# The LogJam Attack

## Cracking 512-bit DHE

February 14, 2019

**Mark Schultz**
UC San Diego

Beamer Template (and some pictures) by Felix Günther

# Outline

- Review Diffie-Hellman (DH) key exchange
- Define the Attack Model for LogJam
- The Computational Diffie-Hellman (CDH) and Discrete Logarithm (DL) problems
- The Number Field Sieve (NFS)
- Estimates of the wide-scale applicability of the attack
- Strategies to protect against it

# DH Key Exchange

## Basic Model

Diffie-Hellman Key Exchange

| Step | Alice | Bob |
|---|---|---|
| 1 | Parameters: $p$, $g$ | |
| 2 | $A = \text{random}()$<br>$a = g^A \pmod p$ | $\text{random}() = B$<br>$g^B \pmod p = b$ |
| 3 | $a \longrightarrow$<br>$\longleftarrow b$ | |
| 4 | $K = g^{BA} \pmod p = b^A \pmod p$ | $a^B \pmod p = g^{AB} \pmod p = K$ |
| 5 | $\longleftarrow E_K(data) \longrightarrow$ | |

# DH Key Exchange

Slightly less basic details

- The Server will have access to some key pair $(pk, sk)$
- The $pk$ is signed by a Certificate Authority
- Two (main) variants of Diffie-Hellman:
  - DH: The key pair is a Diffie-Hellman one $pk = (p, g, g^B)$, $sk = (p, g, B)$
  - DHE: The key pair is Digital Signature (RSA) key pair
    - Used to sign a freshly-generated DH key pair
- DHE provides forward secrecy over DH

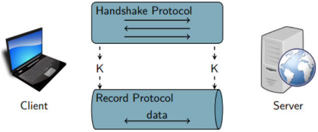
# Attack Model

TLS Recap

## TLS Protocol

- Handshake (Key Exchange)
- ~~Record Protocol~~ (Authenticated Encryption)

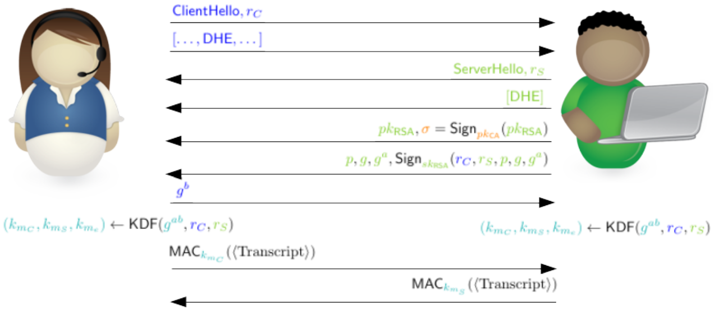
>>



# Attack Model

## Honest TLS Handshake

UC San Diego

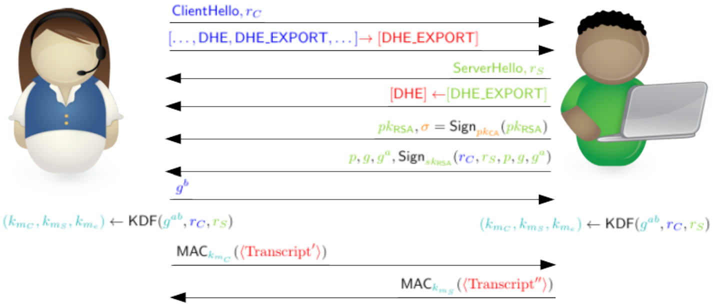



Color Coding:

- Blue: Client
- Green: Server
- Orange: Certificate Authority
- BlueGreen: Client & Server

# Attack Model

## MITM TLS Handshake




Color Coding:

- Blue: Client
- Green: Server
- Orange: Certificate Authority
- BlueGreen: Client & Server
- Red: Adversary

# Attack Model

## Downgrade Discussion



- Reason for vulnerability: Server's cipher choice is not signed
- Requires weak Key Exchange (KE) both Client and Server can use
- Is there weak crypto in TLS? Yes, in Export-Grade Crypto

# Attack Model

## Export-Grade Crypto

UC San Diego

- Cold War led to Export Controls
- Separate controls for Commercial products and Munitions
  - Cryptography classified as a munition
  - Limited export (for asymmetric crypto) to 512-bit keys (2048-bit currently used)
  - Kept in the protocol as most servers will never request it, and backwards compatibility

# CDH and DL

## Computational Diffie-Hellman Assumption

- Input: ($p$ prime, $g$ generator), $\mathbb{G} = \langle g \rangle \leq (\mathbb{Z}/p\mathbb{Z})^\times$, $g^a, g^b$
- Output: $g^{ab}$

# CDH and DL

Discrete Logarithm

UC San Diego

- Input: ($p$ prime, $g$ generator), $\mathbb{G} = \langle g \rangle \leq (\mathbb{Z}/p\mathbb{Z})^\times$, $x \in \mathbb{G}$
- Output: $y$ such that $x = g^y$
- Clearly CDH $\leq_p$ DL, other direction not known in general.
- In practice, CDH attacked via reduction to DL

# CDH and DL

## When is DL Easy?

UC San Diego

- DL is $O(\sqrt{q})$ in a subgroup of order $q$
- DL is $O(\sqrt{t})$ if $\text{dlog}_g y = x$ where $x < t$
- The above parallelize well
- Can use Chinese Remainder Theorem to reduce DL in $\mathbb{G}$ to DL in all $Q_i \leq \mathbb{G}$
- So, we want $|\mathbb{G}| = p - 1$ to be non-smooth — $2q$

# Targeting Safe Primes

## Standardization

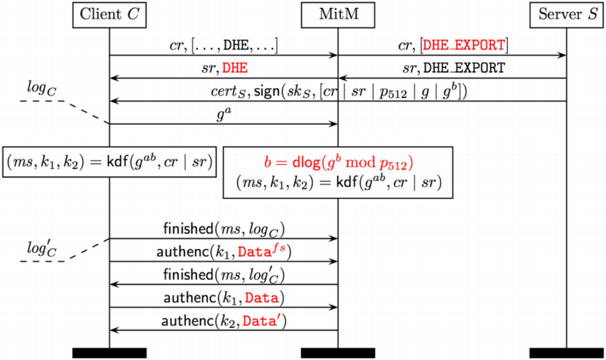
UC San Diego

- 8.4% of Alexa Top 1M HTTPS sites allow for DHE_EXPORT
- 92.3% of them use one of two primes
- Considered safe as most clients never request DHE_EXPORT
- Issue: Breaking DHE_EXPORT during the handshake is a full break
- How fast can DHE_EXPORT be broken?

# Attack Model

## MITM TLS Handshake

UC San Diego




$\text{Data}^{fs}$ is False Start data, and will be discussed later

# NFS

## Online/Offline Distinction

UC San Diego

- Offline (depends on only $p$):

$$\exp\Big((1.923 + o(1))(\log p)^{1/3}(\log\log p)^{2/3}\Big) \quad (1)$$

For $p \approx 2^{512}$, this is $\approx \exp(66.56)$

- Online (depends on $p$ and $x$):
  - Initially: $\approx \exp(66.56)$
  - Then: $\approx \exp(24)$
  - Finally: $\approx \exp(20.5)$
- All of the above parallelizes well

- Start with some Factor Base $F = \{q_1, \ldots, q_k\}$ of primes
- Sieve for relations:

$$\prod_{q_i \in F} q_i^{e_i} \equiv 1 \mod p \tag{2}$$

- Equivalent to:

$$\sum_{q_i \in F} e_i \log_g q_i \equiv 0 \mod (p-1) \tag{3}$$

- For enough relations, can recover $\log_g q_i$ via Linear Algebra over $\mathbb{F}_{p-1}$
- Save these $\log_g q_i$ for use in the online phase

- Have $F = \{q_1, \ldots, q_k\}$ and $\log_g F = \left\{\log_g q_1, \ldots, \log_g q_k\right\}$
- On input $y$, sieve more until we can write:

$$y \equiv \prod_{q_i \in F} q_i^{e_i} \mod p \implies \log_g y \equiv \sum_{q_i \in F} e_i \log_g q_i \mod p - 1 \tag{4}$$

- Recovers $\log_g y$ with much lower cost, so attack has lower amortized cost than asymptotics suggest.
- Requires storing $\log_g F$, in practice this is $\approx 2.5$GB for $|p| \approx 512$
- On a machine with 36 cores and 128 GB ram, compute DL in (median) 70 seconds, and almost always terminates within 140 seconds

# Vulnerabilities Found

## TLS Protections against Downgrading

- TLS can put time limits on the handshake, but:
  - Some non-browser applications (curl and git) have no limits
  - Some web browsers allow the time limit to be extended via TLS warning alerts:
    - Firefix: indefinitely
    - Other browsers: $\approx$ 1 min

# Vulnerabilities Found

Ephemeral Key Caching

- Many TLS servers reuse the $a$ in $(p, g, g^a)$:
  - 17% reuse $g^a$ at least once over 20 handshakes
  - 15% use one value
- Reuse is less common (0.1%) for DHE_EXPORT, attack easily extends to other DHE (it just costs more)

# Vulnerabilities Found

## TLS False Start

UC San Diego

- Reduces connection latency via sending early application data without waiting for the Finished message to arrive
- Often contains passwords and cookies (and still a break)

# Applicability of the Attack

## Cost Estimation

| Size | Online | Offline |
| --- | --- | --- |
| 512 | 10.2 | 10 core-minutes |
| 768 | 36,500 | 48 |
| 1024 | 45,000,000 | 720 |

Units are core-years unless mentioned otherwise. All tasks parallelize well.

## Applicability of the Attack

Is the NSA attacking 1024-bit DHE?

UC San Diego

- The authors estimate that even the most powerful supercomputer in the US (300,000 cores) would take 117 years to finish the Linear Algebra stage
  - This cost $94M in 2012 to build, suggesting $11B for hardware
  - Optimizing CPUs $\mapsto$ ASICs is estimated to increase efficiency 80x
  - Estimated cost to break 1024-bit DHE: a few hundred million
- The NSA gets $10.5B per year in 2012
- Published documents by Der Spiegal indicate NSA is passively decrypting VPN connections at scale
  - Could be solely through malware
  - Could be through larger break, which is consistent with a 1024-bit DHE break (the majority of clients use a single group)
    - Requirements of using LogJam (recovering nonces, cookies, and $g^a$ and $g^b$) match requirements of published NSA techniques
    - Moreover, if a pre-shared key (PSK) is used, both LogJam and the NSA method require the PSK.

# Applicability of the Attack

## Ramifications of attacking 1024-bit DHE

Note: This attack model has a passive attacker who has precomputed a single 1024-bit group

They could attack:

- $\approx$ 64% of VPN connections
- $\approx$ 25% of publicly-accessible SSH servers
- $\approx$ 18% of the top 1M sites

# Circumventing the Attack

- Switch to Elliptic Curve DHE:
  - No known sub-exponential algorithms (like NFS) in general case
  - More efficient
  - Con: NSA influence (a la dual_ec_drbg)
- Increase minimum key strengths
- Don't use fixed safe primes

# The end!