Imperfect Forward Secrecy

David Adrian
@davidcadrian
Who am I?

I’m **David Adrian**, graduate student at the University of Michigan

An Academic.

“Halfademic”
What do I do?
What is this?

Logjam

Weak Diffie-Hellman

Internet-scanning?

Mail security?

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

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

* INRIA Paris-Rocquencourt † INRIA Nancy-Grand Est, CNRS and Université de Lorraine ‡ Microsoft Research † University of Pennsylvania § Johns Hopkins † University of Michigan

For additional materials and contact information, visit WeakDH.org.

ABSTRACT

We investigate the security of Diffie-Hellman key exchange as used in popular Internet protocols and find it to be less secure than widely believed. First, we present a novel flaw in TLS logs in that group, amortizing the cost over all targets that share this parameter. The algorithm can be tuned to reduce individual log cost even further. Although this fact is well known among mathematical cryptographers, it seems to have
Diffie-Hellman Key Exchange
Diffie-Hellman Key Exchange

First published key-exchange algorithm

Two parties agree on a **shared secret key** over an unsecured channel
Public parameters
- $p$, a large prime
- $g$, a generator for a group modulo $p$

$$g^a \mod p$$

$$g^b \mod p$$

$$g^{ba} \mod p = g^{ab} \mod p$$
Shortcomings

Unauthenticated  Fix by signing with a long-term key (certificate)

How to pick $g$ and $p$?  Standardize in protocol, or allow server to choose
Perfect Forward Secrecy

When a new Diffie-Hellman key exchange is completed at the start of every connection, you gain perfect forward secrecy.

Breaking one-connection or the long-term key of a server, does not allow an adversary to decrypt past connections.
“Sites that use perfect forward secrecy can provide better security to users in cases where the encrypted data is being monitored and recorded by a third party."

“With Perfect Forward Secrecy, anyone possessing the private key and a wiretap of Internet activity can decrypt nothing.”
Breaking Diffie-Hellman

Requires finding the solution to a “hard” mathematical problem called \textit{discrete log}

Given \( g^x \equiv y \mod p \), compute \( x \)
Breaking Diffie-Hellman

Requires finding the solution to a “hard” mathematical problem called **discrete log**

Given \( g^x \equiv y \mod p \), compute \( x \)

Want

Have

Conceptually easy ✓, computationally hard ✗
Breaking Diffie-Hellman

State-of-the-art is the **number-field sieve** algorithm.
Number-Field Sieve

Precomputation depends solely on $p$!
Logjam Attack on TLS
The identity of this website has been verified by Google Internet Authority G2. Valid Certificate Transparency information was supplied by the server.

Certificate Information

Your connection to www.google.com is encrypted using a modern cipher suite.
The connection uses TLS 1.2.

The connection is encrypted and authenticated using AES_128_GCM and uses ECDHE_RSA as the key exchange mechanism.

What do these mean?
**Client Hello**: client random, ciphers (...DHE...)

**Server Hello**: server random, chosen cipher

**Certificate**: certificate chain (public key)

**Server Kex Exchange**: $p$, $g$, $g^a$, $\text{Sign}_{\text{CertKey}}(p, g, g^a)$

**Client Key Exchange**: $g^b$

**Client Finished**: $\text{Sign}_{K_{ms}}(\text{Hash}(m1 \mid m2 \mid ...))$

**Server Finished**: $\text{Sign}_{K_{ms}}(\text{Hash}(m1 \mid m2 \mid ...))$

$K_{ms}$: $\text{KDF}(g^{ab}, \text{client random, server random})$
Export Ciphers in TLS

Remnant of the 90s “crypto wars”

It used to be illegal to export “strong crypto” outside of the United States, law was overturned in *Bernstein vs. United States of America*

TLS was designed before the law was overturned

Included weak (short-key) “export ciphers” for use outside of the United States, e.g. **DHE_EXPORT**

**DHE_EXPORT** uses 512-bit primes!
Support for Export Ciphers

8.5% of the Alexa Top 1M support **DHE_EXPORT**

<table>
<thead>
<tr>
<th>Prime</th>
<th>Popularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache mod_ssl</td>
<td>82%</td>
</tr>
<tr>
<td>nginx</td>
<td>10%</td>
</tr>
<tr>
<td>Other (463 primes)</td>
<td>8%</td>
</tr>
</tbody>
</table>
Breaking 512-bit

We did the precomputation for the two most popular 512-bit primes.

<table>
<thead>
<tr>
<th>Method</th>
<th>polysel</th>
<th>sieving</th>
<th>linalg</th>
<th>descent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores</td>
<td>2000-3000</td>
<td>288</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>DH-512</td>
<td>3 hours</td>
<td>15 hours</td>
<td>120 hours</td>
<td>70 seconds</td>
</tr>
</tbody>
</table>

CDF of keys

Seconds
Mitigations

Browsers
- No longer support 512-bit
- Will be sunsetting 768-bit and 1024-bit

Server Operators
- Disable DHE_EXPORT
- Move to 2048-bit or elliptic curve variant
What about 1024-bit?
## Cost of NFS

Rough estimations based on asymptotic complexity

<table>
<thead>
<tr>
<th></th>
<th>Sieving core-years</th>
<th>Linear Algebra core-years</th>
<th>Descent core-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA-512</td>
<td>0.5</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>DH-512</td>
<td>2.5</td>
<td>7.7</td>
<td>10 mins</td>
</tr>
<tr>
<td>RSA-768</td>
<td>800</td>
<td>100</td>
<td>2 days</td>
</tr>
<tr>
<td>DH-768</td>
<td>8,000</td>
<td>28,500</td>
<td></td>
</tr>
<tr>
<td>RSA-1024</td>
<td>1,000,000</td>
<td>120,000</td>
<td></td>
</tr>
<tr>
<td>DH-1024</td>
<td>10,000,000</td>
<td>35,000,000</td>
<td>30 days</td>
</tr>
</tbody>
</table>
Custom Hardware

If you were actually attempting this, would use custom hardware.

Prior work suggests \(~80\times\) speedup from equivalent cost in custom hardware.
<table>
<thead>
<tr>
<th>Prime Length</th>
<th>Broken By...</th>
<th>Precomputation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>512-bit</td>
<td>Academics</td>
<td>1 week</td>
</tr>
<tr>
<td>768-bit</td>
<td>Academics</td>
<td>1 month</td>
</tr>
<tr>
<td>1024-bit</td>
<td>Nation State Large Organization</td>
<td>1 year ~$100-300M</td>
</tr>
</tbody>
</table>
Impact of a 1024-bit break

Precomputing on one 1024-bit prime (Oakley Group 2) would allow passively decrypting connections with:

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

The second most common prime (Apache):

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

<table>
<thead>
<tr>
<th></th>
<th>Top Prime</th>
<th>Top 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS Top 1M</td>
<td>205K (37.1%)</td>
<td>309K (56.1%)</td>
</tr>
<tr>
<td>HTTPS All</td>
<td>1.8M (12.8%)</td>
<td>3.4M (23.8%)</td>
</tr>
<tr>
<td>SSH</td>
<td>3.6M (25.7%)</td>
<td>3.6M (25.7%)</td>
</tr>
<tr>
<td>IKE (VPN)</td>
<td>1.7M (66.1%)</td>
<td>1.7M (66.1%)</td>
</tr>
</tbody>
</table>
Is NSA Breaking 1024-bit?
4. Communicate Results

Can we decrypt the VPN traffic?

- If the answer is “No” then explain how to turn it into a “YES!”

- If the answer is “YES!” then...
Happy Dance!!
Turn that Frown Upside Down! From “No” to “YES!”

- Depends on why we couldn’t decrypt it
- Find Pre-Shared Key
- Locate complete paired collect
- Locate both IKE and ESP traffic
- Have collection sites do surveys for the IP’s
- Find better quality collect with rich metadata
Where did all this data come from?
ZMap

2013  A 1200x performance improvement over Nmap for an Internet-wide single port TCP scan

2014  Scan the Internet in under 5 minutes.

2015  Popular in industry and academia, used by over 104 academic studies
## ZMap Vision

<table>
<thead>
<tr>
<th>Goals</th>
<th>Reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable new and exciting research</td>
<td>Not all researchers can run ZMap</td>
</tr>
<tr>
<td>Decrease the barriers to entry for Internet-wide surveys</td>
<td>Negotiate with network administrators for bandwidth and address space</td>
</tr>
<tr>
<td>Anyone can scan the entire Internet using a single host</td>
<td>Maintain an opt-out list and respond to complaints</td>
</tr>
</tbody>
</table>
Search engine that allows researchers to ask questions about the devices and networks that compose the Internet.
Example

What hosts still support DHE_EXPORT?
133.24.255.156 (tito-nat1.chem.yz.yamagata-u.ac.jp)
- SINET-AS - Research Organization of Infor... (2907)  🇯🇵 Japan
- 443/https, 22/ssh, 53/dns
- ftp-bigip.yz.yamagata-u.ac.jp
- dhe-export rsa-export ssh https

202.133.226.93
- ABOVE-AS-AP - AboveNet Communications Taiwan (17408)  🇹🇼 Taiwan
- 80/http, 443/https, 53/dns
- www.nusoft.com.tw
- dhe-export rsa-export http https

216.206.86.64 (s64.wifieval.adtran.com)
- ADTRAN - Adtran, Inc. (25739)  🇺🇸 United States
- 443/https, 53/dns
- *.wifieval.adtran.com, wifieval.adtran.com
- dhe-export rsa-export https

- ULSK-AS - JSC ER-Telecom Holding (39028)  🇷🇺 Ulyanovsk, Ulyanovsk Oblast, Russia
- 80/http, 110/pop3, 21/ftp, 143/imap, 53/dns, 443/https, 22/ssh
- 403 Forbidden  🇷🇺 panas.mst, www.panas.mst
- ftp http pop3 ssh https dhe-export rsa-export imap

144.133.195.178
What hosts still support DHE_EXPORT?
This tool allows you to generate a report on the breakdown of a value present on the ipv4s returned by your query. For example, to generate a report on the cipher suites chosen by HTTPS servers in the United States, you could query for `location.country_code: US AND protocols:443/https` and then generate a report on the breakdown of the field `443.https.tls.cipher_suite.name`. A list of reportable fields is available here.

Many fields have both parsed and raw values available (e.g., `88.http.get.headers.server` and `88.http.get.headers.server.raw`). In these cases, the raw value will represent the exact string (e.g., `Apache/2.2.22 (Debian)`) and the parsed version will bucket on individual terms (e.g., `Apache` and `Debian`). Incidentally, in this case, you likely want to aggregate on a parsed out version of the web server, `88.http.get.metadata.description.raw`.

**Host Report**

```
443.https.dhe.export.support
```

```
15,331,054
```

```
186,079
```
Contributing

Are you extending ZMap, ZGrab, or another scanner with a new protocol?

Do you have annotations to add to our framework?

We’ll work with researchers to add new scan modules to Censys

https://github.com/zmap/zmap

https://github.com/zmap/zgrab

https://github.com/zmap/ztag
Finishing Up
Diffie-Hellman Recommendations

Transition to elliptic curve cryptography (ECC)

If ECC isn’t an option, use 2048-bit primes or larger

If 2048-bit isn’t an option, use a fresh 1024-bit prime

All major desktop browsers now reject 512-bit groups, and are sunsetting 768-bit and 1024-bit

Turn export ciphers off!
Censys

Censys strives to be research enabling more research

Contribute back scanners and annotations — we do the heavy lifting

Bring measurement-driven security to a wider audience
Questions?

David Adrian
@davidcadrian

https://weakdh.org
https://censys.io
https://zmap.io