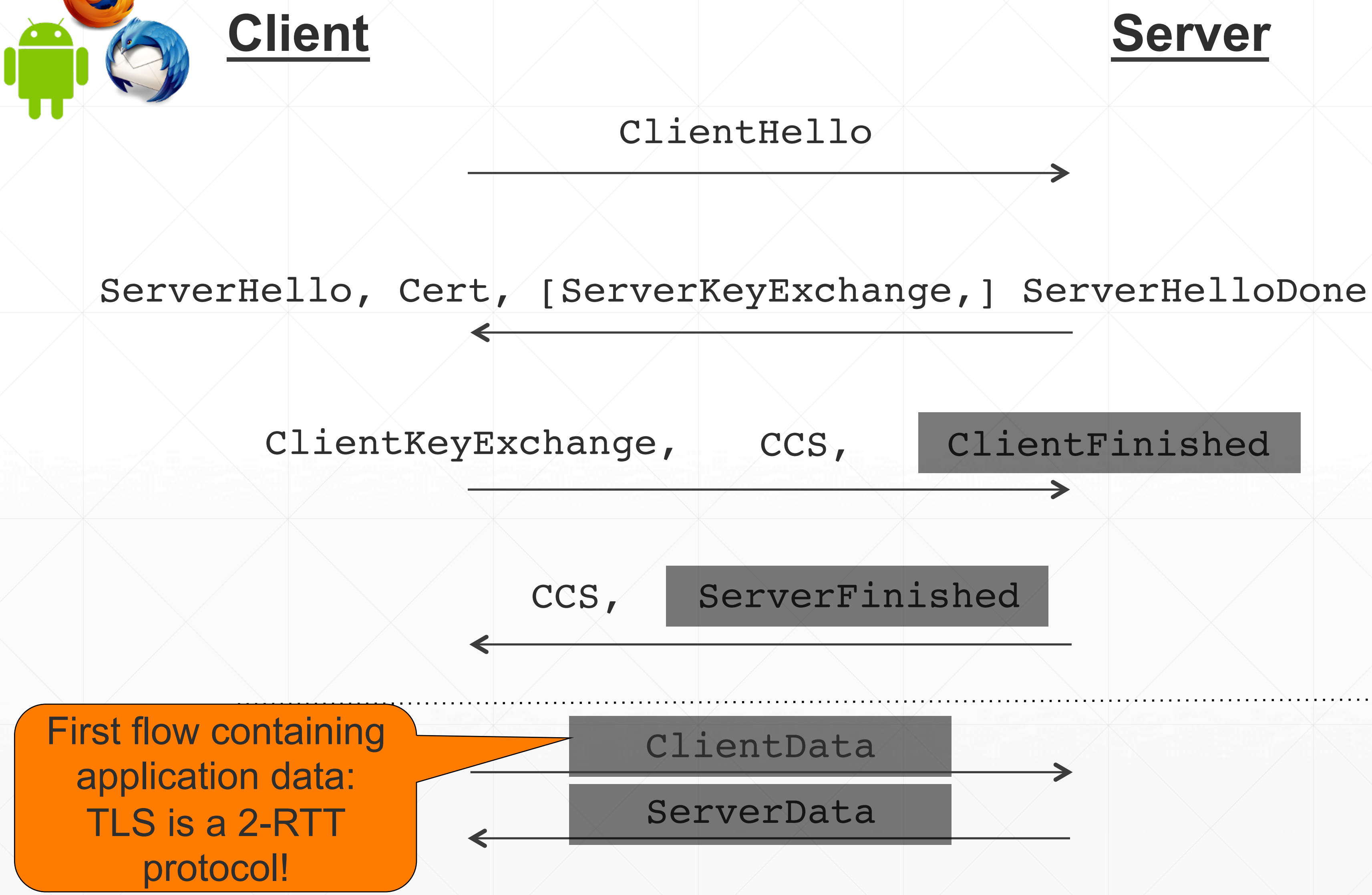
- ❑ Over **60%** of web traffic is now encrypted using TLS
- ❑ Hundreds of millions of people and devices every day



Firefox has detected elements that may track your browsing.

You have not granted this site any special permissions.

Full TLS Handshake Protocol



Client

Server

ClientHello

ServerHello, Cert, [ServerKeyExchange,] ServerHelloDone

ClientKeyExchange, CCS, ClientFinished

CCS, ServerFinished

ClientData

ServerData

First flow containing application data: TLS is a 2-RTT protocol!

TLS Handshake Protocol - Goals

- ❑ Agree on the shared master secret that will be used to protect the session

- ❑ Provides authentication of server (usually) and client (rarely)
  - ❑ Using public key cryptography supported by digital certificates

- ❑ Protects negotiation of all cryptographic parameters.
  - ❑ SSL/TLS version number, encryption and hash algorithms, authentication and key establishment methods.
  - ❑ To prevent version rollback and cipher suite downgrade attacks.

TLS Attacks

Crypto primitives

- RSA, DSA, ECDSA
- Diffie-Hellman, ECDH
- HMAC
- MD5, SHA-1, SHA-2
- DES, 3DES, RC4, AES

Ciphersuite details

- Data structures
- Key derivation
- Encryption modes, IVs
- Padding
- Compression

Protocol “framework”

- Alerts & errors
- Certification/revocation
- Negotiation
- Renegotiation
- Session resumption

Libraries

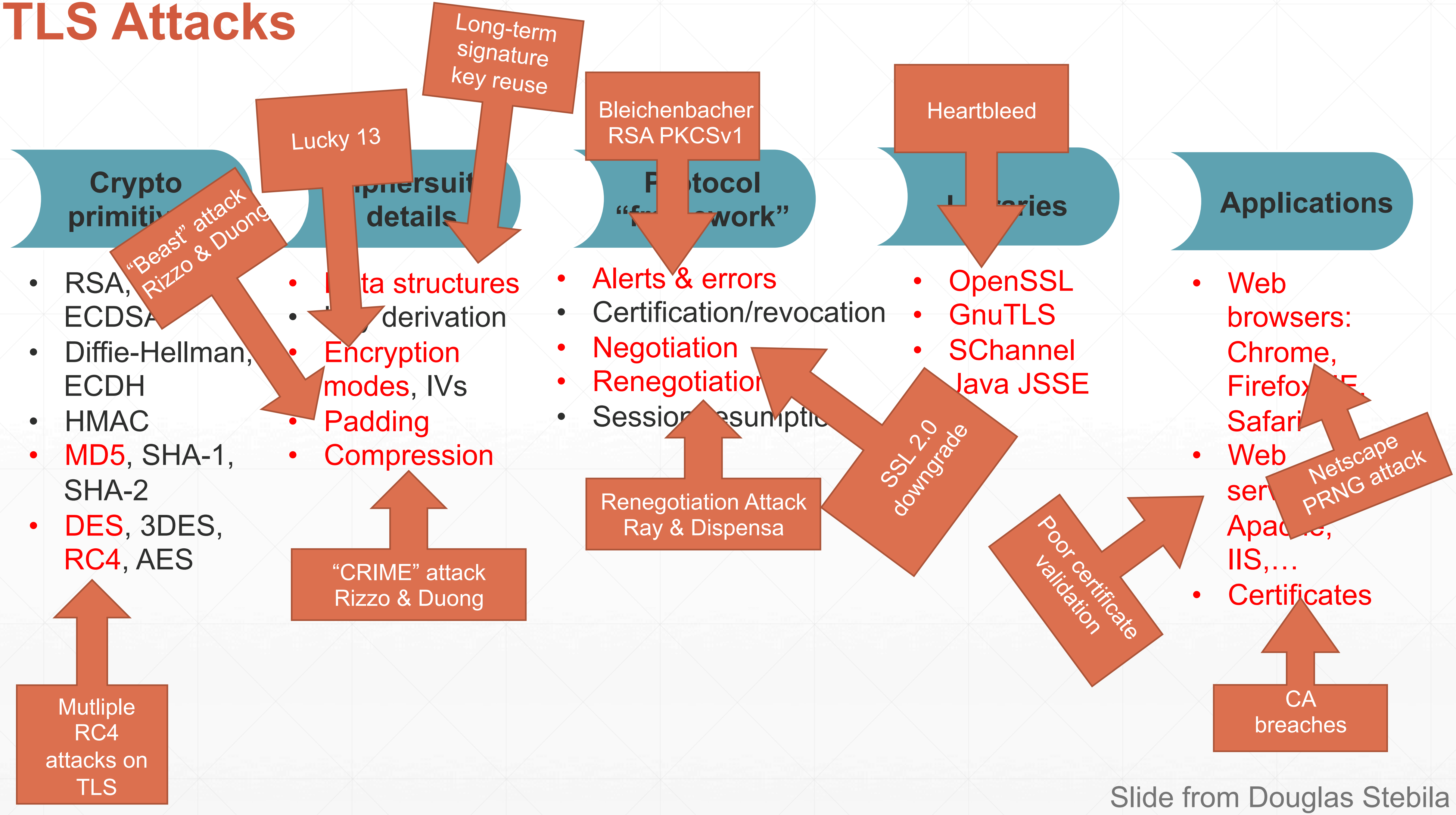
- OpenSSL
- GnuTLS
- SChannel
- Java JSSE

Applications

- Web browsers: Chrome, Firefox, IE, Safari
- Web servers: Apache, IIS,…
- Certificates

Slide from Douglas Stebila

TLS Attacks



Crypto primitives

- RSA, ECDSA
- Diffie-Hellman, ECDH
- HMAC
- MD5, SHA-1, SHA-2
- DES, 3DES, RC4, AES

Ciphersuite details

- Data structures
- Key derivation
- Encryption modes, IVs
- Padding
- Compression

Protocol “framework”

- Alerts & errors
- Certification/revocation
- Negotiation
- Renegotiation
- Session resumption

Libraries

- OpenSSL
- GnuTLS
- SChannel
- Java JSSE

Applications

- Web browsers: Chrome, Firefox, IE, Safari
- Web servers: Apache, IIS,…
- Certificates

Long-term signature key reuse
Lucky 13
“Beast” attack Rizzo & Duong
Bleichenbacher RSA PKCSv1
Heartbleed
Mutliple RC4 attacks on TLS
“CRIME” attack Rizzo & Duong
Renegotiation Attack Ray & Dispensa
SSL 2.0 downgrade
Poor certificate validation
Netscape PRNG attack
CA breaches

Slide from Douglas Stebila

Contributions

- Large scale longitudinal study on TLS ecosystem **since 2012 using 319.3B TLS connections**
- Analyze trends and evolution of the TLS ecosystem both on the client and server side
- Special focus on changes occurring in response to specific high-profile attacks
- Create the largest database of TLS client fingerprints to-date to identify the evolution of client software on the Internet

github.com/platonK/tls_fingerprints

Datasets

ICSI SSL Notary

❑ Metadata from **319.3B** outgoing SSL/TLS connections

❑ **6 years** (Feb 2012 – March 2018)

❑ Universities and research institutions from North America

❑ Collection using Bro Network Security Monitor (now Zeek)

censys

❑ Periodic Internet-wide TLS scans

❑ **~3 years** (Aug 2015 – May 2018)

❑ Scanning using mimicking a 2015 version of Chrome

❑ Temporal view of publicly-reachable TLS servers

Road Map

Intro

TLS Fingerprints

Vulnerability Analysis

Ecosystem Improvements

Identifying Client Software – Building TLS Fingerprints

{ Cipher Suites | Extensions | Supported EC | Supported EC Point Formats }

Each TLS fingerprint maps to a program/library and the version range that it covers

- ❑ 200 cipher suites, 28 extensions, and 35 elliptic curves values
- ❑ Build a groundtruth of **1,684** TLS fingerprints:
  - ❑ Browserstack service for browser and mobile devices
  - ❑ Compile TLS libraries
  - ❑ Prior work

OpenSSL
Cryptography and SSL/TLS Toolkit

Browsers

Libraries

Email Clients

Android
SDK & Apps

Identifying Client Software – Matching TLS Traffic

- ❑ Apply TLS fingerprints to 191,9B (60%) of TLS connections after February 2014

69,874 unique TLS Fingerprints

1,670 matched

191,9B TLS connections

~70% matched

- ❑ 1,203 fingerprints responsible for **~22% of the connections** seen for more than 1,200 days

22% of the TLS connections initiated by software that has not updated their supported ciphersuites since 2014

Road Map

Intro

TLS Fingerprints

Vulnerability Analysis

Ecosystem Improvements

SSL/TLS versions

Version	Release Date
SSL 2	Feb. 1995
SSL 3	Nov. 1996
TLS 1.0	Jan. 1999
TLS 1.1	Apr. 2006
TLS 1.2	Aug. 2008
TLS 1.3	Aug. 2018

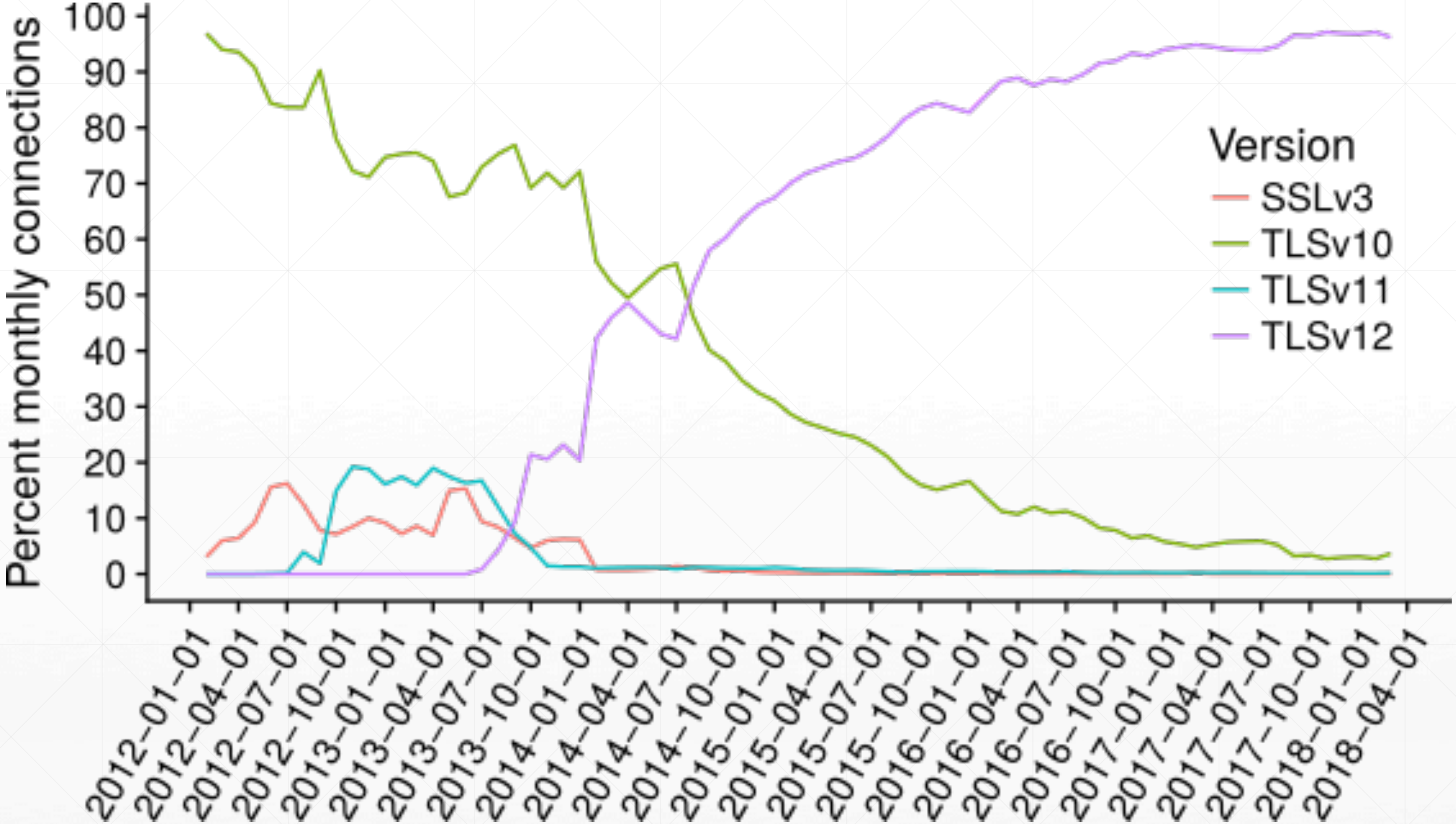
Considered insecure

- PCI council suggests migration from TLSv1.1 to newer versions (before June 2018)

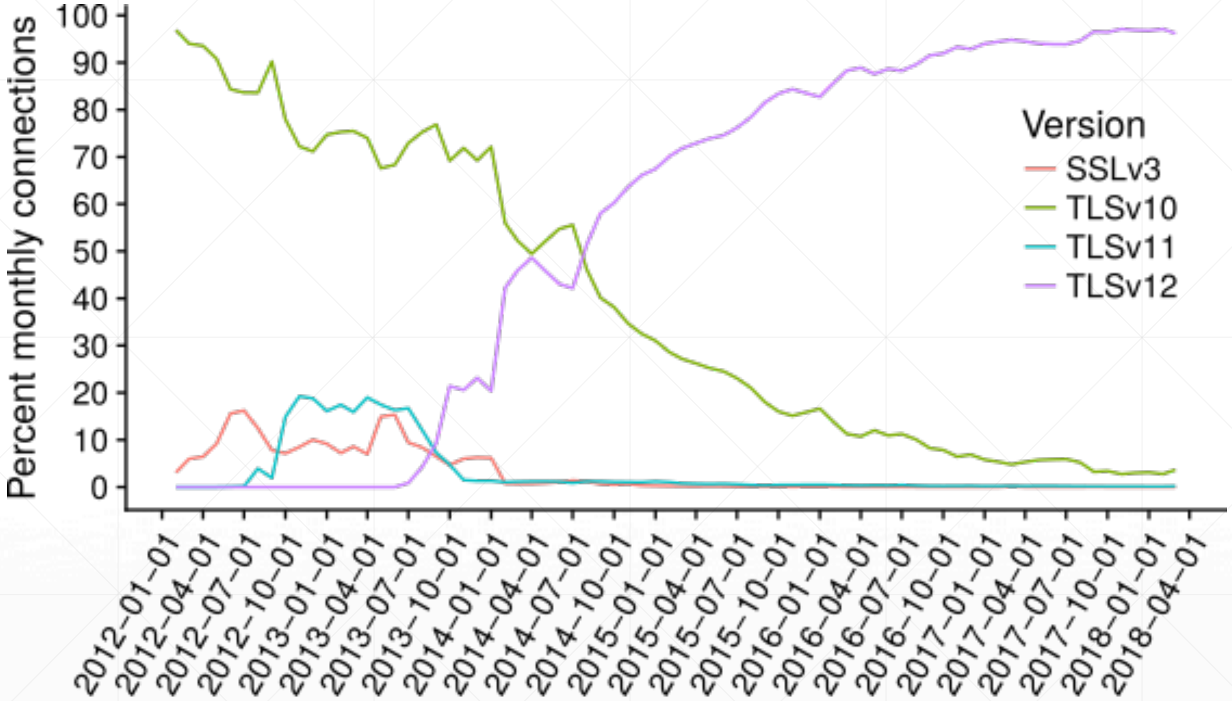
- Main options prior to TLS 1.2:
  - HMAC-then-CBC with DES, 3-DES, AES
  - HMAC-then-RC4

- Support for AEAD algorithms added in TLS 1.2:
  - AES-GCM (2x faster than CBC mode)
  - AES-CCM
  - Chacha20-Poly1305

SSL/TLS Negotiated versions



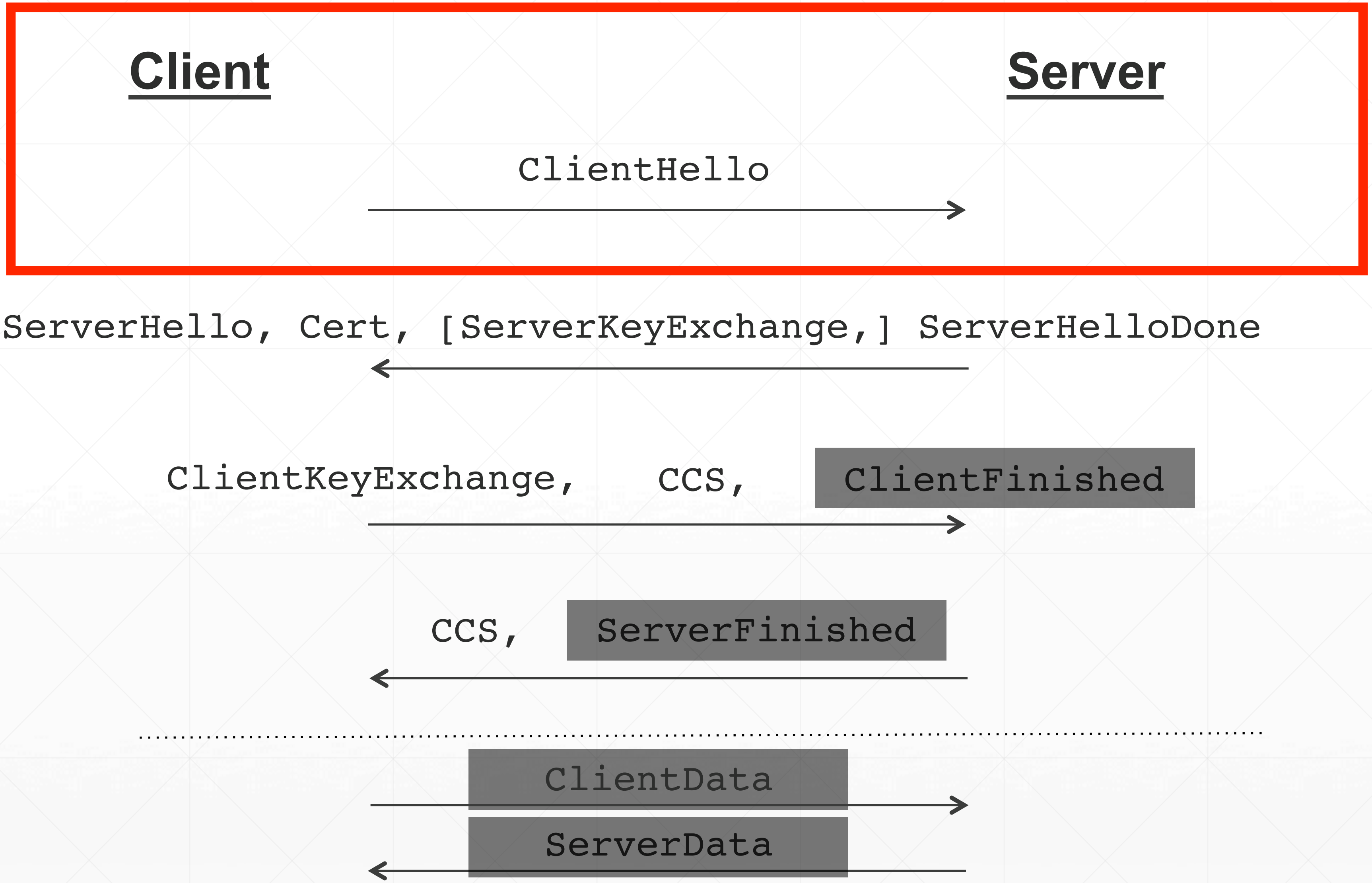
SSL/TLS Negotiated versions



- ❑ Big uptake in TLS 1.2 starting in late 2013
  - ❑ 5 years after it was standardized
- ❑ Almost no SSLv2 (1.2k connections in Feb. 2018).
- ❑ 360.1K SSLv3 connections in Feb. 2018 to 1789 different servers.
- ❑ 4 servers received more than 50,000 SSLv3 connections; all belong to Symantec and Wayport.

- ❑ Less than 25% of servers support SSLv3 (May 2018).

SSL/TLS Record Protocol Algorithms Advertised by Clients



Client

Server

ClientHello

ServerHello, Cert, [ServerKeyExchange,] ServerHelloDone

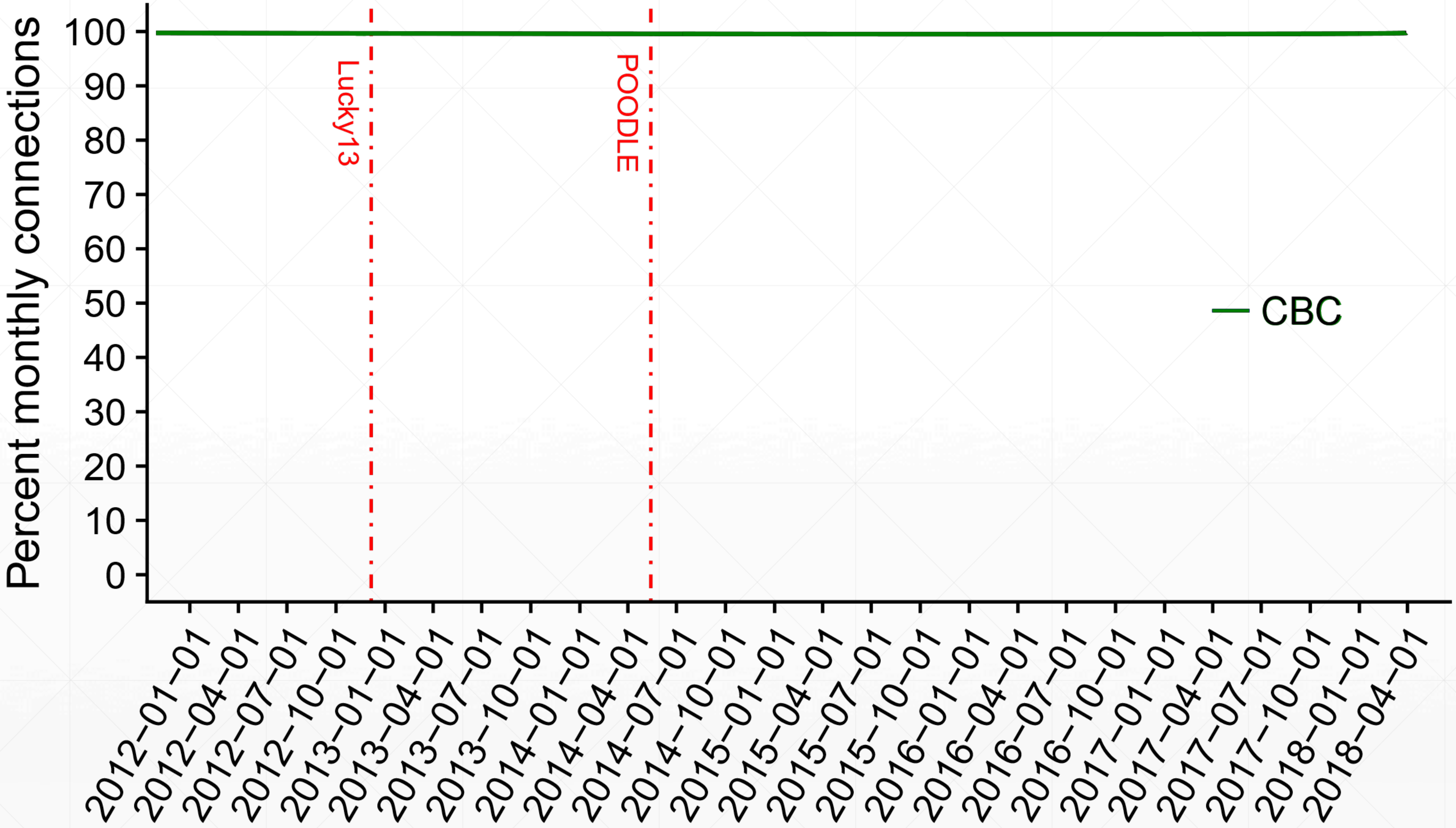
ClientKeyExchange, CCS, ClientFinished

CCS, ServerFinished

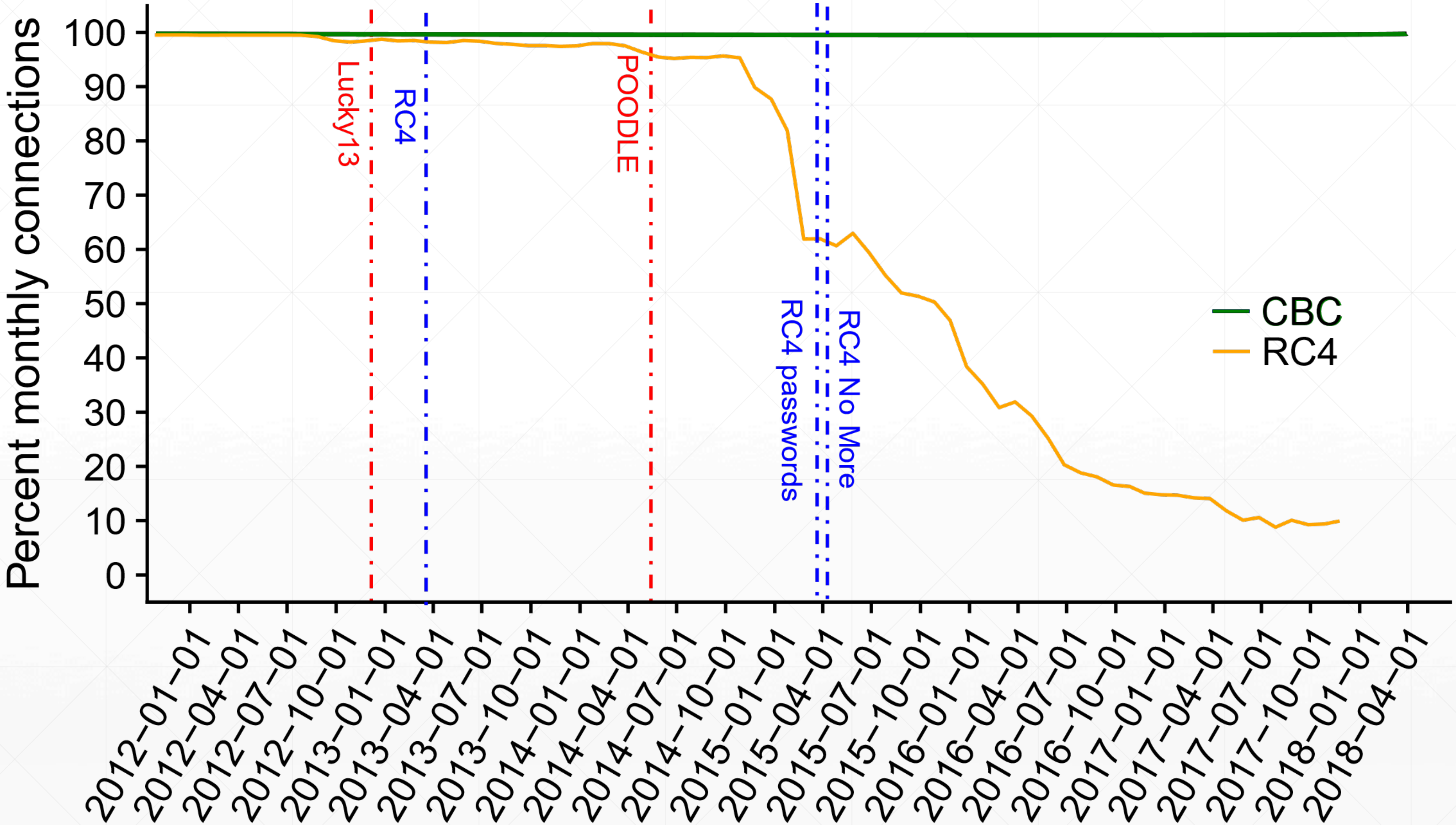
ClientData

ServerData

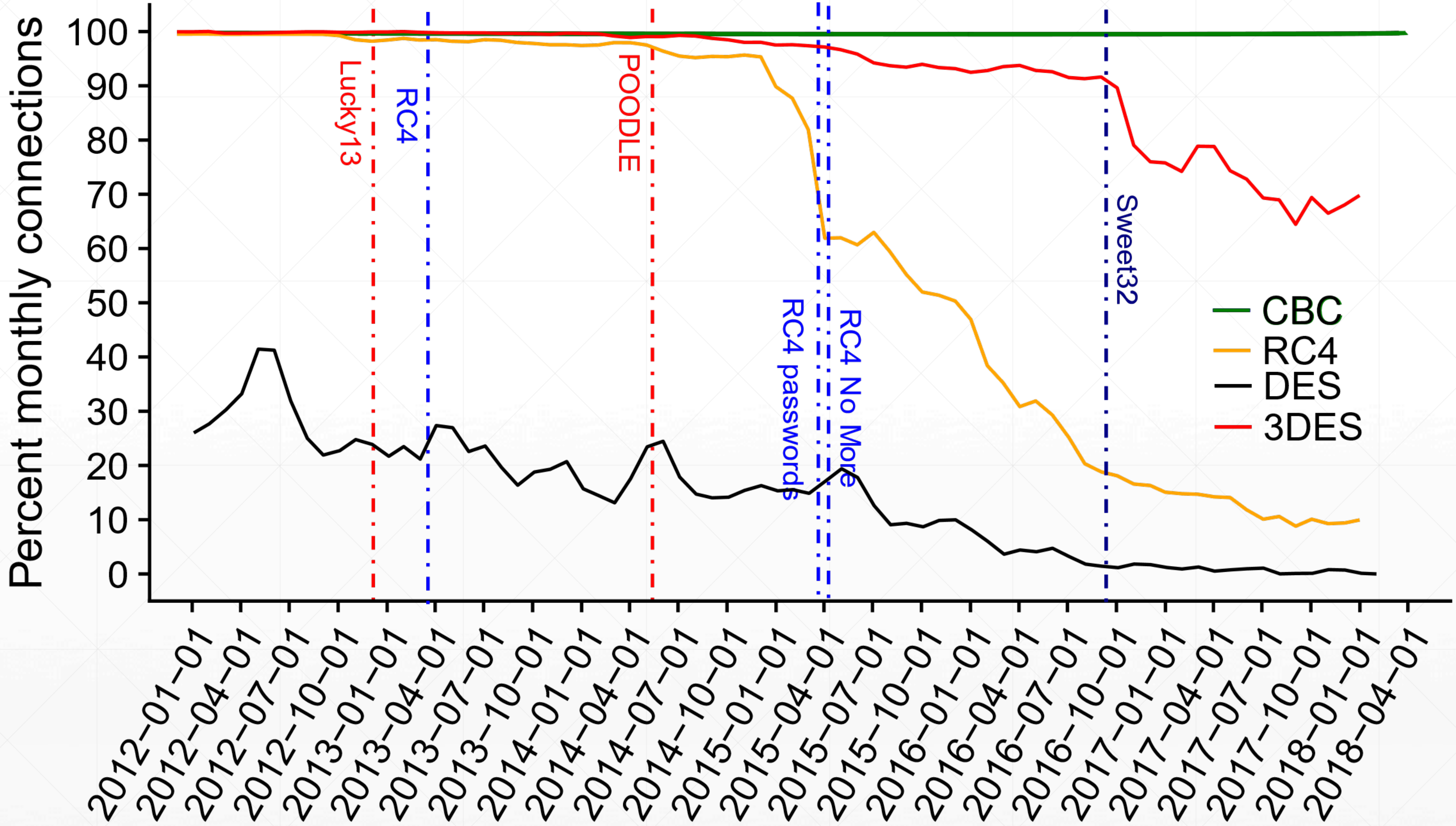
SSL/TLS Record Protocol Algorithms Advertised by Clients



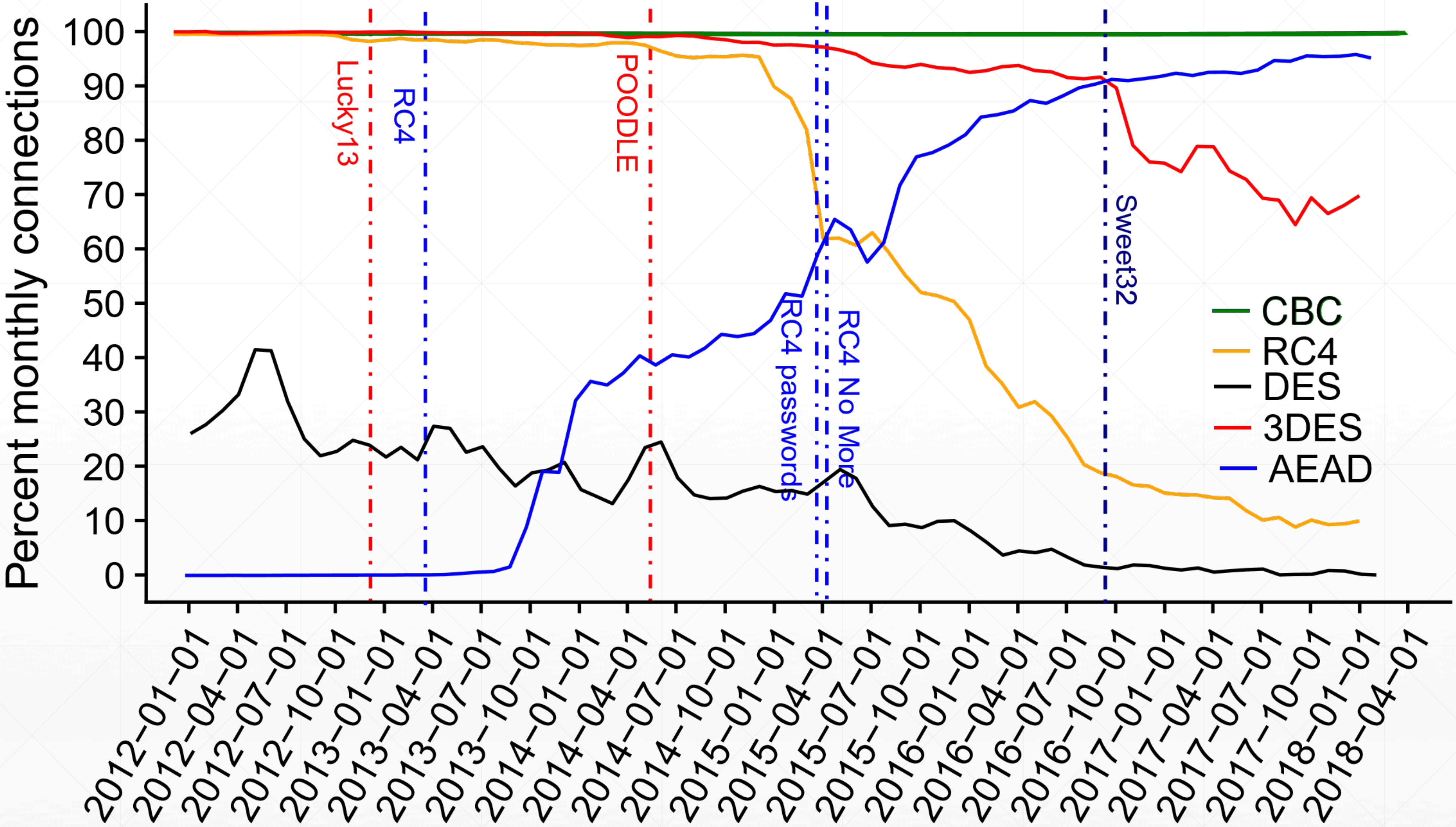
SSL/TLS Record Protocol Algorithms Advertised by Clients



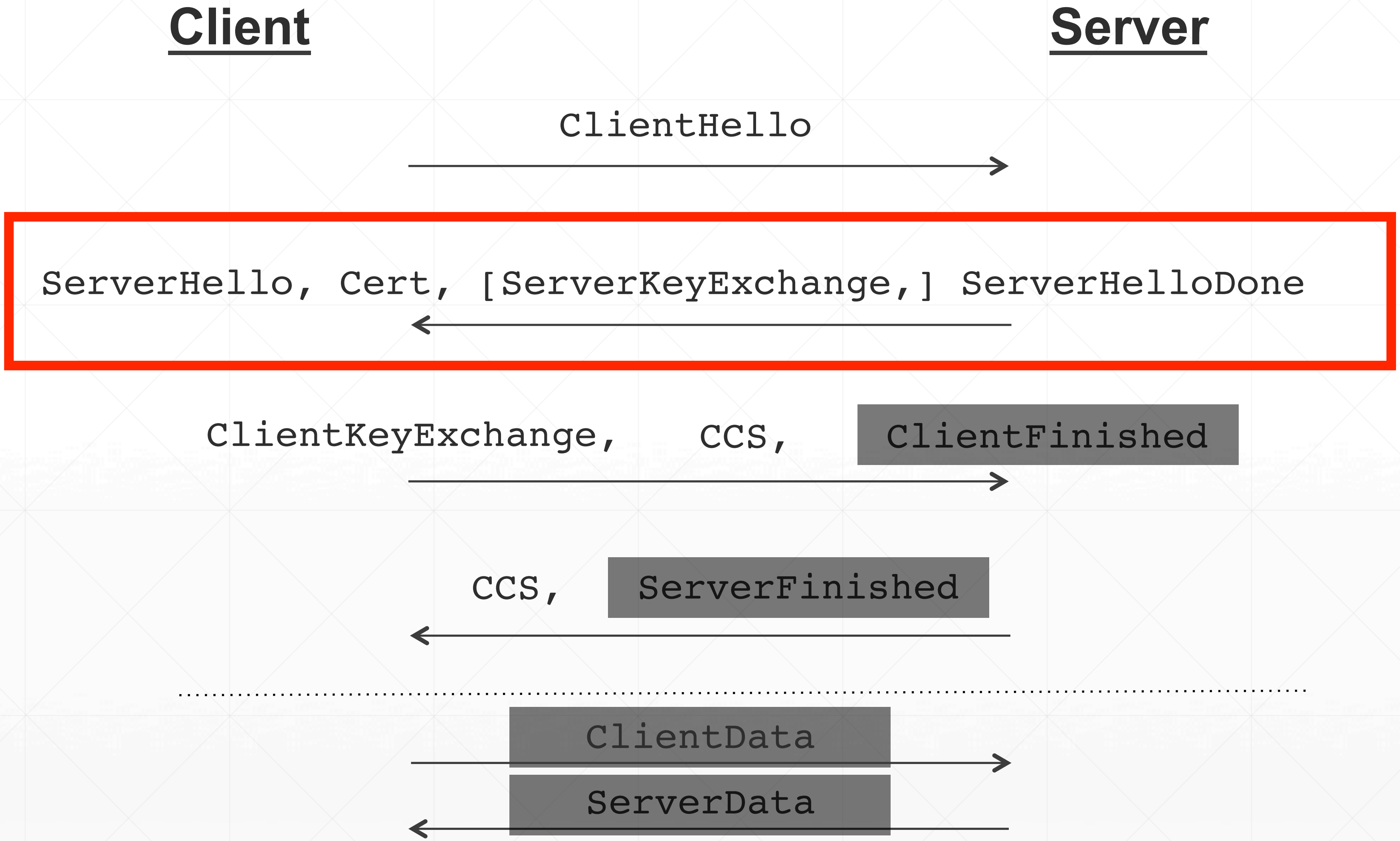
SSL/TLS Record Protocol Algorithms Advertised by Clients



SSL/TLS Record Protocol Algorithms Advertised by Clients



SSL/TLS Record Protocol Algorithms in Use



Client

Server

ClientHello

ServerHello, Cert, [ServerKeyExchange,] ServerHelloDone

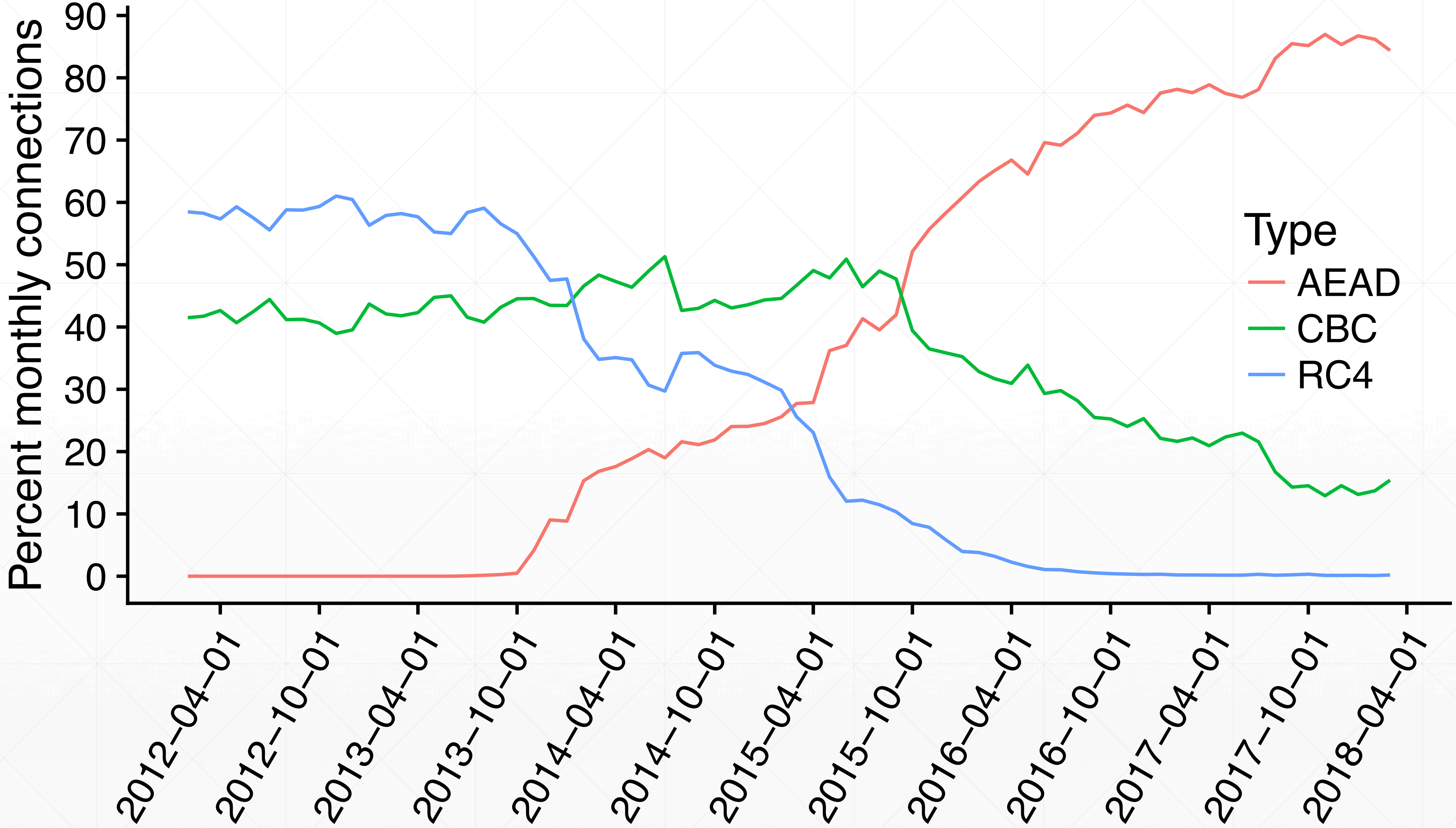
ClientKeyExchange, CCS, ClientFinished

CCS, ServerFinished

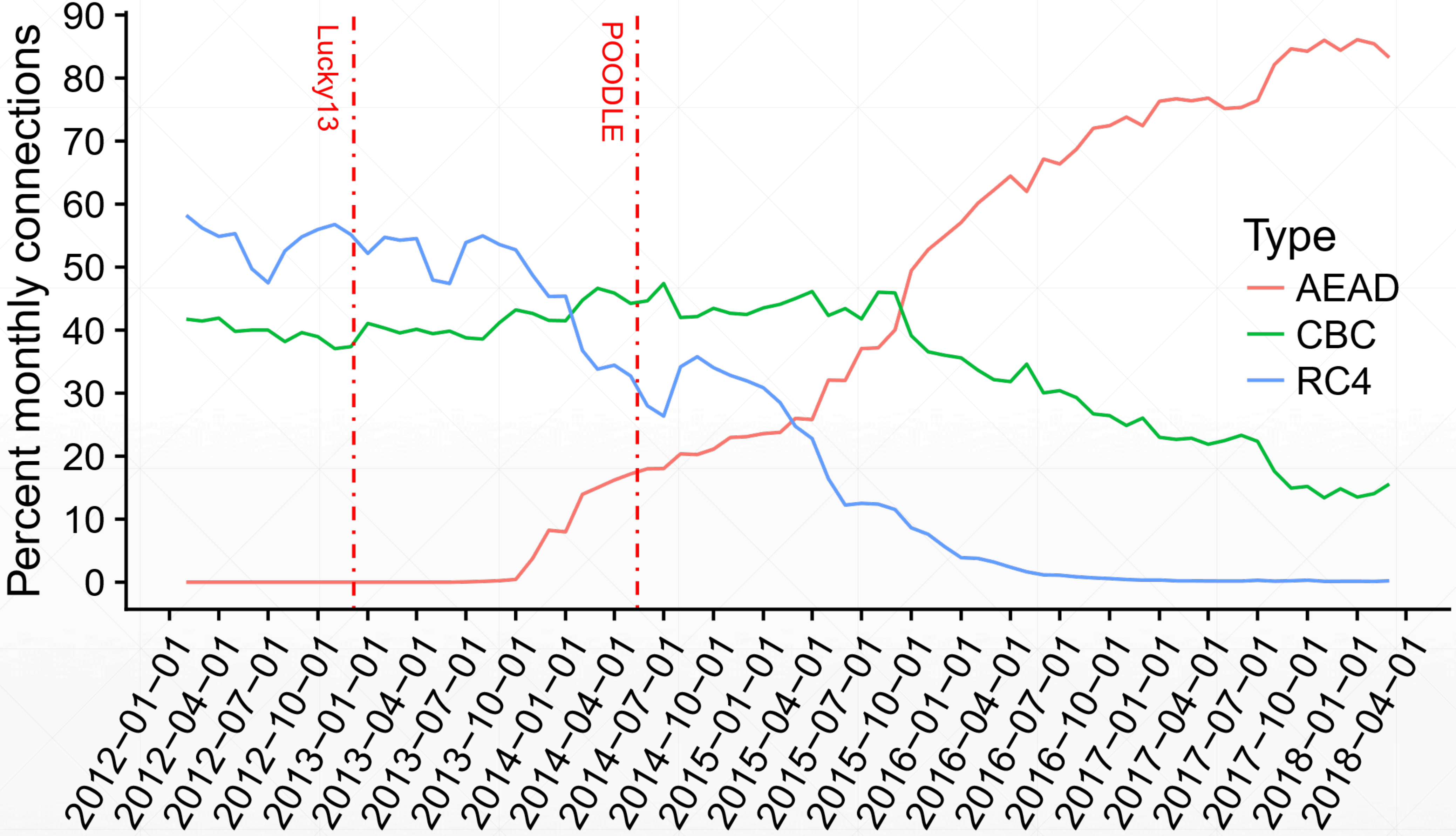
ClientData

ServerData

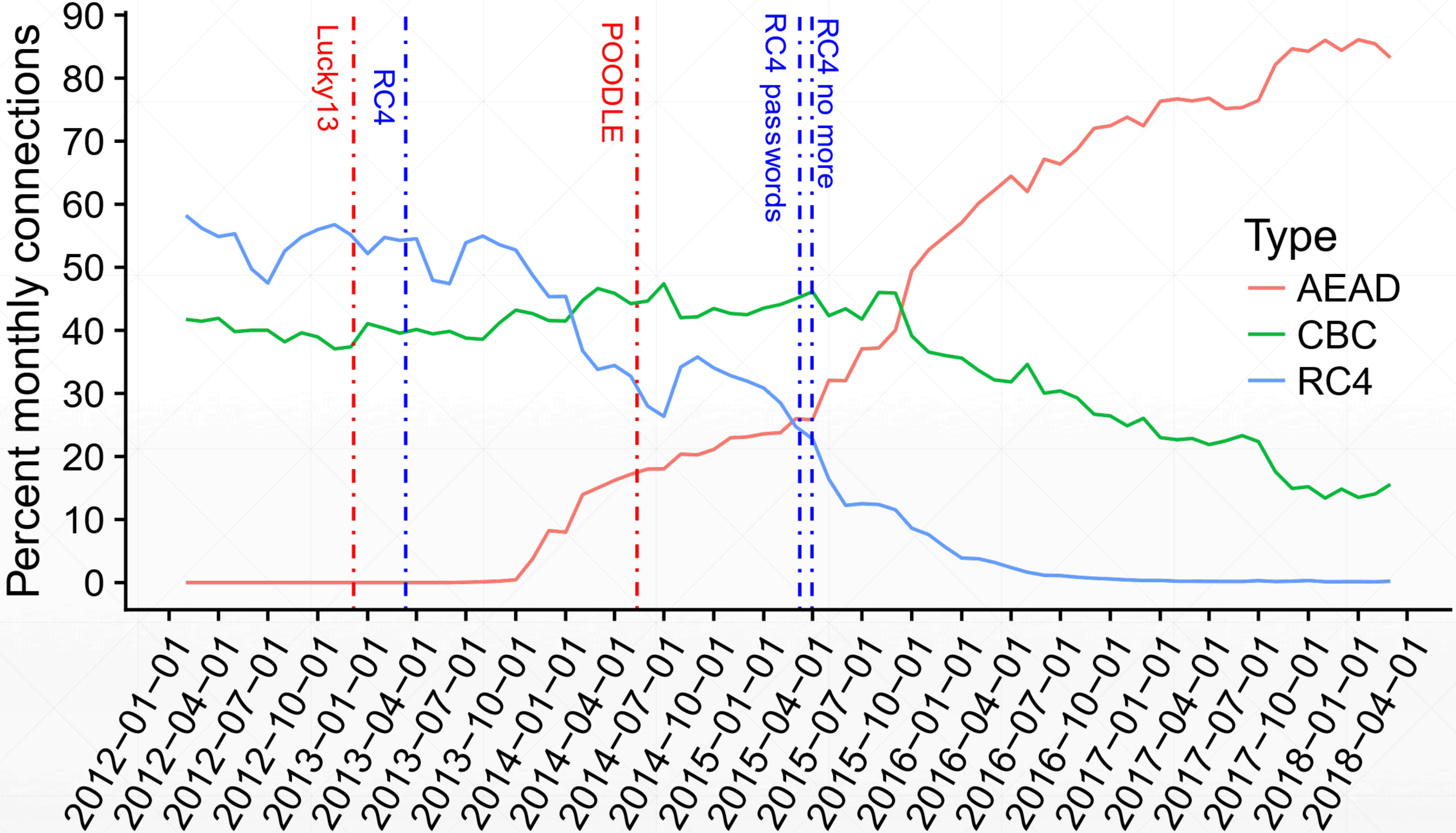
SSL/TLS Record Protocol Algorithms in Use



SSL/TLS Record Protocol Algorithms in Use

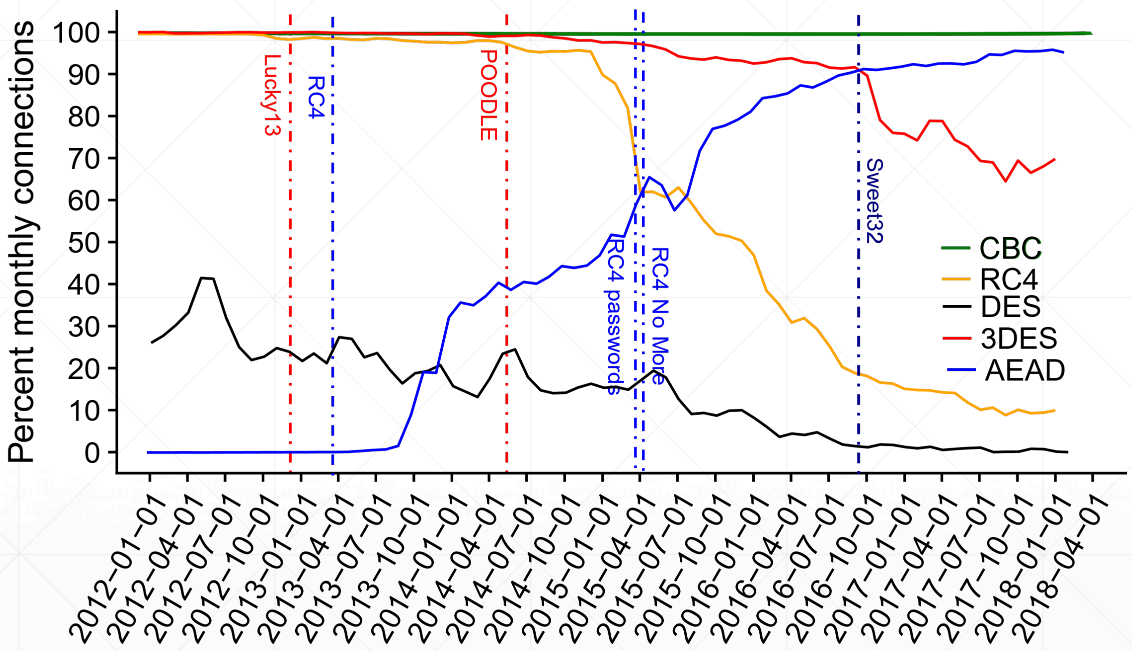


SSL/TLS Record Protocol Algorithms in Use

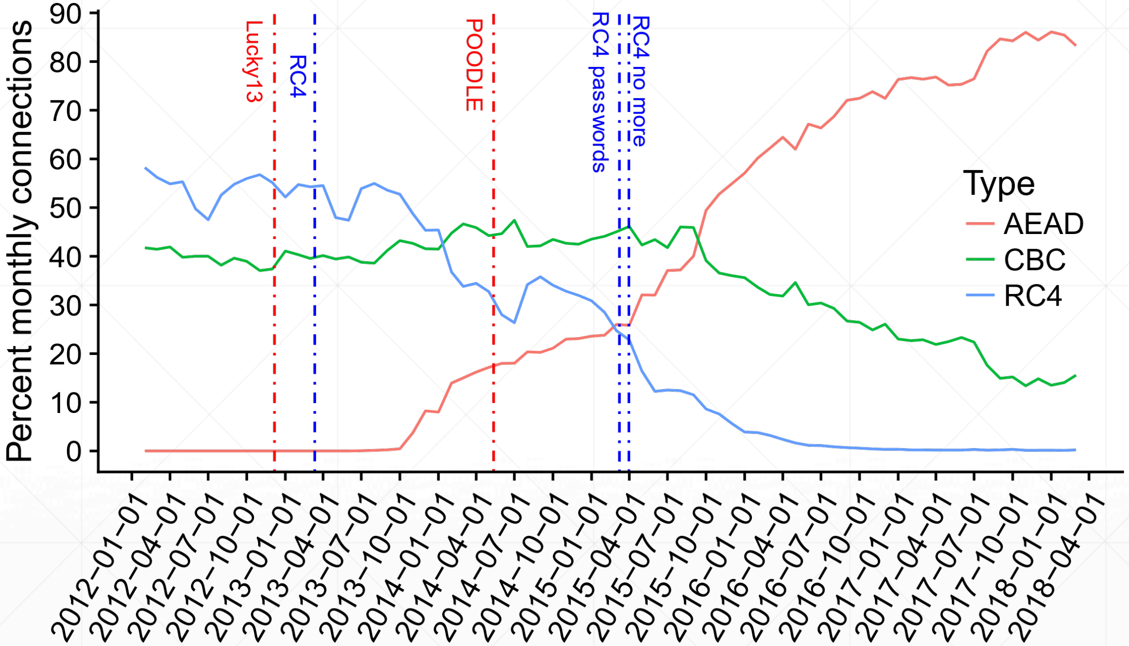


Algorithms advertised vs algorithms used

Advertised



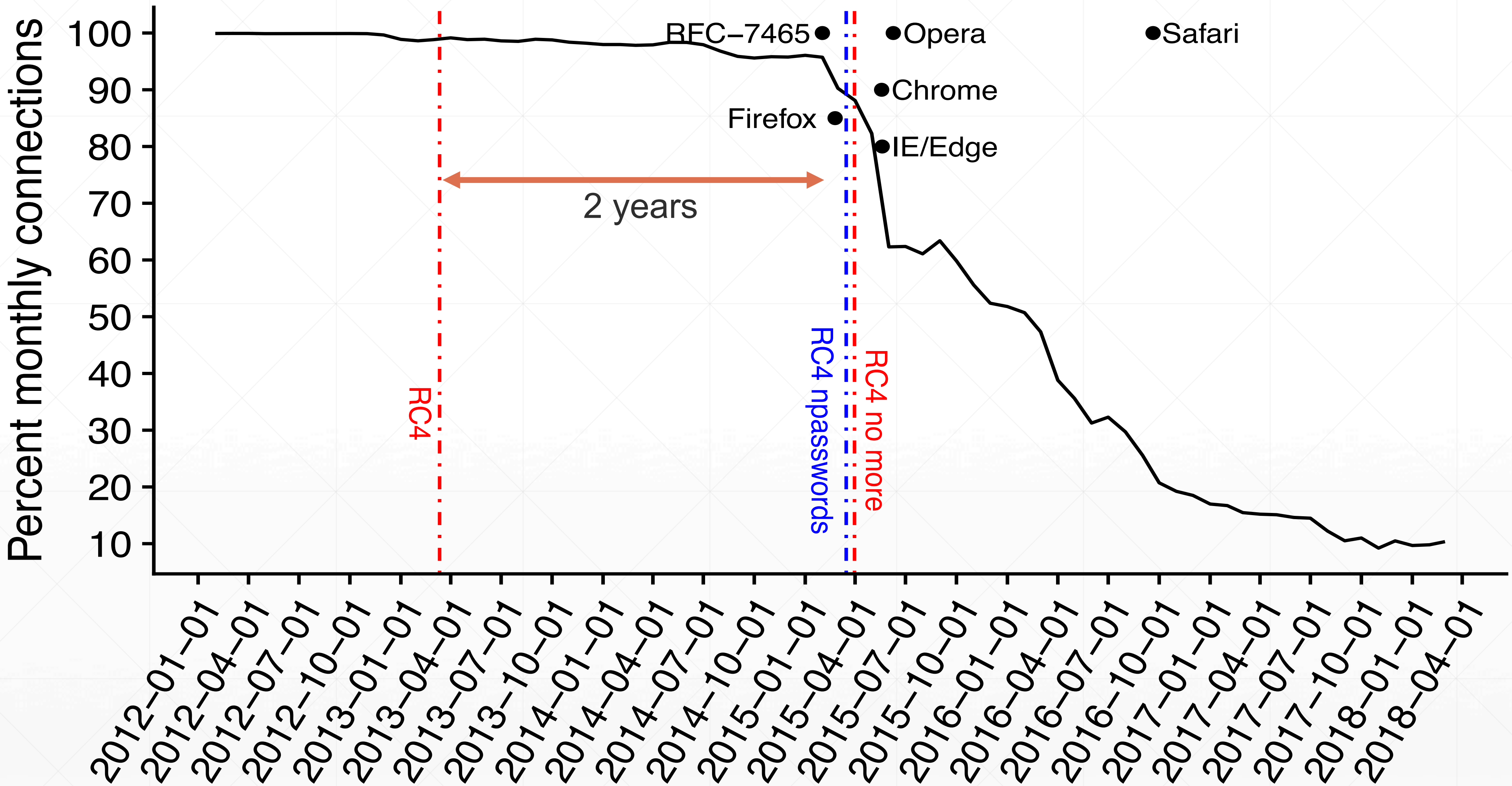
Used



Changes are driven by server-side updates

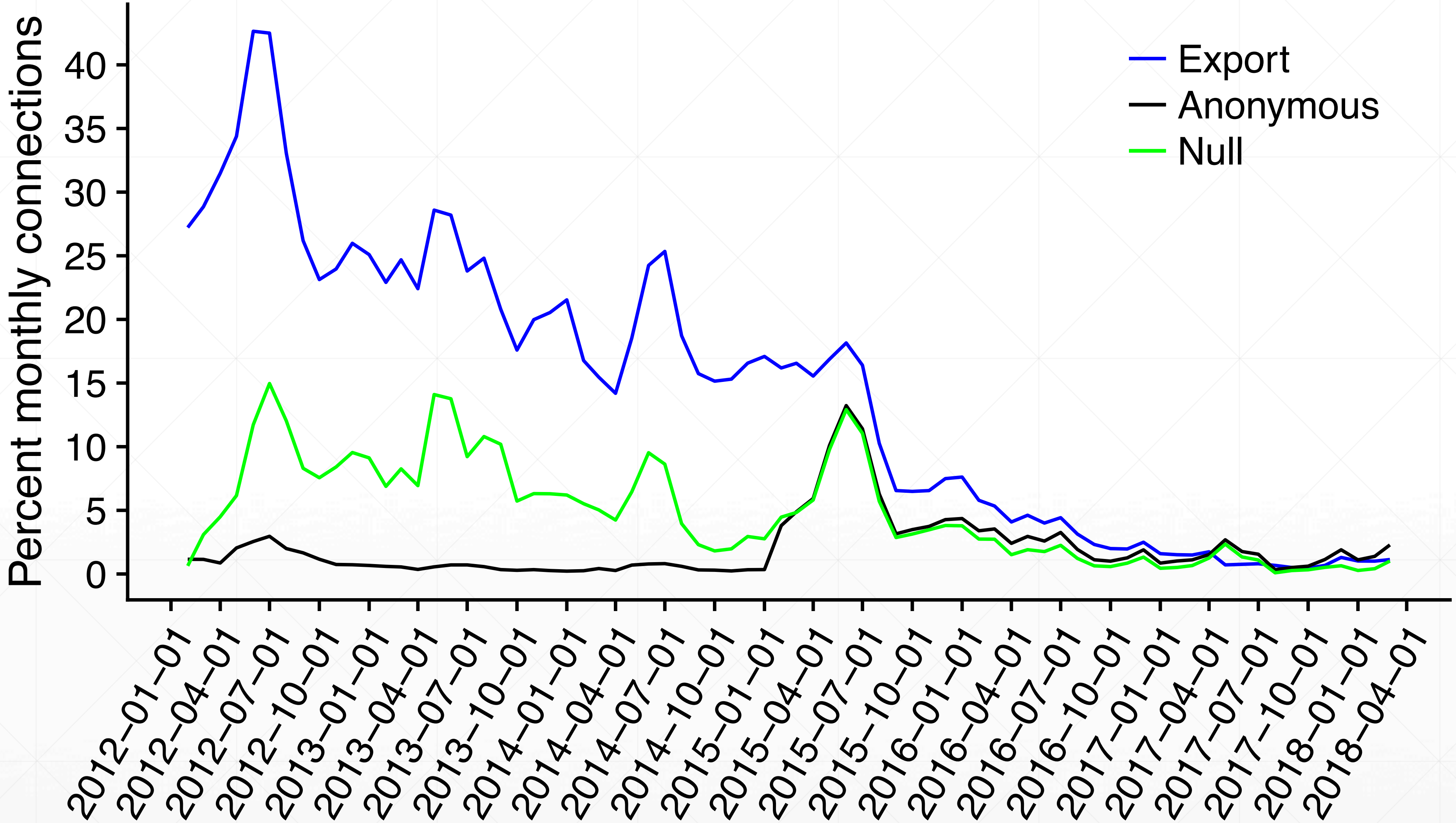
Clients are slow to drop support for older algorithms

Clients Offering RC4



Browsers are the first to drop support of RC4 but still they are slow

Advertised Export, Null, and Anonymous Ciphers

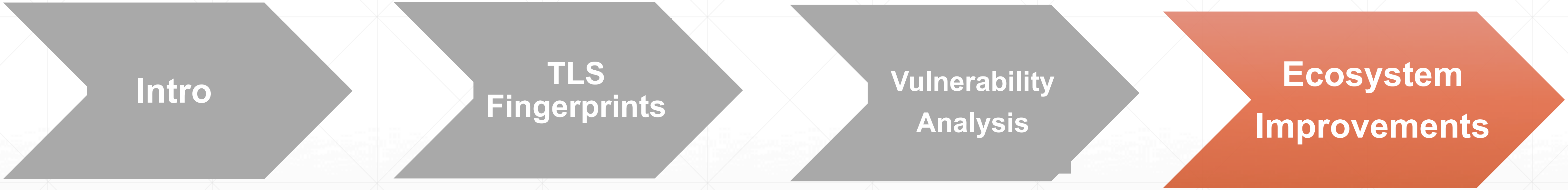


Export: typically 40-bit security level, legacy of 1990s crypto restrictions.

Anonymous: client/server not authenticated.

Null: mostly grid traffic, integrity only (atypical)

Road Map



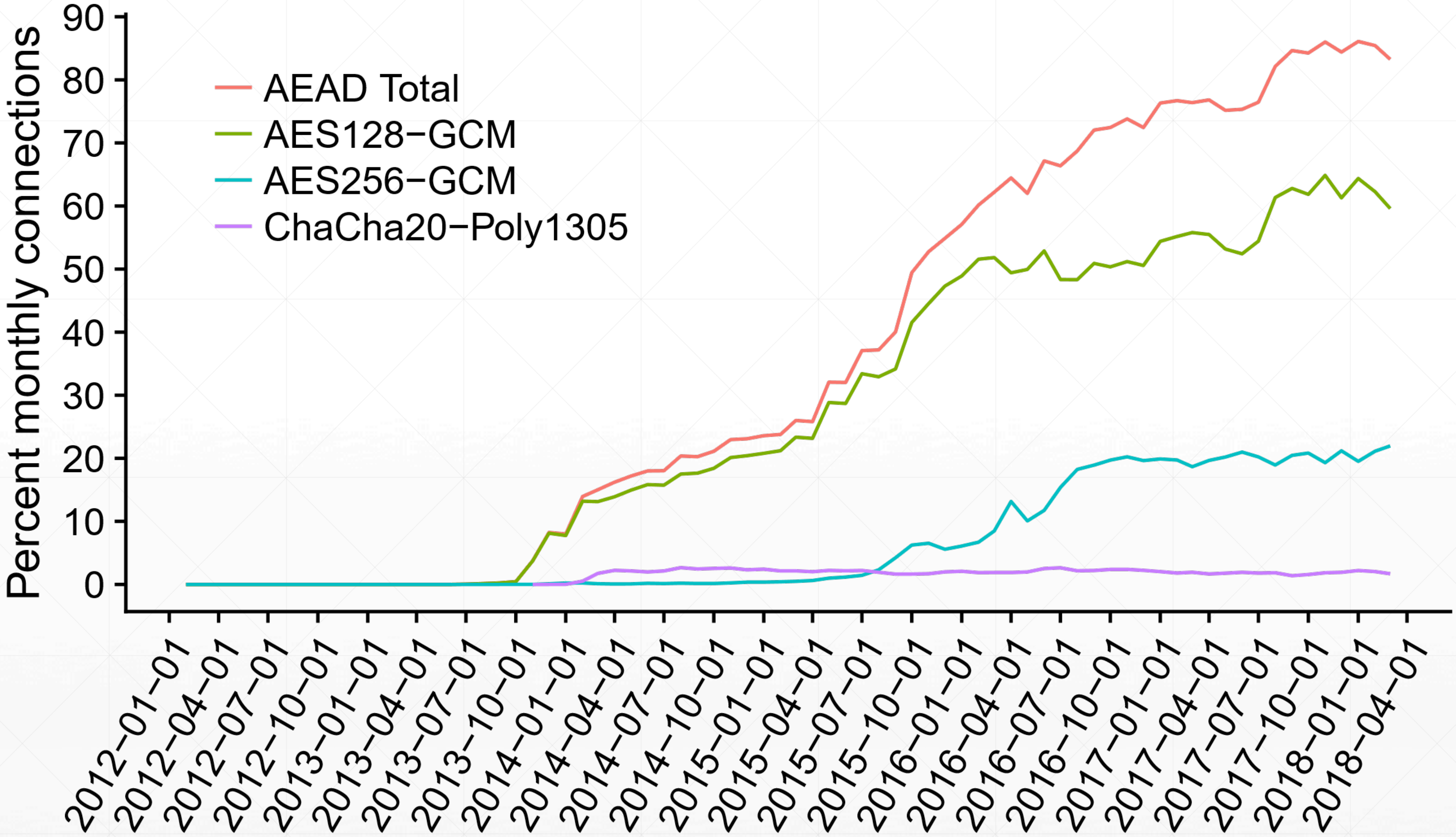
Intro

TLS Fingerprints

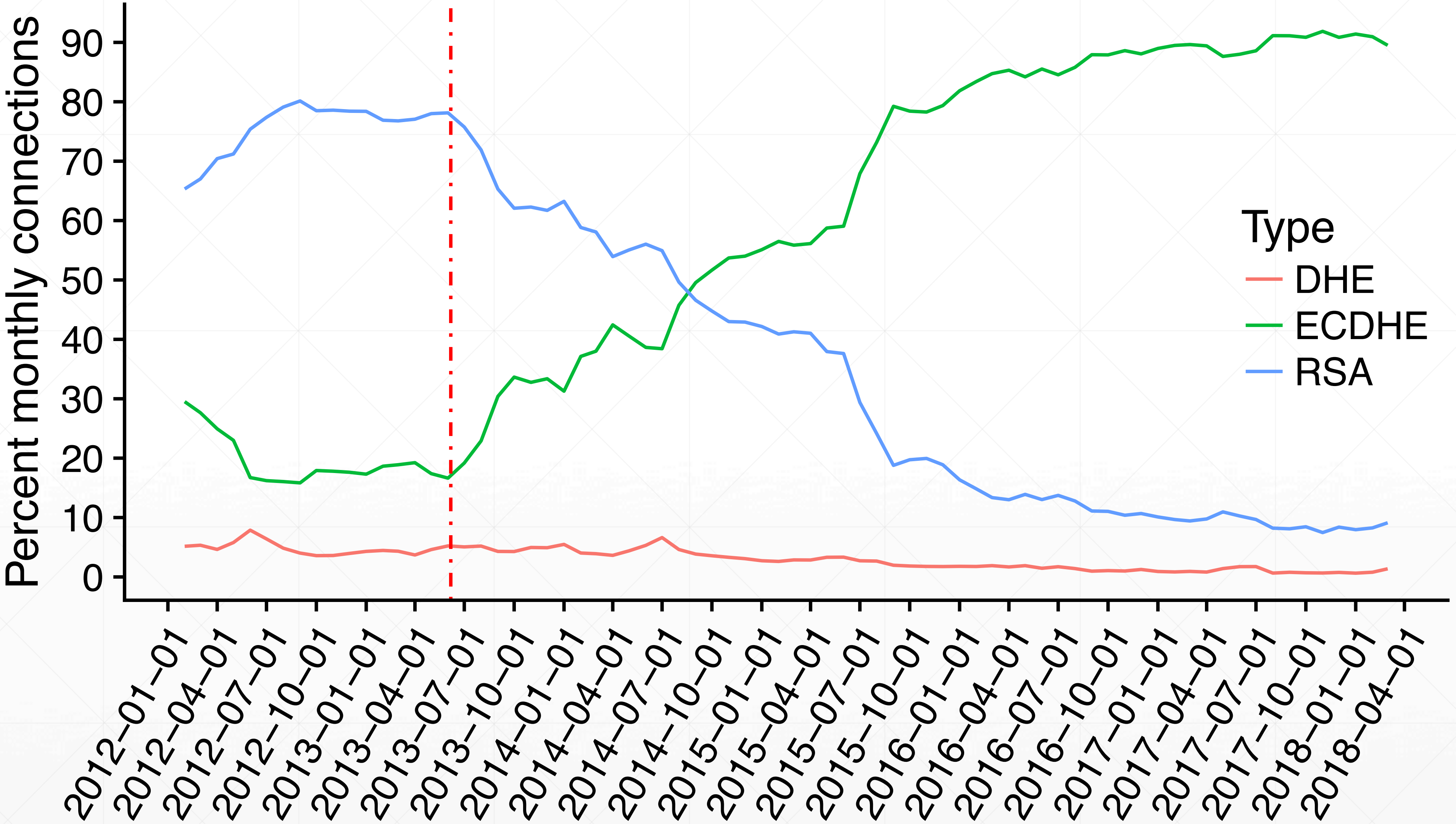
Vulnerability Analysis

Ecosystem Improvements

AEAD Usage


Percent monthly connections
0
10
20
30
40
50
60
70
80
90
AEAD Total
AES128−GCM
AES256−GCM
ChaCha20−Poly1305
2012−01−01
2012−04−01
2012−07−01
2012−10−01
2013−01−01
2013−04−01
2013−07−01
2013−10−01
2014−01−01
2014−04−01
2014−07−01
2014−10−01
2015−01−01
2015−04−01
2015−07−01
2015−10−01
2016−01−01
2016−04−01
2016−07−01
2016−10−01
2017−01−01
2017−04−01
2017−07−01
2017−10−01
2018−01−01
2018−04−01

SSL/TLS: Key Exchange methods



Dotted line: beginning of Snowden revelations

TLS 1.3 – Radical change

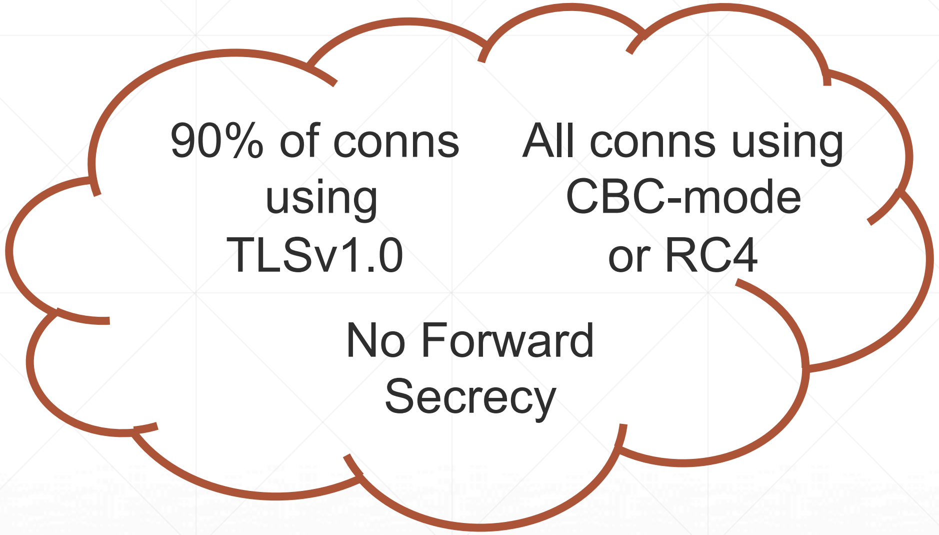
- Touches all parts of the protocol
  - Parts of the handshake are encrypted (e.g., certificates)
  - Cipher suites reduced from hundreds to 5 (CBC-mode, RC4 ciphers removed)

- TLS was just starting to see adoption at the end of our study.
  - 0.5% of clients advertised TLS 1.3 in February 2018.
  - 9.8% in March 2018.
  - 23.6% in April 2018.
  - But only 1.3% of connections actually negotiated TLS 1.3 in April 2018: server-side deployment lagging client-side.

- 6 years for TLS 1.2 to be used in more than 50% of the connections

Summary

- ❑ Several improvements in the ecosystem

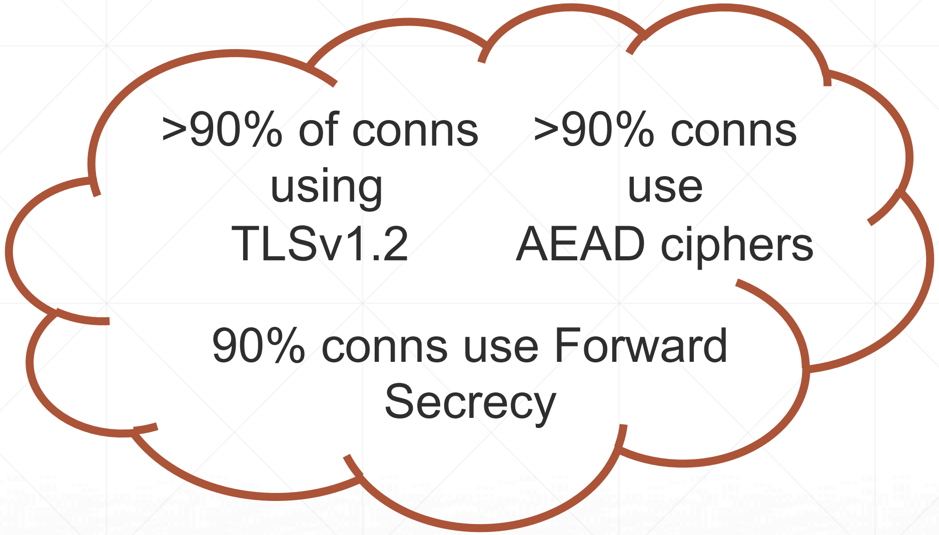


90% of conns using TLSv1.0

All conns using CBC-mode or RC4

No Forward Secrecy

2012



>90% of conns using TLSv1.2

>90% conns use AEAD ciphers

90% conns use Forward Secrecy

2018

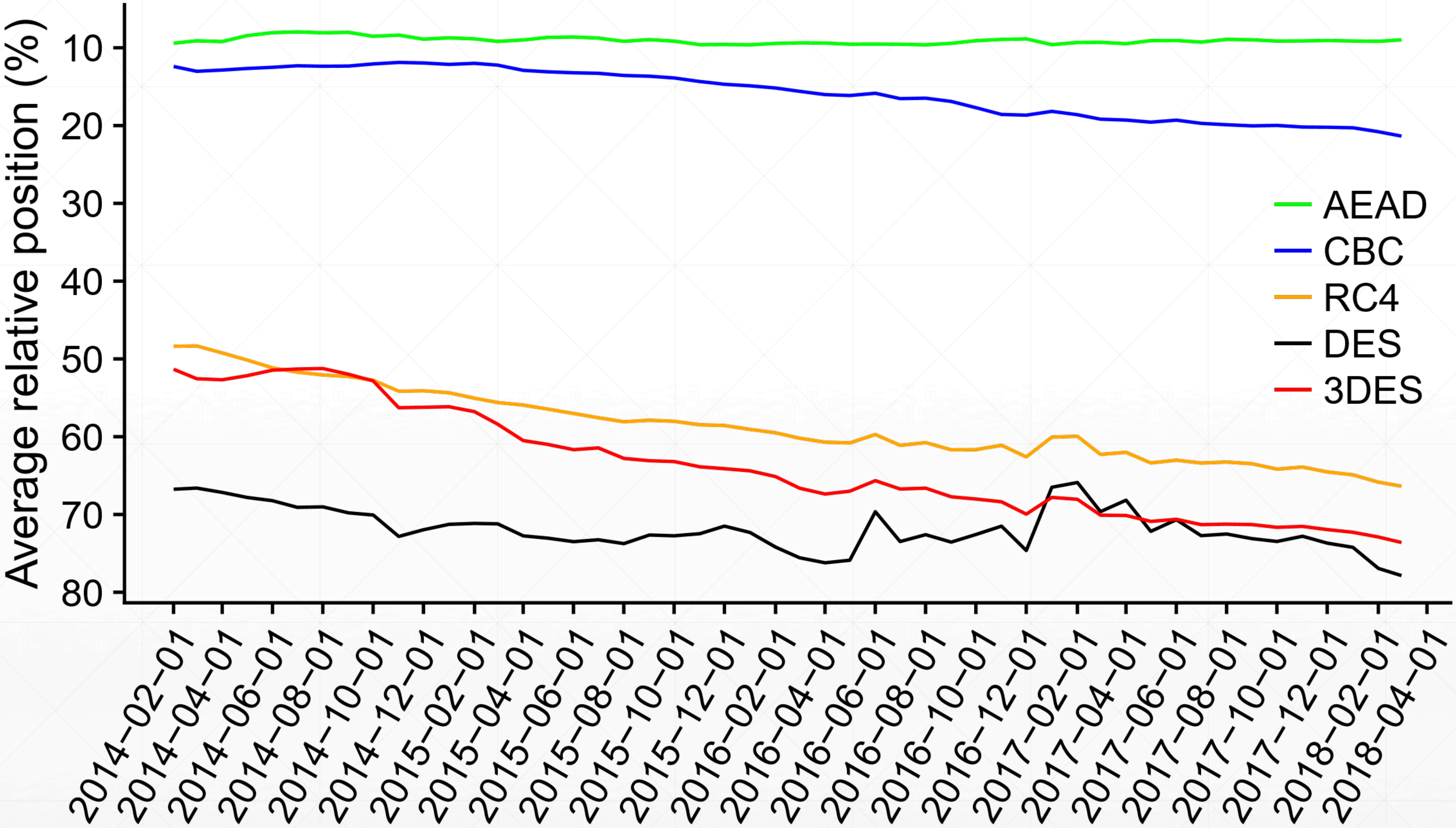
- ❑ Fast support of TLSv1.3 even before the RFC is finalized

Summary

- Backwards compatibility
  - Clients, especially browsers, are quick to adopt new algorithms they are slow to drop support for older ones
  - Risk of (new) downgrade attacks, room for misconfiguration

- Poor implementations
  - Long tail of clients with support of Null, Anonymous and export ciphers

Relative Position of ciphers


Average relative position (%)
AEAD
CBC
RC4
DES
3DES