AES Avalanche Cryptanalysis

CS 6343, Cryptography — Texas Tech University

Abdul Serwadda, Ph.D. Abdul.Serwadda@ttu.edu

et. al. M. Aziz mazizull@ttu.edu

By Scott Weeden sweeden@ttu.edu

April 5, 2026

Abstract

Avalanche is the spread of a one-bit plaintext or key change into many ciphertext bit differences after a few rounds. We plot that spread using a toy encrypt-only block cipher with logged intermediate states, compare electronic codebook and cipher-block chaining on a bitmap with OpenSSL, and summarize client-offered and server-selected ciphers from a classroom packet capture (with a small live backup probe when no capture is available). Figures and tables rebuild from `scripts/run_coursework_outputs.py`. The main text includes background on finite-field mixing and modes. For teaching only; not for production use.

Introduction

Who this is for. We assume **no prior coursework in cryptography**. You should be comfortable with binary, hexadecimal, and the idea that files are bytes; everything else—finite fields, S-boxes, block modes, and what "AES" actually does step by step—is built up in Section before we discuss the Python tracer in Section .

Why trace a toy cipher? In applications, encryption is often a single API call. That hides the internal *round structure* that determines how quickly a one-bit change in plaintext or key *avalanches* through the state. For learning (and for structured homework), we need **visibility**: intermediate 16-byte states after SubBytes, ShiftRows, MixColumns, and AddRoundKey, so we can plot Hamming distance along the pipeline and **see** diffusion.

What this document covers. We follow three threads that mirror a typical undergraduate lab brief and the **research-pipeline** style workflow used in `rd-ralph-template` (extract questions → design experiments → generate artifacts):

1. **AES-128 internals** (Questions 1-style): a toy encrypt-only tracer, `toy_aes128_trace.py`, with the standard S-box, GF(2^8) arithmetic for MixColumns (`xtime`), optional ShiftRows ablation, and an alternate MixColumns matrix M_{new} for comparison.
2. **Block cipher modes** (Questions 2-style): OpenSSL AES-128-ECB vs. AES-128-CBC on a BMP, illustrating why ECB leaks visual structure and CBC masks it at the byte level (Dworkin, 2001).

3. **TLS in the wild** (Questions 3-style): with `output/results/q3/q3.pcapng` present, we decode TCP/TLS handshakes using `tshark`, aggregate Client Hello / Server Hello fields into `output/results/q3-cipher-summary.json` and per-host JSON under `output/results/q3/` (matching the Wireshark-oriented brief: negotiated suite, TLS 1.2 vs. 1.3 behavior, and rationale for common AEAD choices). A classroom capture typically combines multiple browsers and many sites; our checked-in export uses one such capture (optionally pruned to organizational families). If no PCAP is available, the driver falls back to live `openssl s_client` against nine template hosts (Dierks and Rescorla, 2008).

Reproducibility. Every figure in Section can be regenerated by `python scripts/run_coursework_outputs.py` The toy AES path is checked against PyCryptodome on a single block for standard options (`verify_aes_against_pycryptodome`). This is **not** a production or side-channel-hardened implementation.

Preliminaries: cryptography from zero

This section explains the **minimum concepts** needed to read `toy_aes128_trace.py` and the experiments, in the order a reader typically needs them. It aligns with the **learning goals** surfaced when the homework is decomposed into questions about (i) round-wise diffusion, (ii) modes on real file data, and (iii) TLS cipher suites.

Secret writing without jargon

Encryption turns a **plaintext** message into a **ciphertext** that should look random to anyone who does not know the **key**. **Decryption** recovers the plaintext with the correct key. A **symmetric** cipher uses one shared secret key for both directions (unlike public-key systems, which we do not use here).

A **block cipher** acts on a **fixed-size chunk** of data at a time. AES-128 uses 128-bit blocks = **16 bytes**. Longer messages are handled by a **mode of operation** (Section), which repeatedly invokes the block cipher.

Bits, bytes, and XOR

A **byte** is eight bits, often written in hexadecimal (e.g. `0x01`–`0xFF`). Computers store AES state as bytes.

XOR (exclusive OR), written $\oplus$ in equations and `^` in Python for integers, combines two bits: the result is `1` if the bits differ. For bytes, XOR is applied bit by bit. Important properties: $a \oplus a = 0$, and $a \oplus b \oplus b = a$. AES uses XOR heavily (**AddRoundKey** XORs a round key into the state). If you are new to XOR, think: "flip bits where the key has a `1`."

Confusion and diffusion (intuition only)

Shannon's informal goals still guide block cipher design (Daemen and Rijmen, 2002):

- **Confusion**: the ciphertext should depend on the key in a complicated way so that changing one key bit changes the output in a hard-to-predict pattern. Nonlinear **substitution** (the S-box) provides confusion.

- **Diffusion**: changing one input bit should **spread** influence across many output bits quickly. **Permutation** (ShiftRows) and **linear mixing** (MixColumns) spread information across bytes.

Our Hamming-distance plots (Section) are a **quantitative** view of diffusion: after how many steps do almost all 128 bits differ between two close inputs?

Polynomials with coefficients in $\{0, 1\}$

Before finite fields, recall **polynomials** where each coefficient is only 0 or 1, and arithmetic on coefficients is mod 2 (so $1 + 1 = 0$). For example,

$$(x^3 + 1)(x^2 + x) = x^5 + x^4 + x^3 + x$$

with every coefficient reduced mod 2. This is the world of $\mathrm{GF}(2)[x]$.

The field $\mathrm{GF}(2^8)$ in one page

What problem does it solve? MixColumns must **multiply bytes** by small constants $(1, 2, 3)$ in a way that is **invertible** (so decryption exists) and **algebraically clean** (so proofs and attacks are structured). Ordinary integer multiplication mod 256 is not the right structure; AES uses a **finite field** with exactly 256 elements, written $\mathrm{GF}(2^8)$ or $\mathbb{F}_{2^8}$.

Elements = bytes. Fix an irreducible polynomial $m(x)$ of degree 8 over GF(2). AES standardizes

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

(the "Rijndael polynomial"). Every field element is represented uniquely as a polynomial of degree < 8, hence **8 coefficients ⇒ one byte**. The byte value is the integer whose bits are those coefficients (e.g. $x^7 + x^3 + x$ corresponds to a specific 8-bit pattern).

Addition in $\mathrm{GF}(2^8)$. Adding two field elements is **XOR of the two bytes**. This is the same as adding polynomials with coefficients mod 2. In code, every time we "XOR state with key," we are adding in the field *and* XOR-ing bits.

Multiplication in $\mathrm{GF}(2^8)$. Multiply two polynomials, then take the remainder when dividing by $m(x)$. This keeps the result inside one byte. General multiplication is implemented with bit shifts and XORs ("Russian peasant" or similar); our tracer only needs a special case.

The `xtime` operation (multiply by x)

In polynomial language, x denotes the monomial x^1. Multiplying a field element $a(x)$ by x, then reducing mod $m(x)$, is exactly what the Python function `xtime` computes on the 8-bit value a:

- If the high bit of a is `0`, left-shift by one (multiply by x with no wrap).
- If the high bit is `1`, left-shift and XOR with `0x1B` (the bit pattern of $m(x)$ minus x^8), which performs the reduction mod $m(x)$.

MixColumns builds products by 2 and 3 from `xtime`:

$$2 \cdot a := \texttt{xtime}(a), \qquad 3 \cdot a := \texttt{xtime}(a) \oplus a,$$

and uses $1 \cdot a = a$. That is why the tracer never needs a general multiply routine for the default matrix.

AES state layout

AES operates on a 4×4 array of bytes (16 bytes total), indexed in **column-major** order in our code: `s[col][row]`. A **round** applies a fixed sequence of transformations to this grid.

SubBytes and the S-box

SubBytes replaces each byte by another byte using a public lookup table, the **S-box** (`SBOX` in code). The standard table is bijective (invertible) and highly nonlinear; its algebraic structure is related to inversion in $\mathrm{GF}(2^8)$ followed by an affine map (Daemen and Rijmen, 2002). For this lab you only need: **nonlinear substitution breaks simple linear patterns** and contributes to confusion.

ShiftRows and MixColumns

ShiftRows rotates rows of the state grid so bytes move to different columns—cheap **permutation** for diffusion across columns.

MixColumns treats each column as a four-tuple of field elements and multiplies it by a fixed 4×4 matrix over $\mathrm{GF}(2^8)$. Entries are only $1, 2, 3$, implemented with XOR and `xtime`. The matrix is chosen so that changing one input byte affects all four output bytes (**branching**). Our coursework allows replacing this matrix by M_{new} (`M_NEW`) to see how diffusion curves change.

Key schedule and AddRoundKey

The 128-bit key is expanded into **eleven** 128-bit **round keys** `rk[0]..rk[10]` (`key_expansion` uses the S-box and XORs with round constants `rcon`). **AddRoundKey** XORs one round key into the entire state.

Round structure (encrypt):

- **Round 0:** AddRoundKey only.
- **Rounds 1–9:** SubBytes → ShiftRows → MixColumns → AddRoundKey.
- **Round 10:** SubBytes → ShiftRows → AddRoundKey (no MixColumns).

The tracer `encrypt_trace` follows this order and can **skip ShiftRows** or swap MixColumns to `M_NEW` for assigned ablations.

Modes (ECB vs. CBC)

The block cipher only defines $E_K(\cdot)$ on 16 bytes. To encrypt a long file, a **mode** chains blocks.

ECB encrypts each block separately. Equal plaintext blocks $\Rightarrow$ equal ciphertext blocks, so **patterns survive**—visibly for uncompressed images (Dworkin, 2001).

CBC XORs each plaintext block with the **previous ciphertext block** before encrypting; the first block uses a public **initialization vector (IV)**. Identical plaintext blocks at different positions usually yield different ciphertext blocks, hiding trivial repetition.

Hamming distance on traces

We run the tracer twice with slightly different plaintext or key, logging the 16-byte state after the same sequence of steps. The **bit Hamming distance** counts differing bits between aligned states. A steep rise means **avalanche**: small input changes explode into large output differences—the empirical signature of good diffusion.

Pipeline-aligned checklist (concepts ↔ code)

Table 1 summarizes what to understand for each part of the repository.

Table 1: Concepts needed for the toy AES implementation and where they appear.

Idea	Where to look
Byte / XOR / block	`encrypt_trace`, `_add_round_key`
GF(2^8), `xtime`, $\times 2$, $\times 3$	`xtime`, `mix_single_column`
S-box / SubBytes	`SBOX`, `_sub_bytes`
ShiftRows	`_shift_rows`, option `disable_shift_rows`
MixColumns / M_{new}	`mix_single_column`, `M_NEW`, `TraceOpts.mix_matrix`
Key schedule	`key_expansion`, `sub_word`, `rot_word`, `rcon`
Diffusion metrics	`collect_hamming_curve`, `hamming_bits`

Methodology

Scientific methodology

We follow the usual empirical cycle: pose **precise questions**, state **testable expectations** (hypotheses appear with each experiment in Section), define **controlled procedures** and **metrics**, generate **reproducible artifacts** (JSON summaries and PDF figures from the repository scripts), and interpret outcomes against published standards (NIST, 2001; Dworkin, 2001; Dierks and Rescorla, 2008). The three questions below structure *what* we measure; the subsections after them describe *how* the first study is implemented in code.

Three guiding questions

Question 1: AES implementation and round-level behavior. How does an AES-128 **implementation** realize the FIPS-197 round stack (SubBytes, ShiftRows, MixColumns, AddRoundKey), and how does **avalanche**—as bit-level Hamming distance between paired intermediate states—evolve when plaintext or key changes by a single bit? Methodologically, we require an **instrumented**, encrypt-only tracer on one block so every layer is logged, optional **ablations** (ShiftRows off, alternate MixColumns matrix), and **external check** against a library cipher for the same block.

Question 2: AES methodology for images versus typical text or files. For the **same** underlying block cipher, how should **methodology** differ when ciphertext is interpreted as a byte raster (e.g., an uncompressed BMP) versus opaque message or file bytes? Redundant, spatially structured plaintext makes **mode of operation** central: independent block encryption (ECB) versus chaining with an IV (CBC) changes what an analyst can *see* in raw or header-repaired

previews. We compare OpenSSL AES-128-ECB and AES-128-CBC on one image artifact and relate observations to guidance on modes for long, patterned data (Dworkin, 2001).

Question 3: AES in the real world (transport security). How is AES **encountered operationally** when applications use TLS? Methodologically, we observe **ClientHello** cipher-suite offers and **ServerHello** selections from packet capture (or, when no capture is available, a small negotiated-cipher probe), classify bulk algorithms and protocol version (TLS 1.3 vs. TLS 1.2), and summarize how modern stacks cluster on AEAD-based suites.

Instrument: `toy_aes128_trace.py` (Question 1)

The following subsections map the Python module to Section . Read them together: preliminaries give the **vocabulary**; here we show **which function implements which idea**. The module is **encrypt-only**, **single-block**, and for **education**; it mirrors AES-128 in FIPS-197 (NIST, 2001).

Constants and field arithmetic

The global `SBOX` is the AES substitution table (Section). The function `xtime(a)` is multiplication by x in $\mathrm{GF}(2^8)$ mod $m(x) = x^8 + x^4 + x^3 + x + 1$ (Section); every MixColumns coefficient in $\{2, 3\}$ is built from `xtime` and XOR.

MixColumns and M_{new}

`mix_single_column` multiplies one column by a 4×4 matrix over $\mathrm{GF}(2^8)$. The default matrix is the AES circulant with coefficients $\{2, 3, 1, 1\}$. For coursework Part C, `TraceOpts.mix_matrix` can point to `M_NEW`:

$$\begin{bmatrix} 3 & 2 & 1 & 1 \\ 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 2 \\ 2 & 1 & 1 & 3 \end{bmatrix}$$

so you can **compare diffusion curves** when linear mixing changes but ShiftRows and the rest remain.

Key expansion

`key_expansion` implements AES-128 ($N_k = 4$, $N_r = 10$). `sub_word` applies `SBOX` to four bytes; `rot_word` rotates a 32-bit word; `rcon` supplies round constants XORed into the schedule. Output: eleven round keys `rk[0..10]` (Section).

Round function and logging

`encrypt_trace(plaintext, key, opts, log)` encrypts one block. If `log` is provided, it receives `(label, round, state_bytes)` after:

- initial AddRoundKey (`r0_add_round_key`);
- for rounds $1 \ldots 9$: SubBytes, ShiftRows (unless `opts.disable_shift_rows`), MixColumns (using `opts.mix_matrix` if set), AddRoundKey;
- round 10: SubBytes, ShiftRows (optional skip), AddRoundKey (no MixColumns).

State layout is `s[col][row]` (column-major), matching Section .

Hamming analysis

`hamming_bits` / `hamming_bytes` compare two 16-byte states. `collect_hamming_curve` encrypts twice, collects two aligned lists of intermediate states, and returns per-step bit distances—the data behind the Question 1 figures.

External validation

`run_coursework_outputs.py` calls `verify_aes_against_pycryptodome`: under default options, ciphertext must match PyCryptodome AES-ECB for one block. This validates functional agreement with a reference, **not** resistance to timing or power analysis.

Experiments

All figures below are produced by `python scripts/run_coursework_outputs.py` from the repository root (conda environment `rd-ralph-template` with Matplotlib, NumPy, PyCryptodome, Pillow; OpenSSL on `PATH`). Paths are relative to `output/diagrams/`.

Question 1: Toy AES Hamming traces (Parts A–D)

Research question (RQ1). For a toy, instrumented 128-bit implementation, how does **bit-level Hamming distance** evolve step-by-step when plaintext differs by one bit, and separately when the key differs by one bit? How do **ShiftRows ablation** and **replacement of standard MixColumns with** M_{new} change those curves relative to the nominal pipeline?

Hypothesis (H1). After the initial rounds, Hamming distance between traces from a one-bit **plaintext** perturbation rises well above trivial levels. **Removing ShiftRows** or **altering MixColumns** (via M_{new}) should *weaken or reshape* that rise compared to standard Advanced Encryption Standard–128.

Experiment (E1). Implement SubBytes, ShiftRows, MixColumns over GF(2^8) using `xtime` (or equivalent), and AddRoundKey; validate against a library on a **single block** for baseline correctness only. Fix K all-zero and P all-zero; let P' flip one bit (byte 0 = `0x01`). Log states after each traced step for rounds 0–10; compute byte and bit Hamming distance between paired states. Repeat with ShiftRows disabled (Part B), with M_{new} and ShiftRows on (Part C), and with plaintext fixed and a one-bit key flip repeating A–C (Part D); tabulate peak and final distances. An optional extension is to discuss averaging over multiple random plaintext pairs ("ten-plaintext" variant).

Code excerpts (`scripts/toy_aes128_trace.py`). Listing 1 shows `xtime` and `mix_single_column` (default AES matrix vs. optional `opts.mix_matrix`). Listing 2 is `TraceOpts`. Listing 3 is the traced encrypt path (ShiftRows optional; MixColumns uses `opts.mix_matrix`). Listing 4 pairs two traces and builds the Hamming bit curve. Listing 5 defines M_{new}.

Listing 1: `xtime` and `mix_single_column` (coefficients 2 and 3 via `xtime`).

```
def xtime(a: int) -> int:
    return (((a << 1) ^ 0x1B) & 0xFF) if (a & 0x80) else ((a << 1) & 0xFF)


def mix_single_column(col: list[int], matrix: list[list[int]] | None = None) -> list[int]:
```

```
    """Mix one 4-byte column. Default = AES MixColumns matrix."""
    if matrix is None:
        matrix = [[2, 3, 1, 1], [1, 2, 3, 1], [1, 1, 2, 3], [3, 1, 1, 2]]
    out = [0, 0, 0, 0]
    for i in range(4):
        v = 0
        for j in range(4):
            c = matrix[i][j]
            t = col[j]
            if c == 1:
                v ^= t
            elif c == 2:
                v ^= xtime(t)
            elif c == 3:
                v ^= xtime(t) ^ t
            else:
                acc = 0
                bb = t
                for k in range(8):
                    if c & 1:
                        acc ^= bb
                    bb = xtime(bb)
                    c >>= 1
                v ^= acc & 0xFF
        out[i] = v & 0xFF
    return out
```

Listing 2: Tracer options: disable ShiftRows and custom MixColumns matrix.

```
@dataclass
class TraceOpts:
    disable_shift_rows: bool = False
    mix_matrix: list[list[int]] | None = None  # 4x4 coeffs for MixColumns
```

Listing 3: `encrypt_trace`: logged emit after each layer (rounds 1–9 and round 10).

```
def encrypt_trace(
    plaintext: bytes,
    key: bytes,
    opts: TraceOpts | None = None,
    log: Callable[[str, int, bytes], None] | None = None,
) -> bytes:
    """Encrypt 16-byte block; optional log(label, round, state_bytes)."""
    if opts is None:
        opts = TraceOpts()
    rk = key_expansion(key)
    s = _state_from_bytes(plaintext)

    def emit(label: str, rnd: int) -> None:
        if log:
            log(label, rnd, _bytes_from_state(s))

    _add_round_key(s, rk[0])
    emit("r0_add_round_key", 0)

    for rnd in range(1, 10):
        _sub_bytes(s)
        emit(f"r{rnd}_sub_bytes", rnd)
```

```
        if opts.disable_shift_rows:
            pass
        else:
            _shift_rows(s)
        emit(f"r{rnd}_shift_rows", rnd)
        _mix_columns(s, opts.mix_matrix)
        emit(f"r{rnd}_mix_columns", rnd)
        _add_round_key(s, rk[rnd])
        emit(f"r{rnd}_add_round_key", rnd)

    _sub_bytes(s)
    emit("r10_sub_bytes", 10)
    if not opts.disable_shift_rows:
        _shift_rows(s)
    emit("r10_shift_rows", 10)
    _add_round_key(s, rk[10])
    emit("r10_add_round_key", 10)

    return _bytes_from_state(s)
```

Listing 4: `hamming_bits` and `collect_hamming_curve` (aligned step lists).

```
def hamming_bits(a: bytes, b: bytes) -> int:
    x = int.from_bytes(a, "big") ^ int.from_bytes(b, "big")
    return x.bit_count()


def hamming_bytes(a: bytes, b: bytes) -> int:
    return sum(1 for i in range(16) if a[i] != b[i])


def collect_hamming_curve(
    p0: bytes,
    p1: bytes,
    k0: bytes,
    k1: bytes,
    opts: TraceOpts | None = None,
) -> tuple[list[str], list[int]]:
    """Return (step_labels, bit_distances) comparing traces p0,k0 vs p1,k1."""
    seq_a: list[bytes] = []
    seq_b: list[bytes] = []

    def make_logger(bucket: list[bytes]):
        def _log(_lbl: str, _rnd: int, st: bytes) -> None:
            bucket.append(st)

        return _log

    encrypt_trace(p0, k0, opts, log=make_logger(seq_a))
    encrypt_trace(p1, k1, opts, log=make_logger(seq_b))
    labels: list[str] = []
    bits: list[int] = []
    for i, (sa, sb) in enumerate(zip(seq_a, seq_b, strict=True)):
        labels.append(f"s{i}")
        bits.append(hamming_bits(sa, sb))
    return labels, bits
```

Listing 5: Alternate MixColumns matrix M_{new} (`M_NEW`).

```
# M_new: invertible circulant-style alternative (assignment matrix not in sidecar; documented in
    JSON)
M_NEW = [[3, 2, 1, 1], [1, 3, 2, 1], [1, 1, 3, 2], [2, 1, 1, 3]]
```

Recorded inputs (`q1-summary.json`). The archived run uses `plaintext_pair` `P_hex` `0000...00` and `P_prime_hex` `0100...00` (one-bit flip in byte 0). Aggregated metrics appear under `hamming_scenarios` (Parts A–C, plaintext flip) and `part_d_key_flip` (Part D). Timestamps and `mix_matrix_M_new` match the matrix in Listing 5.

Part A (standard pipeline, plaintext flip) — observation. Figure 1 plots `part_a_standard`: maximum intermediate Hamming distance **75** bits, final ciphertext distance **65** bits. The curve leaves the single-bit baseline within the first round stack, consistent with rapid mixing under H1.

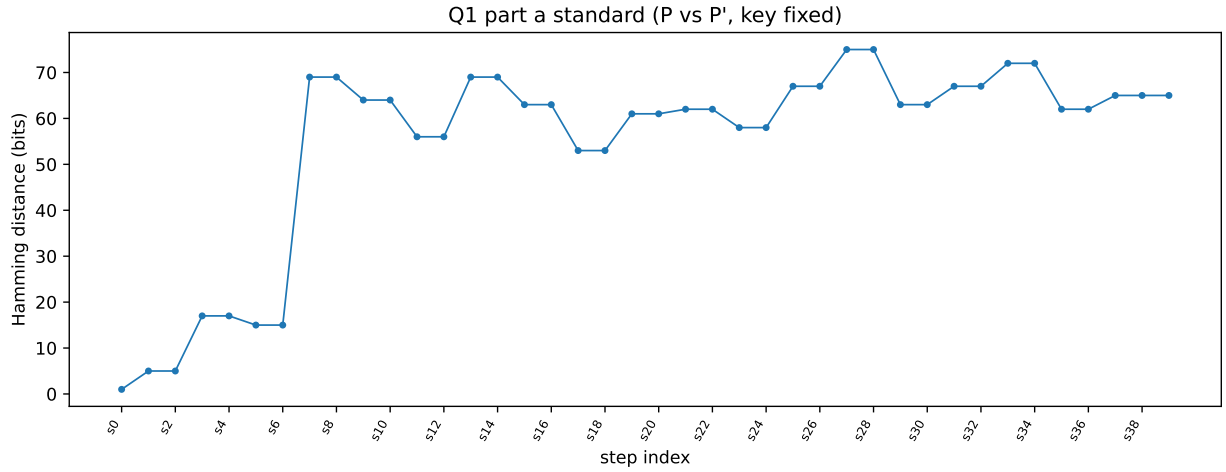

Figure 1: Question 1, Part A: bit Hamming distance vs. trace step (standard ShiftRows and MixColumns). Source: `output/diagrams/q1-hamming-rounds.pdf`.

Part B (ShiftRows off, plaintext flip) — observation. Figure 2 shows `part_b_no_shiftrows`: **max = 18**, **final = 13**. Compared to Part A, diffusion is *severely weakened*; H1 is strongly supported for the ShiftRows ablation.

Part C (M_{new}, plaintext flip) — observation. Figure 3 shows `part_c_custom_mix`: **max = 73**, **final = 63**. Peaks and finals differ from Part A (75 / 65) but remain far above Part B; the alternate matrix *reshapes* the Hamming trajectory without collapsing avalanche—also consistent with H1.

Part D (key-bit flip, fixed plaintext) — observation. Table 3 summarizes `part_d_key_flip`. Standard pipeline `d_standard` reaches a peak of **76** bits and final **62**; with ShiftRows off, peak **73** and final **62**; with M_{new}, peak **78** and final **76**. Unlike the plaintext-flip case, the custom-mix key-flip run ends at **76** final bits versus **62** for the nominal pipeline—key diffusion interacts differently with M_{new} on this fixed P. ShiftRows-off key flips still show substantially higher peaks than plaintext-flip Part B (73 vs. 18), so row motion is not the only source of key sensitivity.

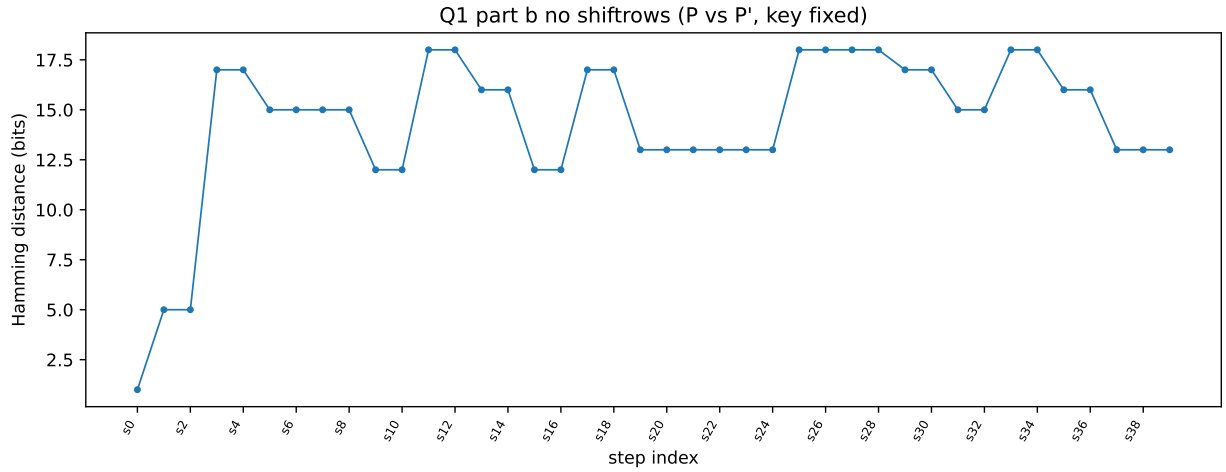

Figure 2: Question 1, Part B: ShiftRows disabled. Source: `output/diagrams/q1-ablation-shiftrows.pdf`.

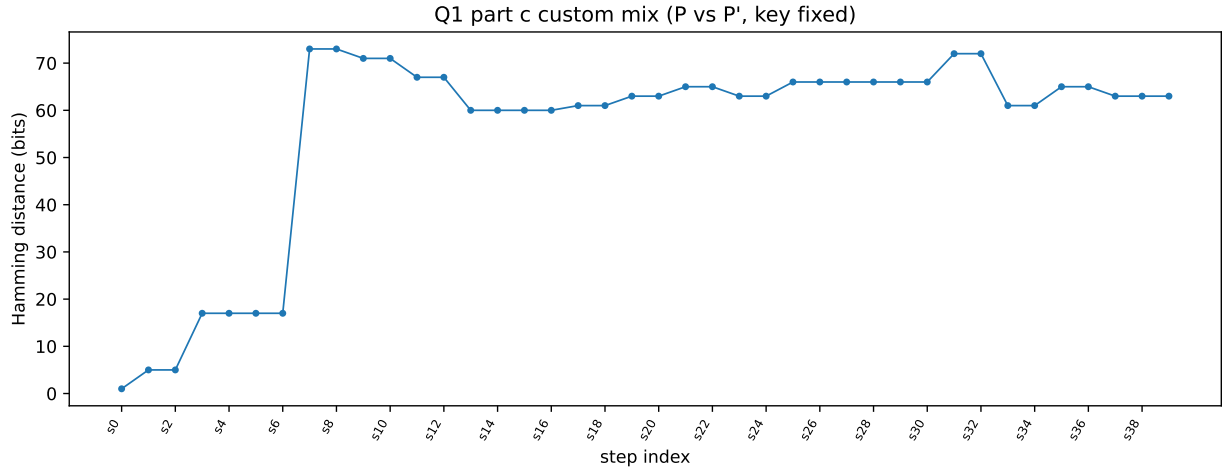

Figure 3: Question 1, Part C: `M_NEW` MixColumns with ShiftRows on. Source: `output/diagrams/q1-ablation-mixcolumns.pdf`.

Table 2: Question 1, Parts A–C: plaintext one-bit perturbation (from `output/results/q1-summary.json`).

Part	JSON block	Max. intermediate	Final ciphertext
A	`part_a_standard`	75	65
B	`part_b_no_shiftrows`	18	13
C	`part_c_custom_mix`	73	63

Table 3: Question 1, Part D: one key-bit flip (from `q1-summary.json`).

Scenario (`part_d_key_flip`)	**Max. intermediate**	**Final ciphertext**
Standard (`d_standard`)	76	62
ShiftRows disabled (`d_no_shiftrows`)	73	62
M_{new} (`d_custom_mix`)	78	76

Synthesis. RQ1 is answered qualitatively by Figures 1–3 and Tables 2–3: step-resolved Hamming distance grows under the nominal pipeline, collapses when ShiftRows is removed for plaintext flips, and shifts under M_{new}; key-bit flips produce a distinct family of curves with higher residual finals for `d_custom_mix`. H1 holds for **weakening** under ShiftRows ablation and for **reshaping** under M_{new} on plaintext flips.

Optional extension. The ten-plaintext variant would average curves over multiple (P, P') pairs via an outer loop around `collect_hamming_curve`.

Question 2: Modes on structured data (ECB vs. CBC on a BMP)

Research question (RQ2). Under OpenSSL AES-128-ECB versus AES-128-CBC with a random key and IV on a BMP, how does **visual structure** in the ciphertext (or in header-spliced previews) differ, and what does that imply about **mode choice** for data with redundancy?

Hypothesis (H2). **ECB** ciphertext *preserves* coarse image structure in byte-as-grayscale previews; **CBC** with a random IV yields visually *noisier* patterns at the same resolution; **splicing** the plaintext BMP header makes file-format viewing faithful while *leaving ECB macro-structure interpretable.*

Experiment (E2). Using OpenSSL `enc`, encrypt `assets/Secret.bmp` with AES-128-ECB and AES-128-CBC (`-nosalt`), drawing a random 16-byte key and (for CBC) IV. If the asset is absent, the driver generates a minimal gradient BMP. Produce two families of figures: (i) ciphertext *bodies* rendered as grayscale grids, and (ii) the same bodies with the 54-byte *plaintext* BMP header prepended so decoders interpret dimensions and bit depth correctly. Key and IV material for replay appear in `output/results/q2/openssl_run.json` (handle per your data policy).

Observations. Figure 4 compares raw ciphertext-body previews: ECB (left) still suggests large-scale spatial organization, whereas CBC (right) looks comparatively like high-frequency noise at this visualization scale. Figure 5 repeats the comparison after header repair; viewers can open the files as BMPs while the ECB run continues to exhibit block-aligned structure and the CBC run does not.

Findings. RQ2 is answered qualitatively by Figures 4–5: redundancy in uncompressed image data creates repeated or related 16-byte AES blocks, so **ECB is a poor choice** for concealing image content—previews remain structurally suggestive unless one changes the threat model (e.g., compresses or uses a mode that mixes blocks). **CBC with a random IV** breaks the direct block-for-block mapping and produces previews that look far less organized, illustrating why modes that

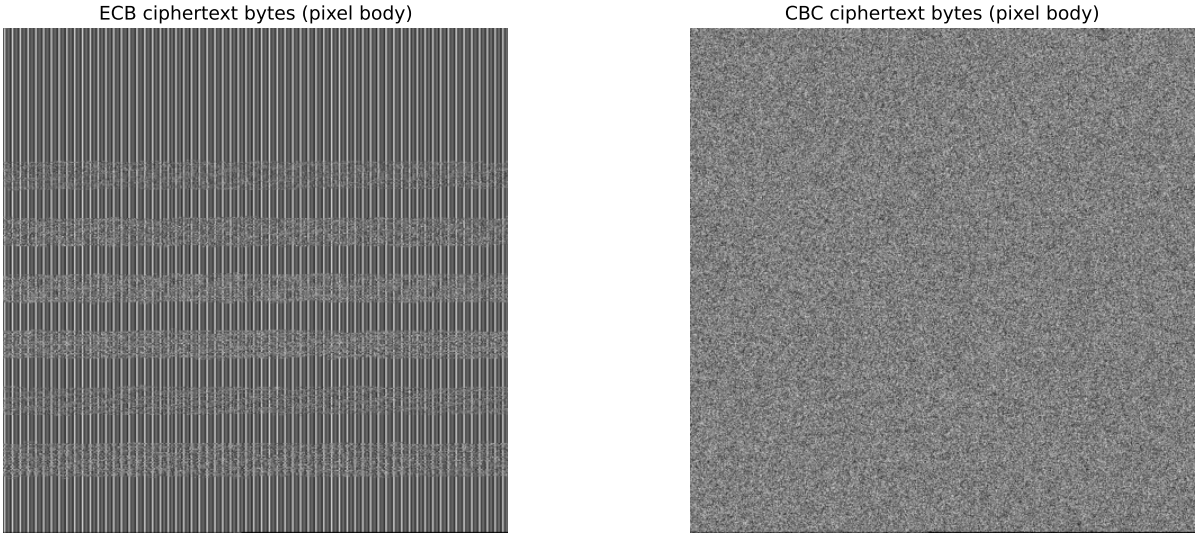

Figure 4: Question 2: ciphertext-body previews (bytes as grayscale). Left: ECB (`output/diagrams/q2-ecb-noise.pdf`). Right: CBC (`output/diagrams/q2-cbc-noise.pdf`).

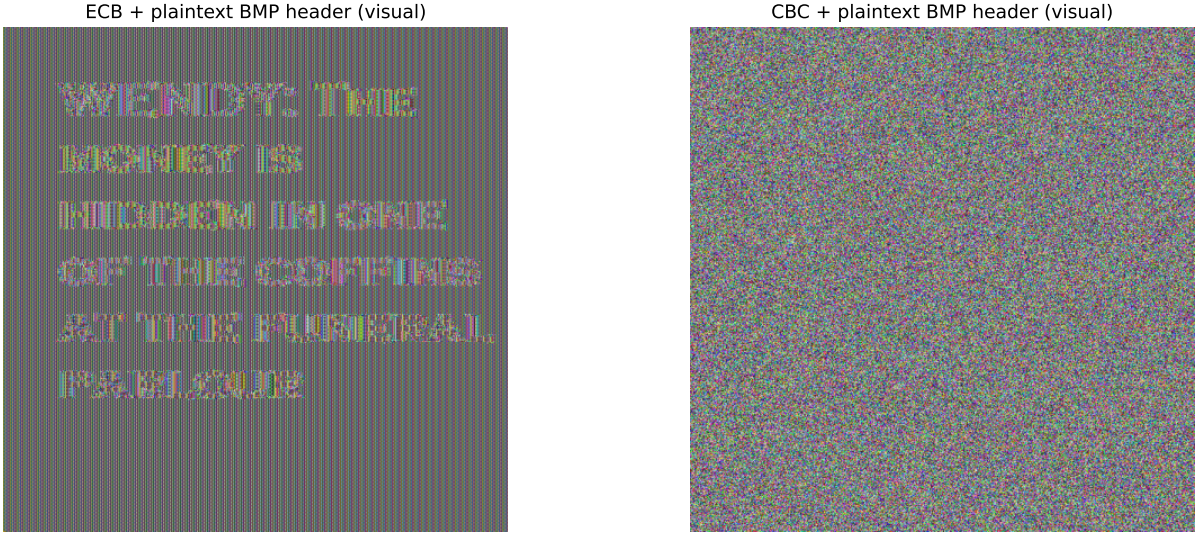

Figure 5: Question 2: plaintext BMP header (54 bytes) spliced onto ciphertext bodies. Left: ECB (`output/diagrams/q2-header-fixed-ecb.pdf`). Right: CBC (`output/diagrams/q2-header-fixed-cbc.pdf`).

chain or randomize per message are preferred for bulk data with spatial structure. This aligns with H2.

For **authoritative guidance**, NIST SP 800-38A states that in ECB, under a fixed key, any plaintext block always encrypts to the same ciphertext block, and that "if this property is undesirable in a particular application, the ECB mode should not be used" (Dworkin, 2001). That property is what makes uncompressed-image ciphertext under ECB recognizable at a glance unless one adds per-message randomization (for example a random IV with CBC) or otherwise avoids independent block encryption. The wording was checked against the official CSRC edition at https://csrc.nist.gov/pubs/sp/800/38/a/final (Sec. 6.1, ECB mode).

Question 3: TLS negotiation (Wireshark / `tshark` pipeline)

Research question (RQ3). From **captured** (or **probed**) TLS handshakes across multiple browsers and sites, what cipher suites appear in **ClientHello** offers, what does the server select in **ServerHello**, and how do selections **cluster** across TLS 1.3 versus TLS 1.2?

Hypothesis (H3). In contemporary captures, **most** handshakes negotiate TLS 1.3 with **AEAD** suites; **residual** TLS 1.2 flows show **ECDHE**-based key exchange with GCM or CBC-style names, consistent with deprecation of older stack-ups.

Experiment (E3). Record TCP and TLS handshakes; enumerate ClientHello cipher suites and classify (key exchange, authentication, bulk cipher, mode, MAC) where applicable; note TLS 1.3 versus 1.2; record ServerHello selection; repeat for two browsers and nine sites; summarize common algorithms and rationale. This repository automates PCAP export via `tshark` (primary path: `q3.pcapng` under `output/results/q3/`, or legacy `output/diagrams/q3-wireshark/`), emits one JSON file per SNI under `output/results/q3/`, and rolls negotiated protocol, cipher, frame numbers, and handshake counts into `output/results/q3-cipher-summary.json`. Optional twelve-family pruning is available via `scripts/q3_prune_orgs.py`. If no PCAP is present, `run_coursework_outputs.py` falls back to `openssl s_client` against public HTTPS hosts; that path records only the **negotiated** protocol and cipher (not the full ClientHello inventory), as documented in the summary JSON's `method` field. Full ClientHello offer lists and IANA-style decomposition remain in Wireshark on the capture and in `handshakes` blocks inside each per-SNI JSON file; Appendix tabulates offered-suite decomposition.

Summary artifact (`q3-cipher-summary.json`). Listing 6 shows the file header and the first eight `hosts` rows from the checked-in run: metadata (`generated_at`, `method`, `sni_count`), then negotiated `protocol`, `negotiated_cipher`, IANA-style `negotiated_cipher_suite_hex`, and handshake frame indices. The excerpt includes both TLS 1.3 AES-GCM selections and sample TLS 1.2 ECDHE + AES-GCM rows; the full file lists all summarized SNIs.

Listing 6: Excerpt from `output/results/q3-cipher-summary.json` (metadata and first eight `hosts`).


```
{
  "generated_at": "2026-04-05T04:47:10.993804+00:00",
  "method": "Pruned per-SNI JSON in output/results/q3/ to twelve org families (Google, Microsoft,
    Texas Tech, GitHub, Apple, Wikipedia/Wikimedia, EFF, Cloudflare, Cursor, Charter Spectrum, NSF
     Access CI, GoDaddy). Removed 16 host files; rebuilt summary from remaining JSON. Re-run
    export overwrites pruning.",
  "sni_count": 96,
```

```
"hosts": [
  {
    "host": "aadcdn.msauth.net",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_256_GCM_SHA384",
    "negotiated_cipher_suite_hex": "0x1302",
    "client_hello_frame": 50573,
    "server_hello_frame": 50602,
    "handshake_count": 13
  },
  {
    "host": "aadcdn.msftauth.net",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_256_GCM_SHA384",
    "negotiated_cipher_suite_hex": "0x1302",
    "client_hello_frame": 32641,
    "server_hello_frame": 32649,
    "handshake_count": 2
  },
  {
    "host": "accounts.google.com",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_128_GCM_SHA256",
    "negotiated_cipher_suite_hex": "0x1301",
    "client_hello_frame": 647,
    "server_hello_frame": 695,
    "handshake_count": 5
  },
  {
    "host": "agentn.global.api5.cursor.sh",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_128_GCM_SHA256",
    "negotiated_cipher_suite_hex": "0x1301",
    "client_hello_frame": 12766,
    "server_hello_frame": 12768,
    "handshake_count": 1
  },
  {
    "host": "ajax.googleapis.com",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_256_GCM_SHA384",
    "negotiated_cipher_suite_hex": "0x1302",
    "client_hello_frame": 33142,
    "server_hello_frame": 33167,
    "handshake_count": 1
  },
  {
    "host": "alive.github.com",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_128_GCM_SHA256",
    "negotiated_cipher_suite_hex": "0x1301",
    "client_hello_frame": 18059,
    "server_hello_frame": 18067,
    "handshake_count": 2
  },
  {
    "host": "allocations.access-ci.org",
    "protocol": "TLSv1.2",
    "negotiated_cipher": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
```

```
    "negotiated_cipher_suite_hex": "0xc02f",
    "client_hello_frame": 7000,
    "server_hello_frame": 7003,
    "handshake_count": 1
  },
  {
    "host": "android.clients.google.com",
    "protocol": "TLSv1.3",
    "negotiated_cipher": "TLS_AES_128_GCM_SHA256",
    "negotiated_cipher_suite_hex": "0x1301",
    "client_hello_frame": 1394,
    "server_hello_frame": 1409,
    "handshake_count": 5
  },
  {
    "host": "anon-stats.eff.org",
    "protocol": "TLSv1.2",
    "negotiated_cipher": "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
    "negotiated_cipher_suite_hex": "0xc02c",
    "client_hello_frame": 6974,
    "server_hello_frame": 6977,
    "handshake_count": 2
  },
```


ServerHello selection (aggregate). Figure 6 plots the frequency of *server-selected* cipher strings over the same summarized sample (capture-derived after pruning when applicable, or probe fallback). A wide table of the same per-host negotiated fields is also emitted as `output/diagrams/q3-client-hello` by the driver.

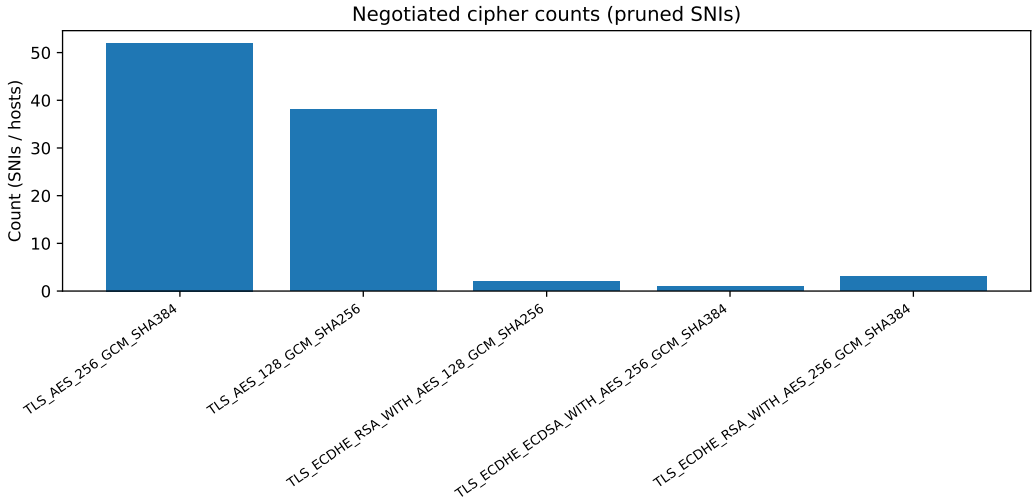

Figure 6: Question 3: frequency of negotiated (ServerHello) cipher strings. Source: `output/diagrams/q3-server-hello-summary.pdf`.

Findings. The JSON excerpt and Figure 6 support RQ3 and H3: TLS 1.3 rows dominate with `TLS_AES_128_GCM_SHA256` and `TLS_AES_256_GCM_SHA384` (AEAD); remaining TLS 1.2 rows in the

sample use ECDHE-based names with AES-GCM, matching expected modern baselines while older CBC-style names can still appear on some endpoints in larger captures.

Related Work

The Advanced Encryption Standard (AES) and Rijndael's design rationale are treated comprehensively by Daemen and Rijmen (2002); FIPS-197 specifies the standard (NIST, 2001). Modes of operation including CBC and ECB are catalogued by NIST SP 800-38A (Dworkin, 2001). TLS 1.2 framing is standardized in RFC 5246 (Dierks and Rescorla, 2008). Appendix summarizes implementation constants tied to our codebase.

Conclusion

We presented a unified, script-driven lab narrative for AES diffusion tracing (S-box, ShiftRows, MixColumns with optional M_{new}, key schedule, and logged states), for visual comparison of ECB and CBC on structured bitmap data, and for TLS summaries derived from `tshark` on a classroom PCAP (with a documented `s_client` fallback). The toy AES implementation in `toy_aes128_trace.py` is intentionally narrow—single-block, encrypt-only, validated against PyCryptodome for standard options—but it makes round-wise avalanche analysis inspectable. Future extensions include automated tables that expand every ClientHello cipher offer into fixed columns (KEX, auth, bulk, MAC) directly in the PDF, richer multi-plaintext Hamming statistics, and hardening guidance distinguishing pedagogy from production cryptography.

Reproducibility Statement

- **Code:** `scripts/toy_aes128_trace.py` (AES tracer, S-box, `xtime`, MixColumns including `M_NEW`, key expansion, `encrypt_trace`, Hamming helpers).
- **Driver:** `scripts/run_coursework_outputs.py` regenerates `output/results/*.json` and `output/diagrams/*.pdf`; requires Python 3 with `numpy`, `matplotlib`, `pycryptodome`, `Pillow`, and `openssl` on `PATH`. Q3 prefers `tshark` on `q3.pcapng`; if the capture is missing, Q3 uses live TLS via `openssl s_client`.
- **LaTeX:** Build from `src/` with `pdflatex main`, `bibtex main`, then `pdflatex main` twice. Bibliography source is `main.bib`; processed file `main.bbl` is produced by BibTeX.
- **Environment:** `conda activate rd-ralph-template` matches template defaults.

Ethics Statement

Scope and data. This report uses only **educational** materials: a toy AES-128 tracer, a course bitmap encrypted locally, and TLS **metadata** (cipher names, protocol versions, frame numbers) from a classroom packet capture or from short probes to public HTTPS services. No passwords, cookies, message bodies, or personally identifiable information are extracted or displayed.

Security claims. The implementation is not side-channel hardened and is not a substitute for audited, standards-track cryptography. ECB/CBC figures are **pedagogical**: they illustrate mode misuse and leakage; they are not recommended designs for protecting real data.

Secrets and publication. Regenerated OpenSSL key and IV values for Question 2 are stored under `output/results/q2/` for reproducibility. Before pushing to a **public** repository or sharing broadly, remove or redact those files if your instructor or institution requires it.

Network measurement. Live TLS fallback probes and PCAP-based analysis should follow course rules, applicable law, and reasonable respect for operator terms of service. Host lists and captures in this template are illustrative and may be replaced.

References

Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES—The Advanced Encryption Standard*. Springer, 2002. 2, 4, 17

T. Dierks and E. Rescorla. The transport layer security (TLS) protocol version 1.2. RFC 5246, 2008. 2, 5, 17

Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques. Special Publication 800-38A, National Institute of Standards and Technology, 2001. Revision 2007. 1, 4, 5, 6, 14, 17

NIST. Announcing the advanced encryption standard (AES). Federal Information Processing Standards Publication 197, National Institute of Standards and Technology, 2001. 5, 6, 17

Implementation constants and trace labels

S-box. The byte array `SBOX` in `toy_aes128_trace.py` matches the AES specification; SubBytes applies it to each state byte.

`xtime` and MixColumns. `xtime` realizes multiplication by x in GF(2^8) with AES modulus. Default MixColumns uses coefficients 2 and 3 via `xtime` and XOR. Generic coefficient multiplication uses a double-and-add style loop for other integers.

Alternate matrix M_{new}. Coursework Part C uses `M_NEW`:

$$M_{\text{new}} = \begin{pmatrix} 3 & 2 & 1 & 1 \\ 1 & 3 & 2 & 1 \\ 1 & 1 & 3 & 2 \\ 2 & 1 & 1 & 3 \end{pmatrix}$$

(row/column convention matches `mix_single_column` in code).

Logged step order. For each round, `encrypt_trace` emits states after SubBytes, ShiftRows (if enabled), MixColumns (rounds $1 \ldots 9$ only), and AddRoundKey, enabling paired Hamming curves from `collect_hamming_curve`.

ClientHello cipher offers (bulk, MAC)

Table 4 lists each suite name advertised in captured ClientHello messages (first handshake per SNI in this build), with heuristic columns for bulk cipher and MAC/integrity; key exchange and authentication appear in the IANA suite name where applicable (TLS 1.3 identifiers are AEAD-only). The table is set in `landscape` (rotated pages), uses the full landscape line width, and gives the largest column shares to **Host** and **Offered suite**. The **Sel.** column marks the ServerHello-selected suite for that handshake. Regenerate the fragment with: `python scripts/q3_offered_suites_longtable.py --primary-handshake-only -o output/article_iterations/q3_offered_suites_longtable_primary.tex`.

Table 4: ClientHello-offered TLS cipher suites from capture exports (heuristic bulk and MAC columns; IANA suite names encode KEX/auth where applicable). Sel. marks the ServerHello-selected suite for that handshake.

Host	Offered suite	Bulk	MAC	Sel.
aadcdn.msauth.net	GREASE_cipher(0xfafa)	–	–	
aadcdn.msauth.net	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
aadcdn.msauth.net	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
aadcdn.msauth.net	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
aadcdn.msauth.net	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msauth.net	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
aadcdn.msauth.net	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
aadcdn.msauth.net	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
aadcdn.msauth.net	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msauth.net	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
aadcdn.msauth.net	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
aadcdn.msauth.net	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
aadcdn.msauth.net	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
aadcdn.msauth.net	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msauth.net	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
aadcdn.msauth.net	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
aadcdn.msftauth.net	GREASE_cipher(0x1a1a)	–	–	
aadcdn.msftauth.net	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
aadcdn.msftauth.net	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
aadcdn.msftauth.net	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
aadcdn.msftauth.net	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
aadcdn.msftauth.net	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
aadcdn.msftauth.net	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
aadcdn.msftauth.net	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
aadcdn.msftauth.net	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
accounts.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
accounts.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
accounts.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
agentn.global.api5.cursor.sh	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
agentn.global.api5.cursor.sh	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
agentn.global.api5.cursor.sh	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
agentn.global.api5.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
agentn.global.api5.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
agentn.global.api5.cursor.sh	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	AES-128-CBC	HMAC-SHA256	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
agentn.global.api5.cursor.sh	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
agentn.global.api5.cursor.sh	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
agentn.global.api5.cursor.sh	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
agentn.global.api5.cursor.sh	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ajax.googleapis.com	GREASE_cipher(0xdada)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
ajax.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
ajax.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
ajax.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ajax.googleapis.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ajax.googleapis.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ajax.googleapis.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ajax.googleapis.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ajax.googleapis.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
alive.github.com	GREASE_cipher(0xbaba)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
alive.github.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
alive.github.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
alive.github.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
alive.github.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
alive.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
alive.github.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
alive.github.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
alive.github.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
alive.github.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
alive.github.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
alive.github.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
alive.github.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
alive.github.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
alive.github.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
alive.github.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
allocations.access-ci.org	GREASE_cipher(0xaaaa)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
allocations.access-ci.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
allocations.access-ci.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
allocations.access-ci.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
allocations.access-ci.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
allocations.access-ci.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
allocations.access-ci.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
allocations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
allocations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	✓
allocations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
allocations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
allocations.access-ci.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
allocations.access-ci.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
allocations.access-ci.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
allocations.access-ci.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
allocations.access-ci.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
android.clients.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
android.clients.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
android.clients.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
anon-stats.eff.org	GREASE_cipher(0x9a9a)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
anon-stats.eff.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
anon-stats.eff.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
anon-stats.eff.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
anon-stats.eff.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
anon-stats.eff.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	✓
anon-stats.eff.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
anon-stats.eff.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
anon-stats.eff.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
anon-stats.eff.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
anon-stats.eff.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
anon-stats.eff.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
anon-stats.eff.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
anon-stats.eff.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
anon-stats.eff.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
anon-stats.eff.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api-safari-ause2c.smoot.apple.com	GREASE_cipher(0x2a2a)	–	–	
api-safari-ause2c.smoot.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
api-safari-ause2c.smoot.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
api-safari-ause2c.smoot.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
api.apple-cloudkit.com	GREASE_cipher(0xcaca)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
api.apple-cloudkit.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
api.apple-cloudkit.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
api.apple-cloudkit.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
api.github.com	GREASE_cipher(0x0a0a)	–	–	
api.github.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
api.github.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
api.github.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
api.github.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api.github.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api.github.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.github.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.github.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api.github.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api.github.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
api.github.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.github.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.github.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.github.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api.individual.githubcopilot.com	GREASE_cipher(0x3a3a)	–	–	
api.individual.githubcopilot.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
api.individual.githubcopilot.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
api.individual.githubcopilot.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api.individual.githubcopilot.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api.individual.githubcopilot.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api.individual.githubcopilot.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
api.individual.githubcopilot.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api.individual.githubcopilot.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api.individual.githubcopilot.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api2.cursor.sh	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
api2.cursor.sh	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
api2.cursor.sh	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
api2.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api2.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api2.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api2.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api2.cursor.sh	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	AES-128-CBC	HMAC-SHA256	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
api2.cursor.sh	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
api2.cursor.sh	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
api2.cursor.sh	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
api2.cursor.sh	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
api2.cursor.sh	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
apple-relay.cloudflare.com	GREASE_cipher(0xbaba)	–	–	
apple-relay.cloudflare.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
apple-relay.cloudflare.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
apple-relay.cloudflare.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
avatars.githubusercontent.com	GREASE_cipher(0x8a8a)	–	–	
avatars.githubusercontent.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
avatars.githubusercontent.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
avatars.githubusercontent.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
avatars.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
avatars.githubusercontent.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
avatars.githubusercontent.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
avatars.githubusercontent.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
avatars.githubusercontent.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
blogs.msdn.com	GREASE_cipher(0x7a7a)	–	–	
blogs.msdn.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
blogs.msdn.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
blogs.msdn.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
blogs.msdn.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
blogs.msdn.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
blogs.msdn.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
blogs.msdn.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
blogs.msdn.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
blogs.msdn.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
blogs.msdn.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
browser.events.data.microsoft.com	GREASE_cipher(0x2a2a)	–	–	
browser.events.data.microsoft.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
browser.events.data.microsoft.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
browser.events.data.microsoft.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
browser.events.data.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
browser.events.data.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
browser.events.data.microsoft.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
browser.events.data.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
browser.events.data.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
browser.events.data.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
browser.events.data.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
browser.events.data.microsoft.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
browser.events.data.microsoft.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
browser.events.data.microsoft.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
browser.events.data.microsoft.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
browser.events.data.microsoft.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
camo.githubusercontent.com	GREASE_cipher(0xdada)	–	–	
camo.githubusercontent.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
camo.githubusercontent.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
camo.githubusercontent.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
camo.githubusercontent.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
camo.githubusercontent.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
camo.githubusercontent.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
camo.githubusercontent.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
camo.githubusercontent.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
camo.githubusercontent.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
cdn2.smoot.apple.com	GREASE_cipher(0x9a9a)	–	–	
cdn2.smoot.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
cdn2.smoot.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
cdn2.smoot.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
cdn2.smoot.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
cdn2.smoot.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
cdn2.smoot.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
cdn2.smoot.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
cdn2.smoot.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
cdn2.smoot.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clients1.google.com	GREASE_cipher(0xbaba)	–	–	
clients1.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
clients1.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
clients1.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clients1.google.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
clients1.google.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clients1.google.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
clients1.google.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clients1.google.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
clients1.google.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
clients1.google.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
clients1.google.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clients1.google.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
clients1.google.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clients4.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
clients4.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
clients4.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
clientservices.googleapis.com	GREASE_cipher(0x6a6a)	–	–	
clientservices.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
clientservices.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
clientservices.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
clientservices.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clientservices.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clientservices.googleapis.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
clientservices.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
clientservices.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clientservices.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
clientservices.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
clientservices.googleapis.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
clientservices.googleapis.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
clientservices.googleapis.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
clientservices.googleapis.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
clientservices.googleapis.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
collector.github.com	GREASE_cipher(0x1a1a)	–	–	
collector.github.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
collector.github.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
collector.github.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
collector.github.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
collector.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
collector.github.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
collector.github.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
collector.github.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
collector.github.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
collector.github.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
collector.github.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
collector.github.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
collector.github.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
collector.github.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
collector.github.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
configuration.ls.apple.com	GREASE_cipher(0x3a3a)	–	–	
configuration.ls.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
configuration.ls.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
configuration.ls.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
configuration.ls.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
configuration.ls.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
configuration.ls.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
configuration.ls.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
configuration.ls.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
configuration.ls.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
content-autofill.googleapis.com	GREASE_cipher(0xaaaa)	–	–	
content-autofill.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
content-autofill.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
content-autofill.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
content-autofill.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
content-autofill.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
content-autofill.googleapis.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
content-autofill.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
content-autofill.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
content-autofill.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
content-autofill.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
content-autofill.googleapis.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
content-autofill.googleapis.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
content-autofill.googleapis.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
content-autofill.googleapis.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
content-autofill.googleapis.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dcc.godaddy.com	GREASE_cipher(0xdada)	–	–	
dcc.godaddy.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
dcc.godaddy.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
dcc.godaddy.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dcc.godaddy.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dcc.godaddy.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
dcc.godaddy.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dcc.godaddy.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dcc.godaddy.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
devblogs.microsoft.com	GREASE_cipher(0x1a1a)	–	–	
devblogs.microsoft.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
devblogs.microsoft.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
devblogs.microsoft.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
devblogs.microsoft.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
devblogs.microsoft.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
devblogs.microsoft.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
devblogs.microsoft.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
devblogs.microsoft.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dns.google	GREASE_cipher(0x0a0a)	–	–	
dns.google	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
dns.google	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
dns.google	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
dns.google	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dns.google	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dns.google	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
dns.google	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dns.google	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dns.google	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
dns.google	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
dns.google	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
dns.google	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
dns.google	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
dns.google	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
dns.google	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-01.spectrum.com	GREASE_cipher(0xdada)	–	–	
doh-01.spectrum.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
doh-01.spectrum.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
doh-01.spectrum.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
doh-01.spectrum.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-01.spectrum.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-01.spectrum.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
doh-01.spectrum.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
doh-01.spectrum.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-01.spectrum.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
doh-01.spectrum.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-01.spectrum.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
doh-01.spectrum.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
doh-01.spectrum.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-01.spectrum.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
doh-01.spectrum.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-02.spectrum.com	GREASE_cipher(0x4a4a)	–	–	
doh-02.spectrum.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
doh-02.spectrum.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
doh-02.spectrum.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
doh-02.spectrum.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-02.spectrum.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-02.spectrum.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
doh-02.spectrum.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
doh-02.spectrum.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-02.spectrum.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
doh-02.spectrum.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
doh-02.spectrum.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
doh-02.spectrum.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
doh-02.spectrum.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
doh-02.spectrum.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
doh-02.spectrum.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
en.wikipedia.org	GREASE_cipher(0x5a5a)	–	–	
en.wikipedia.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
en.wikipedia.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
en.wikipedia.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
en.wikipedia.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
en.wikipedia.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
en.wikipedia.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
en.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
en.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
en.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
en.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
en.wikipedia.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
en.wikipedia.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
en.wikipedia.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
en.wikipedia.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
en.wikipedia.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
encrypted-tbn0.gstatic.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
encrypted-tbn0.gstatic.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
encrypted-tbn0.gstatic.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
fonts.googleapis.com	GREASE_cipher(0x3a3a)	–	–	
fonts.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
fonts.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
fonts.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fonts.googleapis.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fonts.googleapis.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fonts.googleapis.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fonts.googleapis.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fonts.googleapis.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fonts.gstatic.com	GREASE_cipher(0x9a9a)	–	–	
fonts.gstatic.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
fonts.gstatic.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
fonts.gstatic.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
fortawesome.github.io	GREASE_cipher(0x2a2a)	–	–	
fortawesome.github.io	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
fortawesome.github.io	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
fortawesome.github.io	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fortawesome.github.io	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fortawesome.github.io	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fortawesome.github.io	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fortawesome.github.io	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fortawesome.github.io	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fpinit.itunes.apple.com	GREASE_cipher(0xbaba)	–	–	
fpinit.itunes.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
fpinit.itunes.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
fpinit.itunes.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fpinit.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
fpinit.itunes.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
fpinit.itunes.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
fpinit.itunes.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
fpinit.itunes.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
gateway.icloud.com	GREASE_cipher(0x0a0a)	–	–	
gateway.icloud.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
gateway.icloud.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
gateway.icloud.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
gdmf.apple.com	GREASE_cipher(0x8a8a)	–	–	
gdmf.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
gdmf.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
gdmf.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
gdmf.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
gdmf.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
gdmf.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
gdmf.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
gdmf.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
gemini.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
gemini.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
gemini.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
github.githubassets.com	GREASE_cipher(0x1a1a)	–	–	
github.githubassets.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
github.githubassets.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
github.githubassets.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
github.githubassets.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
github.githubassets.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
github.githubassets.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
github.githubassets.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
github.githubassets.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
github.githubassets.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
github.githubassets.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
github.githubassets.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
github.githubassets.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
github.githubassets.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
identity.nel.measure.office.net	GREASE_cipher(0x1a1a)	–	–	
identity.nel.measure.office.net	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
identity.nel.measure.office.net	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
identity.nel.measure.office.net	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
identity.nel.measure.office.net	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
identity.nel.measure.office.net	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
identity.nel.measure.office.net	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
identity.nel.measure.office.net	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
identity.nel.measure.office.net	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
identity.nel.measure.office.net	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
identity.nel.measure.office.net	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
identity.nel.measure.office.net	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
identity.nel.measure.office.net	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
identity.nel.measure.office.net	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
identity.nel.measure.office.net	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
identity.nel.measure.office.net	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
idp.shibboleth.ttu.edu	GREASE_cipher(0x6a6a)	–	–	
idp.shibboleth.ttu.edu	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
idp.shibboleth.ttu.edu	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
idp.shibboleth.ttu.edu	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
idp.shibboleth.ttu.edu	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	✓
idp.shibboleth.ttu.edu	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
idp.shibboleth.ttu.edu	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
idp.shibboleth.ttu.edu	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
idp.shibboleth.ttu.edu	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
init.itunes.apple.com	GREASE_cipher(0xbaba)	–	–	
init.itunes.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
init.itunes.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
init.itunes.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
init.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
init.itunes.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
init.itunes.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
init.itunes.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
init.itunes.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
itunes.apple.com	GREASE_cipher(0x9a9a)	–	–	
itunes.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
itunes.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
itunes.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
itunes.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
itunes.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
itunes.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
itunes.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
itunes.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
itunes.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
itunes.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
itunes.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
learn.microsoft.com	GREASE_cipher(0xcaca)	–	–	
learn.microsoft.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
learn.microsoft.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
learn.microsoft.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
learn.microsoft.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
learn.microsoft.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
learn.microsoft.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
learn.microsoft.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
learn.microsoft.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
lensfrontend-pa.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
lensfrontend-pa.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
lensfrontend-pa.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
lh3.googleusercontent.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
lh3.googleusercontent.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
lh3.googleusercontent.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
login.microsoft.com	GREASE_cipher(0x0a0a)	–	–	
login.microsoft.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
login.microsoft.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
login.microsoft.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
login.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
login.microsoft.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
login.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
login.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
login.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
login.microsoft.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
login.microsoft.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
login.microsoft.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoft.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
login.microsoft.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
login.microsoftonline.com	GREASE_cipher(0x6a6a)	–	–	
login.microsoftonline.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
login.microsoftonline.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
login.microsoftonline.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
login.microsoftonline.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
login.microsoftonline.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
login.microsoftonline.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
login.microsoftonline.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
login.microsoftonline.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
login.microsoftonline.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mask-api.icloud.com	GREASE_cipher(0xbaba)	–	–	
mask-api.icloud.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
mask-api.icloud.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
mask-api.icloud.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mask-api.icloud.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mask-api.icloud.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
mask-api.icloud.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
mask-api.icloud.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
mask-api.icloud.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
mask-api.icloud.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mask.icloud.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
memex-pa.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
memex-pa.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
memex-pa.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
mtalk.google.com	GREASE_cipher(0x8a8a)	–	–	
mtalk.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
mtalk.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
mtalk.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
mtalk.google.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
mtalk.google.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mtalk.google.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
mtalk.google.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
mtalk.google.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
mtalk.google.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
mtalk.google.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
mtalk.google.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
mtalk.google.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
mtalk.google.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
mtalk.google.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
mtalk.google.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
musicstatus.itunes.apple.com	GREASE_cipher(0x2a2a)	–	–	
musicstatus.itunes.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
musicstatus.itunes.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
musicstatus.itunes.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
musicstatus.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
musicstatus.itunes.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
musicstatus.itunes.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
musicstatus.itunes.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
musicstatus.itunes.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
oauthaccountmanager.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
oauthaccountmanager.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
oauthaccountmanager.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ocsp2.apple.com	GREASE_cipher(0x4a4a)	–	–	
ocsp2.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
ocsp2.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
ocsp2.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ocsp2.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
ocsp2.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ocsp2.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ocsp2.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ocsp2.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ogads-pa.clients6.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
ogads-pa.clients6.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
ogads-pa.clients6.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ogs.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
ogs.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
ogs.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
operations.access-ci.org	GREASE_cipher(0xcaca)	–	–	
operations.access-ci.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
operations.access-ci.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
operations.access-ci.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
operations.access-ci.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
operations.access-ci.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
operations.access-ci.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
operations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
operations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
operations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
operations.access-ci.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
operations.access-ci.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
operations.access-ci.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
operations.access-ci.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
operations.access-ci.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
operations.access-ci.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
optimizationguide-pa.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
optimizationguide-pa.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
optimizationguide-pa.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
outlook.office365.com	GREASE_cipher(0x3a3a)	–	–	
outlook.office365.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
outlook.office365.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
outlook.office365.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
outlook.office365.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
outlook.office365.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
outlook.office365.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
outlook.office365.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
outlook.office365.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
outlook.office365.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
outlook.office365.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
outlook.office365.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
outlook.office365.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
outlook.office365.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-contacts.icloud.com	GREASE_cipher(0x6a6a)	–	–	
p142-contacts.icloud.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
p142-contacts.icloud.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
p142-contacts.icloud.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-contacts.icloud.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-contacts.icloud.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p142-contacts.icloud.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p142-contacts.icloud.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-contacts.icloud.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-quota.icloud.com	GREASE_cipher(0x7a7a)	–	–	
p142-quota.icloud.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
p142-quota.icloud.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
p142-quota.icloud.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-quota.icloud.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p142-quota.icloud.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p142-quota.icloud.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p142-quota.icloud.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p142-quota.icloud.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p70-buy.itunes.apple.com	GREASE_cipher(0x6a6a)	–	–	
p70-buy.itunes.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
p70-buy.itunes.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
p70-buy.itunes.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p70-buy.itunes.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
p70-buy.itunes.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
p70-buy.itunes.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
p70-buy.itunes.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
p70-buy.itunes.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
passwordsleakcheck-pa.googleapis.com	GREASE_cipher(0x7a7a)	–	–	
passwordsleakcheck-pa.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
passwordsleakcheck-pa.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
passwordsleakcheck-pa.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
passwordsleakcheck-pa.googleapis.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
passwordsleakcheck-pa.googleapis.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
passwordsleakcheck-pa.googleapis.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
passwordsleakcheck-pa.googleapis.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
passwordsleakcheck-pa.googleapis.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
play.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
play.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
play.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
play.googleapis.com	GREASE_cipher(0x8a8a)	–	–	
play.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
play.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
play.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
portal.azure.com	GREASE_cipher(0x5a5a)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
portal.azure.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
portal.azure.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
portal.azure.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
portal.azure.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
portal.azure.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
portal.azure.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
portal.azure.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
portal.azure.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
portal.azure.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
portal.azure.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
portal.azure.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
portal.azure.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
portal.azure.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
repo42.cursor.sh	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
repo42.cursor.sh	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
repo42.cursor.sh	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
repo42.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
repo42.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
repo42.cursor.sh	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	AES-128-CBC	HMAC-SHA256	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
repo42.cursor.sh	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
repo42.cursor.sh	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
repo42.cursor.sh	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
repo42.cursor.sh	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemetrics.apple.com	GREASE_cipher(0xcaca)	–	–	
securemetrics.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
securemetrics.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
securemetrics.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemetrics.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemetrics.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
securemetrics.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemetrics.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemetrics.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemvt.apple.com	GREASE_cipher(0xfafa)	–	–	
securemvt.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
securemvt.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
securemvt.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemvt.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securemvt.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
securemvt.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
securemvt.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
securemvt.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
securitydomain-pa.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
securitydomain-pa.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
securitydomain-pa.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
signaler-pa.clients6.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
signaler-pa.clients6.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
signaler-pa.clients6.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ssl.gstatic.com	GREASE_cipher(0x3a3a)	–	–	
ssl.gstatic.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
ssl.gstatic.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
ssl.gstatic.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ssl.gstatic.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ssl.gstatic.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ssl.gstatic.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ssl.gstatic.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ssl.gstatic.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ssl.gstatic.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ssl.gstatic.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ssl.gstatic.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ssl.gstatic.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ssl.gstatic.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
ssl.gstatic.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ssl.gstatic.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swdist.apple.com	GREASE_cipher(0xaaaa)	–	–	
swdist.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
swdist.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
swdist.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swdist.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
swdist.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
swdist.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swdist.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swdist.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
swdist.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swdist.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swdist.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
swdist.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swdist.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swscan.apple.com	GREASE_cipher(0x3a3a)	–	–	
swscan.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
swscan.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
swscan.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swscan.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
swscan.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
swscan.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swscan.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
swscan.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
swscan.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
swscan.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
swscan.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
swscan.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
swscan.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
texastech.instructure.com	GREASE_cipher(0x8a8a)	–	–	
texastech.instructure.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
texastech.instructure.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
texastech.instructure.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
texastech.instructure.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
texastech.instructure.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
texastech.instructure.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
texastech.instructure.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
texastech.instructure.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ttusystem.myplannedgift.org	GREASE_cipher(0x6a6a)	–	–	
ttusystem.myplannedgift.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
ttusystem.myplannedgift.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
ttusystem.myplannedgift.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ttusystem.myplannedgift.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
ttusystem.myplannedgift.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
ttusystem.myplannedgift.org	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
ttusystem.myplannedgift.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
ttusystem.myplannedgift.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
update.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
update.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
update.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
upload.wikimedia.org	GREASE_cipher(0xaaaa)	–	–	
upload.wikimedia.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
upload.wikimedia.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
upload.wikimedia.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
upload.wikimedia.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
upload.wikimedia.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
upload.wikimedia.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
upload.wikimedia.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
upload.wikimedia.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
upload.wikimedia.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
upload.wikimedia.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
upload.wikimedia.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
upload.wikimedia.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
upload.wikimedia.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
upload.wikimedia.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
upload.wikimedia.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
us-only.gcpp.cursor.sh	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
us-only.gcpp.cursor.sh	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
us-only.gcpp.cursor.sh	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
us-only.gcpp.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
us-only.gcpp.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
us-only.gcpp.cursor.sh	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	AES-128-CBC	HMAC-SHA256	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
us-only.gcpp.cursor.sh	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
us-only.gcpp.cursor.sh	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
us-only.gcpp.cursor.sh	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
us-only.gcpp.cursor.sh	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
visualstudiogallery.msdn.microsoft.com	GREASE_cipher(0x3a3a)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
visualstudiogallery.msdn.microsoft.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
visualstudiogallery.msdn.microsoft.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
visualstudiogallery.msdn.microsoft.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	✓
visualstudiogallery.msdn.microsoft.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
visualstudiogallery.msdn.microsoft.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
visualstudiogallery.msdn.microsoft.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
visualstudiogallery.msdn.microsoft.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
vpn.ttu.edu	GREASE_cipher(0xfafa)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
vpn.ttu.edu	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
vpn.ttu.edu	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
vpn.ttu.edu	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
vpn.ttu.edu	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	✓
vpn.ttu.edu	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
vpn.ttu.edu	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
vpn.ttu.edu	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
vpn.ttu.edu	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
waa-pa.clients6.google.com	GREASE_cipher(0x9a9a)	–	–	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
waa-pa.clients6.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
waa-pa.clients6.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
waa-pa.clients6.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
waa-pa.clients6.google.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
waa-pa.clients6.google.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
waa-pa.clients6.google.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
waa-pa.clients6.google.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
waa-pa.clients6.google.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
waa-pa.clients6.google.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
waa-pa.clients6.google.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
waa-pa.clients6.google.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
waa-pa.clients6.google.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
waa-pa.clients6.google.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
waa-pa.clients6.google.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
waa-pa.clients6.google.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
weblogs.asp.net	GREASE_cipher(0x3a3a)	–	–	
weblogs.asp.net	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
weblogs.asp.net	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
weblogs.asp.net	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
weblogs.asp.net	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
weblogs.asp.net	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
weblogs.asp.net	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
weblogs.asp.net	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
weblogs.asp.net	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
wps.apple.com	GREASE_cipher(0xcaca)	–	–	
wps.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
wps.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
wps.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
wps.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
wps.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
wps.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
wps.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
wps.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
wps.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
wps.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
wps.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
wps.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
wps.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
wps.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
wps.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
wps.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
wps.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
wps.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
wps.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
wps.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.apple.com	GREASE_cipher(0x4a4a)	–	–	
www.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
www.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
www.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
www.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
www.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
www.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.apple.com	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.apple.com	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.apple.com	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.eff.org	GREASE_cipher(0x8a8a)	–	–	
www.eff.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
www.eff.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
www.eff.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.eff.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
www.eff.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.eff.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.eff.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.eff.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	✓
www.eff.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.eff.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.eff.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.eff.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.eff.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.eff.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.eff.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.google.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
www.google.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
www.google.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.googleapis.com	GREASE_cipher(0x6a6a)	–	–	
www.googleapis.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
www.googleapis.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
www.googleapis.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.googletagmanager.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
www.googletagmanager.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
www.googletagmanager.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.gstatic.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
www.gstatic.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
www.gstatic.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.texastech.edu	GREASE_cipher(0x3a3a)	–	–	
www.texastech.edu	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
www.texastech.edu	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
www.texastech.edu	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.texastech.edu	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.texastech.edu	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
www.texastech.edu	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.texastech.edu	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.texastech.edu	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
www.texastech.edu	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.texastech.edu	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.texastech.edu	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
www.texastech.edu	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.texastech.edu	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.texastech.edu	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.texastech.edu	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.wikipedia.org	GREASE_cipher(0x7a7a)	–	–	
www.wikipedia.org	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	✓
www.wikipedia.org	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	
www.wikipedia.org	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
www.wikipedia.org	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.wikipedia.org	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.wikipedia.org	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
www.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.wikipedia.org	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
www.wikipedia.org	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
www.wikipedia.org	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	Offered suite	Bulk	MAC	Sel.
www.wikipedia.org	TLS_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
www.wikipedia.org	TLS_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
www.wikipedia.org	TLS_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
xp.apple.com	GREASE_cipher(0xeaea)	–	–	
xp.apple.com	TLS_AES_128_GCM_SHA256	AES-128-GCM	AEAD (SHA-256)	
xp.apple.com	TLS_AES_256_GCM_SHA384	AES-256-GCM	AEAD (SHA-384)	✓
xp.apple.com	TLS_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD (SHA-256)	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
xp.apple.com	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
xp.apple.com	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	AES-128-GCM	AEAD + SHA-256 PRF	
xp.apple.com	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	AES-256-CBC	HMAC-SHA1	
xp.apple.com	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	AES-256-GCM	AEAD + SHA-384 PRF	
xp.apple.com	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	ChaCha20-Poly1305	AEAD	
xp.apple.com	TLS_RSA_WITH_3DES_EDE_CBC_SHA	3DES-EDE-CBC	HMAC-SHA1	
xp.apple.com	TLS_RSA_WITH_AES_128_CBC_SHA	AES-128-CBC	HMAC-SHA1	

Continued on next page

Table 4 – *continued*

Host	**Offered suite**	**Bulk**	**MAC**	**Sel.**
xp.apple.com	`TLS_RSA_WITH_AES_128_GCM_SHA256`	AES-128-GCM	AEAD + SHA-256 PRF	
xp.apple.com	`TLS_RSA_WITH_AES_256_CBC_SHA`	AES-256-CBC	HMAC-SHA1	
xp.apple.com	`TLS_RSA_WITH_AES_256_GCM_SHA384`	AES-256-GCM	AEAD + SHA-384 PRF	

Authorship, reproducibility, and generative AI

Course context. This document supports CS 6343 (Spring 2026) at Texas Tech University. Figures and numbers are tied to scripts and JSON under `output/` in the `rd-ralph-template` repository; regenerate them with `python scripts/run_coursework_outputs.py` before claiming a specific machine's artifacts.

Human review. Technical content (AES structure, modes, TLS summaries) was checked against FIPS-197, SP 800-38A, and the cited RFCs, and against the Python and shell drivers referenced in the main text.

Generative AI. An LLM-based coding assistant may have been used for drafting prose and LaTeX structure; numerical results, file paths, and security claims were not accepted without verification against the repository outputs and standards. If your course requires a stricter disclosure, replace this paragraph with the instructor's template.