TLS 1.3 - DIE ZUKUNFT DER NETZVERSCHLÜSSELUNG

Hanno Böck
https://hboeck.de

VORSTELLUNG

Hanno Böck

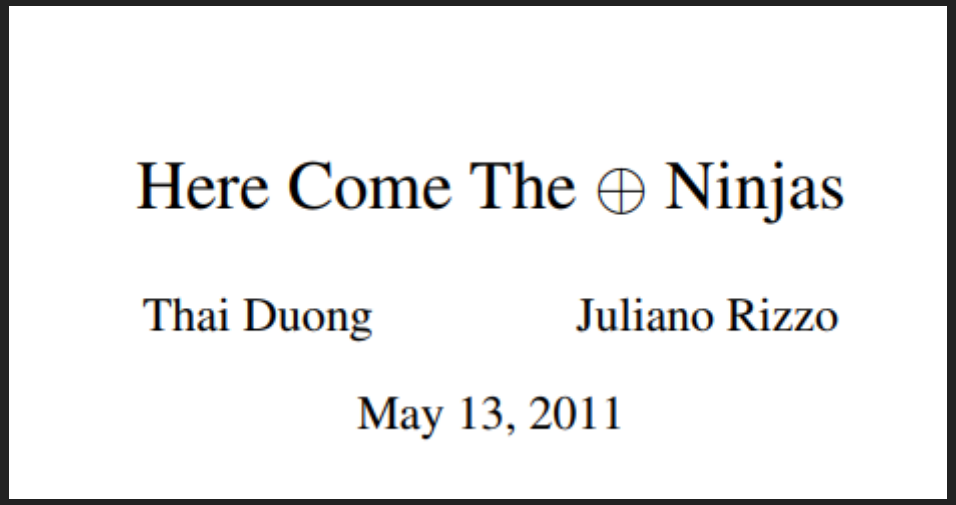
Schreibe regelmäßig für Golem.de und andere, veröffentliche monatlichen Bulletproof TLS Newsletter.

Fuzzing Project - unterstützt von der Core Infrastructure Initiative der Linux Foundation.

Nebenher: Administrator bei kleinem Webhoster schokokeks.org.

EIN BLICK ZURÜCK

BEAST-ANGRIFF (2011)



Here Come The ⊕ Ninjas

Thai Duong Juliano Rizzo

May 13, 2011

Source

BEAST

Angriff auf CBC-Verschlüsselung in TLS 1.0 und SSL 3.

TLS 1.1/1.2 nicht betroffen, aber damals paktisch nicht in Verwendung.

PADDING ORACLE

Security Flaws Induced by CBC Padding Applications to SSL, IPSEC, WTLS... Vaudenay (Eurocrypt 2002)

HINTERGRUND CBC/HMAC

TLS: Verschlüsselung und Authentifizierung

TLS wurde lange Zeit üblicherweise mit AES im CBC-Modus und HMAC verwendet

MAC-then-Pad-then-Encrypt

([Plaintext][MAC][Padding])Encrypt

TLS PADDING

0x00

0x01, 0x01

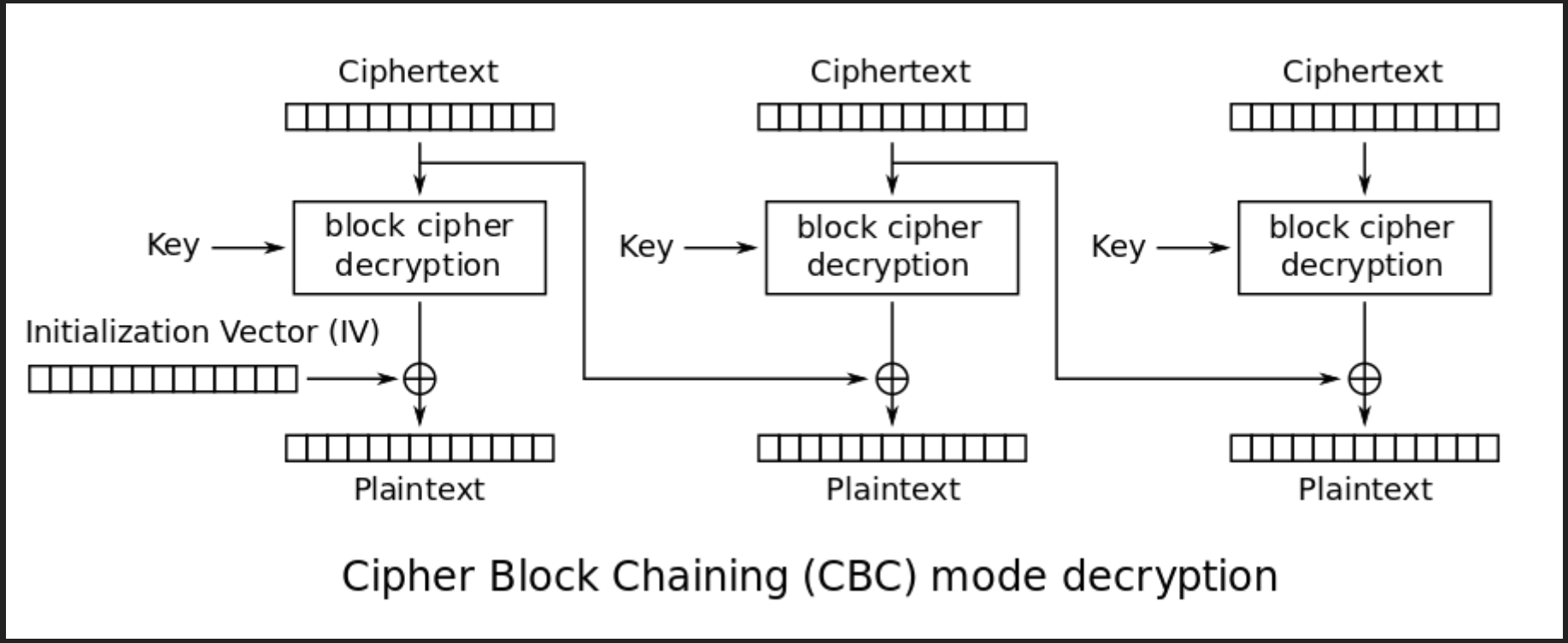
0x02, 0x02, 0x02

0x03, 0x03, 0x03, 0x03

TLS 1.0 / 1.1 ERROR ALERTS

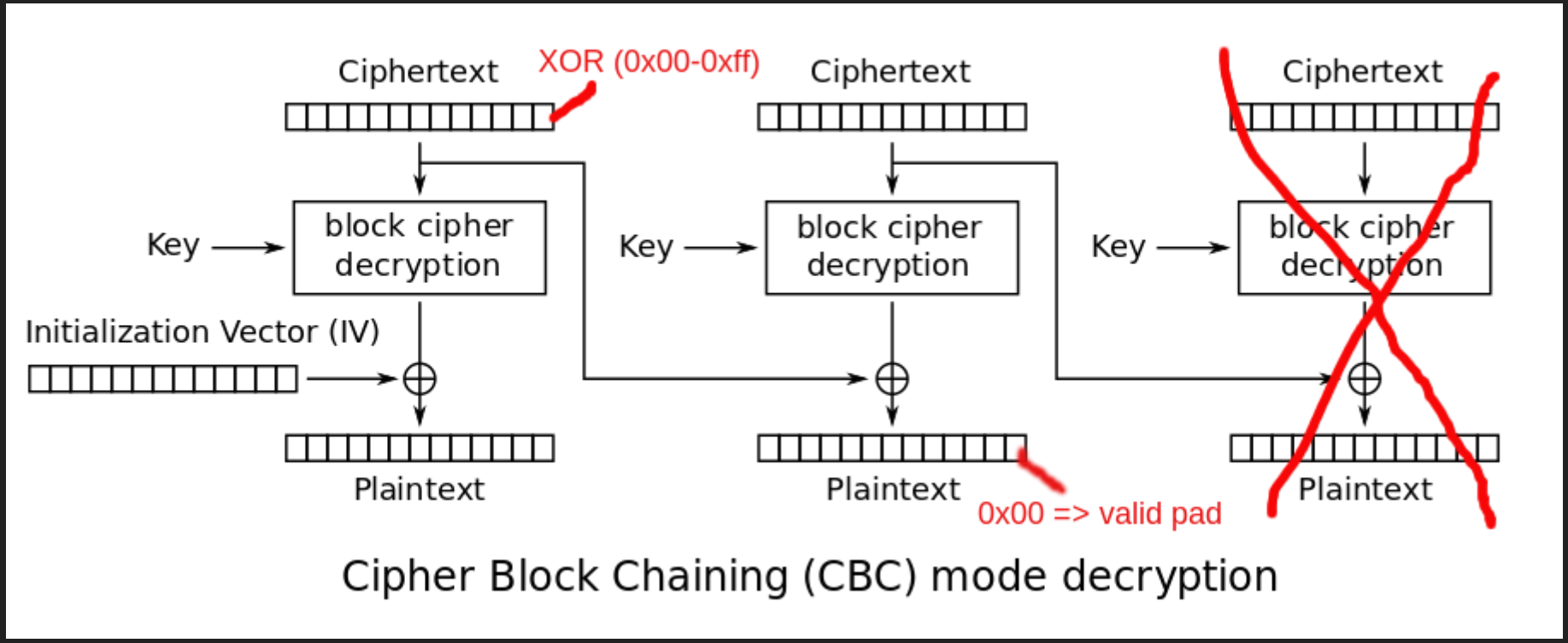
bad_record_mac
decryption_failed

CBC DECRYPTION



Cipher Block Chaining (CBC) mode decryption

CBC PADDING ORACLE



Cipher Block Chaining (CBC) mode decryption

TLS 1.2 (RFC 5246)

Canvel et al. [CBCTIME] have demonstrated a timing attack on CBC padding based on the time required to compute the MAC. [...] ***This leaves a small timing channel,*** *since MAC performance depends to some extent on the size of the data fragment,* ***but it is not believed to be large enough to be exploitable*** *[...].*

LUCKY THIRTEEN (2013)

Timing-Variante von Vaudenay-Angriff trotz Gegenmaßnahmen.

Schwer zu vermeiden, später weitere Varianten (Lucky Microseconds).

Fix in OpenSSL führte zu weiterem Padding-Problem.

RC4

2013: Erster Angriff auf RC4 in TLS, seither mehrere verbesserte Angriffe.

2015: RFC 7465, Prohibiting RC4 Cipher Suites

SSL IST TOT

SSLv2 DROWN

SSLv3 POODLE

Dank Downgrade- und Cross-Protokoll-Angriffen selbst dann ein Risiko, wenn diese Versionen überhaupt nicht verwendet werden.

TLS NACH LUCKY13 UND RC4-ANGRIFFEN

	CBC	RC4	GCM
TLS 1.0	unsicher	unsicher	n/a
TLS 1.1	unsicher	unsicher	n/a
TLS 1.2	unsicher	unsicher	sicher

GCM (RFC 5288 / TLS 1.2)

Each value of the nonce_explicit MUST be distinct for each distinct invocation of the GCM encrypt function for any fixed key. Failure to meet this uniqueness requirement can significantly degrade security. The nonce_explicit MAY be the 64-bit sequence number.

DOPPELTE NONCES BEI GCM

Eine kleine Zahl von Servern (184) nutzt doppelte Nonces.

Ca. 70.000 Hosts mit zufälligen Nonces (geringes Risiko).

Allerdings: Alles Implementierungsfehler, korrekte Implementierungen nicht betroffen.

Nonce-disrespecting Adversaries (Böck et al)

WAS BLEIBT?

Nur GCM-Modi in TLS 1.2 sind nicht kryptographisch gebrochen.

TLS 1.3

Entwicklung startete 2014, inzwischen 20 Entwürfe.

Zahlreiche Implementierungen bereits verfügbar.

Verabschiedung in Kürze.

FORWARD SECRECY

Früher zwei Möglichkeiten für Generierung des Session-Schlüssels: RSA-Verschlüsselung und Schlüsselaustausch (Diffie Hellman oder Elliptic Curve Diffie Hellman).

Nur der Schlüsselaustausch bietet Forward Secrecy.

RSA-Verschlüsselung sowieso sehr fragil (Bleichenbacher-Angriff).

TLS 1.3 UND FORWARD SECRECY

RSA-Schlüsselaustausch fliegt raus, Forward Secrecy nicht mehr optional.

UND DANN WAREN DA NOCH DIE BANKEN...

Vertreter von Banken-Verband BITS bittet im September 2016 darum, den unsicheren RSA-Schlüsselaustausch beizubehalten (wg. passivem Überwachungsequipment).

ANTWORT VON KENNY PATERSON

My view concerning your request: no.

Rationale: We're trying to build a more secure internet.

Meta-level comment:

You're a bit late to the party. We're metaphorically speaking at the stage of emptying the ash trays and hunting for the not quite empty beer cans.

More exactly, we are at draft 15 and RSA key transport disappeared from the spec about a dozen drafts ago. I know the banking industry is usually a bit slow off the mark, but this takes the biscuit.

NUR NOCH AUTHENTIFIZIERTE VERSCHLÜSSELUNG

TLS garantiert Verschlüsselung und Authentifizierung.

SSL/TLS nutzte meistens CBC/HMAC in der fragwürdigen Kombination MAC-then-Encrypt (Padding Oracles).

Authentifizierte Verschlüsselung: Verschlüsselungsmodi, die Verschlüsselung und Integrität kombinieren (GCM, Poly1305).

KOMPRESSION RAUS

CRIME-Angriff nutzte TLS-Kompression.

Problem nicht gelöst: Kompressionsangriffe funktionieren auch mittels HTTP-Kompression (BREACH, TIME, HEIST).

WEITERE ÄNDERUNGEN

RSA-PSS statt PKCS #1 1.5 (Bleichenbacher Signature Forgery, BERserk).

MD5, SHA1 raus (SLOTH).

Keine frei wählbaren elliptischen Kurven und Diffie-Hellman-Gruppen (Logjam)

Renegotiation raus (Triple Handshake).

Neue Berechnung Pre-Master-Secret (Triple Handshake).

CLIENT-ZERTIFIKATE

Bisher wurden Client-Zertifikate unverschlüsselt übertragen (Privacy).

TLS 1.3 verschlüsselt Client-Zertifikate.

CIPHERSUITES

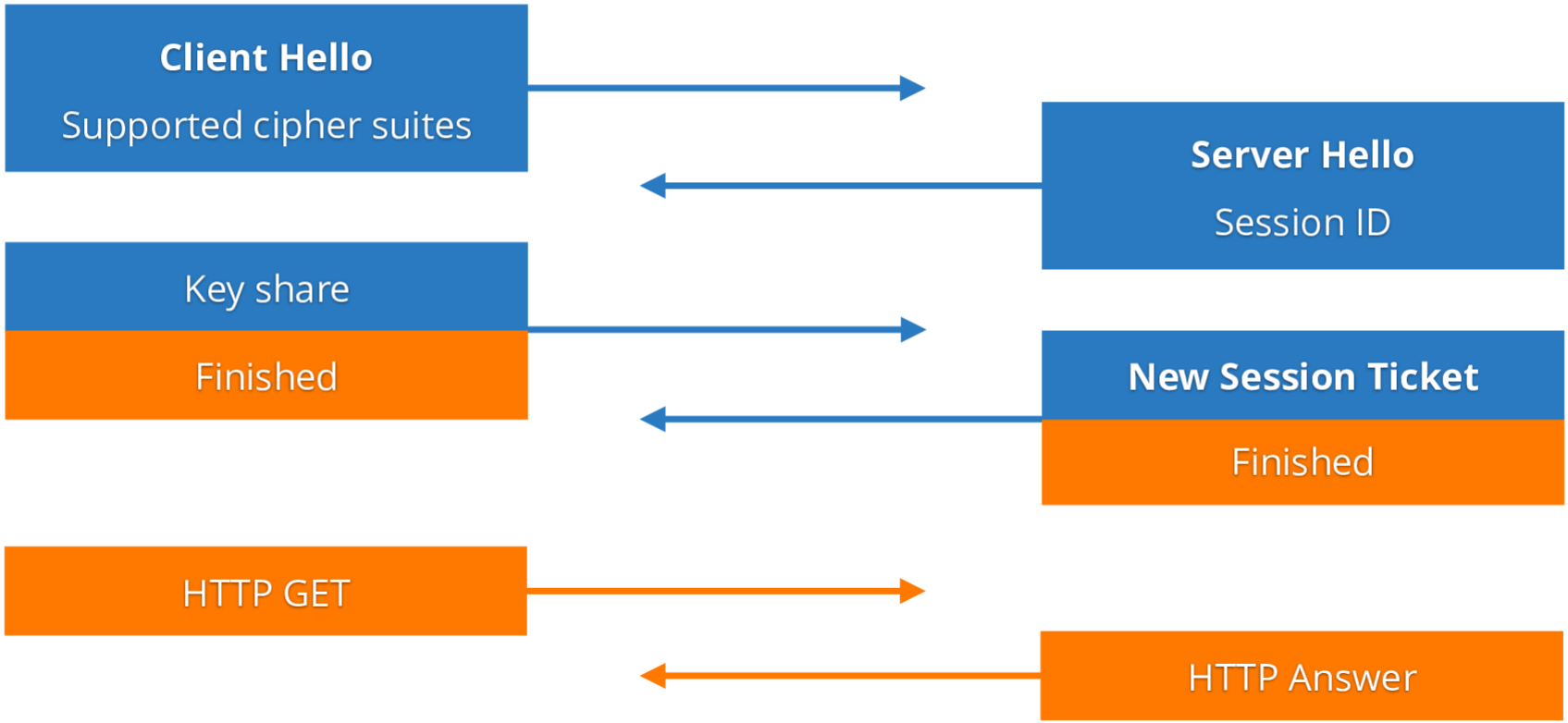
Ciphersuite alt: ECDHE-RSA-AES128-GCM-SHA256

Ciphersuite neu: AES128-GCM-SHA256

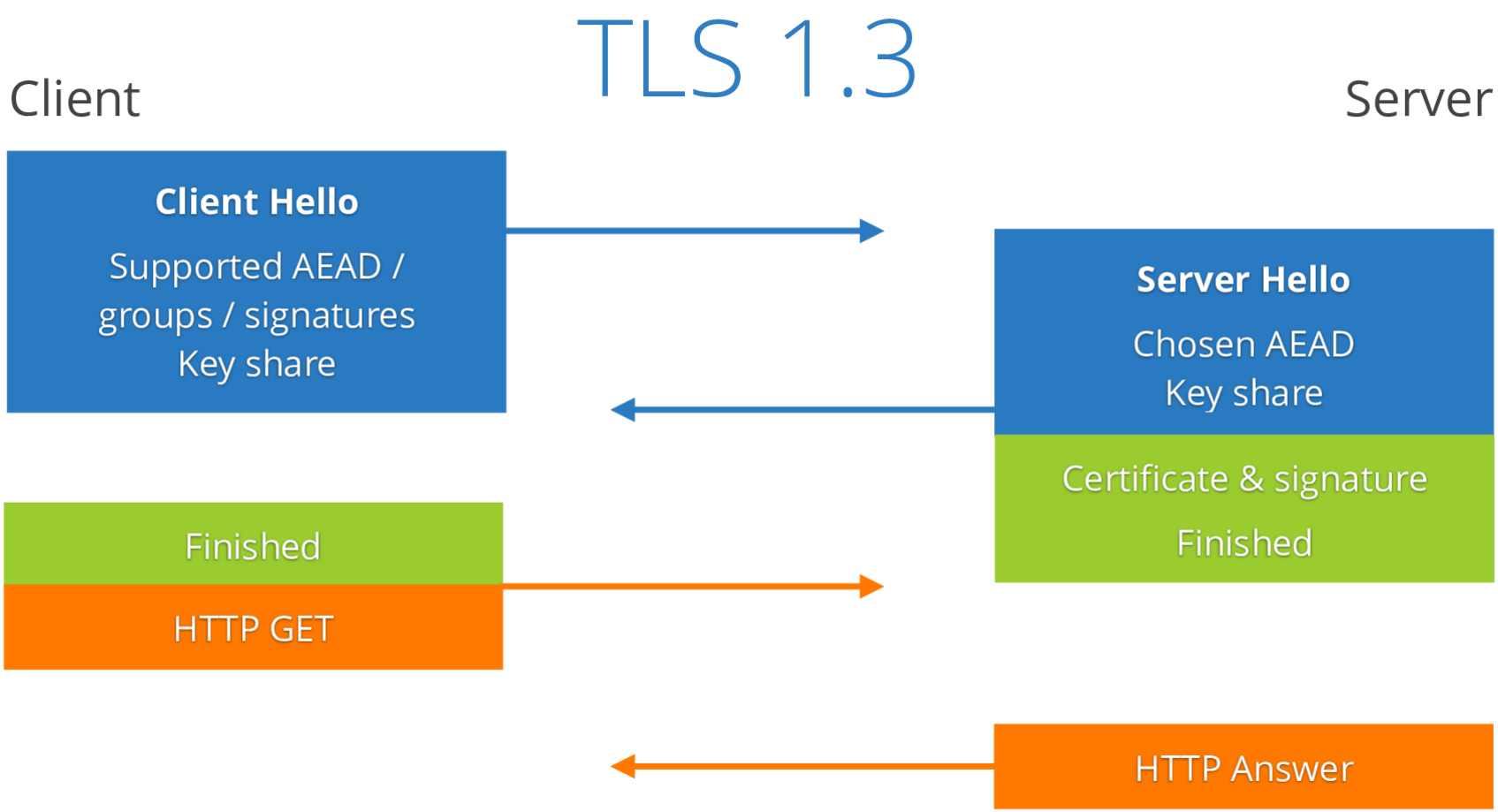
Key Exchange wird unabhängig davon gewählt.

OPTIMIERTER HANDSHAKE

TLS 1.2 ECDHE



Quelle: Filippo Valsorda, Nick Sullivan, Cloudflare



Quelle: Filippo Valsorda, Nick Sullivan, Cloudflare

0-RTT

0-RTT

Bei vorhandemen Pre-Shared-Key von voriger Verbindung können unmittelbar im ClientHello verschlüsselte Daten mitgeschickt werden.

PROBLEME MIT 0-RTT

Replay-Angriffe.

Seitenkanalangriffe.

0-RTT

Aus Performance-Sicht wünschenswert, aber mit vielen Risiken verbunden.

0-RTT erfordert detailierte Kenntnisse über die Applikationslogik, darf nicht per default genutzt werden.

0-RTT UND HTTPS

HTTP-GET-Requests müssen idempotent sein (dürfen auf dem Server keine Änderungen hervorrufen).

VERSIONSNUMMER

TLS hat ein extrem komplexes Versionsaushandlungsverfahren.

TLS-VERSIONSAUSHANDLUNG

Client schickt ClientHello mit maximal unterstützter TLS-Version.

Falls Server diese nicht unterstützt antwortet er mit niedrigerer, maximal unterstützter TLS-Version.

BEISPIEL VERSIONSAUSHANDLUNG

ClientHello mit TLS 1.2

Server unterstützt nur TLS 1.0, daher:

ServerHello mit TLS 1.0

Anschließend: Verbindung mit TLS 1.0.

EXTREM KOMPLEX?

NAJA...

Zumindest zu komplex für Cisco, IBM, Citrix und zahlreiche andere Hersteller.

VERSION INTOLERANCE

ClientHello mit TLS 1.2

Server unterstützt nur TLS 1.0, daher...

Absturz, Verbindungsabbruch, TLS-Alert, ...

FALLBACKS

Da es zu viele defekte Server gab, die die Versionsaushandlung nicht korrekt durchführen, hatten Browser Fallbacks eingeführt.

Schlägt Verbindung mit TLS 1.2 fehl: Versuche es nochmal mit TLS 1.1, TLS 1.0, SSLv3.

FALLBACKS? IST DAS SICHER?

Nein, natürlich nicht.

Angriffe nutzen Versions-Fallbacks: Virtual Host Confusion, POODLE.

Daher: SCSV, Mitigation für Fallbacks.

TLS 1.3 UND VERSIONSAUSHANDLUNG

Versionsaushandlung als Extension, alte Version wird auf 3.3 (== TLS 1.2) festgenagelt.

GREASE

Clients schicken gelegentlich zufällig bestimmte reservierte "GREASE"-Werte mit, um sicherzustellen, dass Server diese korrekt ignorieren.

Bei der Entwicklung von TLS 1.3 wurde mit allen Mitteln versucht, eine Kompatibilität mit defekten Legacy-Implementierungen herzustellen.

Dann gabs da aber noch Blue Coat.

BlueCoat and other proxies hang up during TLS 1.3

Project Member Reported by jayhlee@google.com, Feb 21

Chrome Version: 56
OS: Chrome and Windows

What steps will reproduce the problem?
(1) BlueCoat 6.5 proxy.
(2) Chrome OS 56 or Chrome browser 56
(3) Attempt to connect to a Google service (youtube, accounts.google.com, etc.

What is the expected result?
Successful connection. Client and proxy may negotiate down to TLS 1.2 instead of TLS 1.3.

What happens instead?
When Chrome attempts to connect via TLS 1.3, BlueCoat hangs up connection.

Further details:

We have at least one very large customer seeing similar issues against BlueCoat. The connection fails with SSL_HANDSHAKE_ERROR / ERR_CONNECTION_CLOSED. Customer found that restricting to TLS 1.2 via policy resolves the issue for Chrome 56 stable. Net internals logs are at:

https://drive.google.com/corp/drive/folders/0B3BtTQPWWixOMk1FNkhMekJnNEU (google.com view only)

Other large EDU customers are seeing similar issues and I'm working to gather details from them on proxy / firewall in use. Suspect many are using SSL / TLS inspection which is common among EDUs.

Marking this as ReleaseBlock-Stable and P1 as I believe this is breaking Chrome for many customers.

Bluecoat version is 6.5 for affected customer.

SOFTWARE

Unterstützung in Firefox (NSS) und Chrome (BoringSSL).

Experimentelle Unterstützung in OpenSSL (zukünftige Version 1.1.1).

[DEMO]

OPENSSL UND TLS 1.3

Die aktuelle Version von OpenSSL (1.1.0, noch ohne TLS-1.3-Support) hat viele API-Änderungen.

Viel inkompatible Software.

Testet Eure Software jetzt mit OpenSSL 1.1.0.

TLS 1.3 IST NICHT ALLES

Viele Verbesserungen in Sachen TLS haben nicht direkt mit der Protokollversion zu tun.

Certificate Transparency, HTTP Public Key Pinning, CAA-Records, SameSite-Cookies, ...

Es tut sich viel - und TLS ist heute viel sicherer als früher.

CERTIFICATE TRANSPARENCY

Öffentliche Logs mit Zertifikaten.

Webinterface: crt.sh

Notification-Services: Certspotter, Facebook.

ZUKUNFT

0-RTT aussichtsreicher Kandidat für zukünftige Sicherheitslücken.

QUIC statt TLS?

Post-Quanten-Kryptographie steht an.

TLS 1.3 KOMMT BALD

Fragen?