Measuring small subgroup attacks against Diffie-Hellman

Luke Valenta*, David Adrian†, Antonio Sanso‡, Shaanan Cohney*, Joshua Fried*, Marcella Hastings*, J. Alex Halderman†, Nadia Heninger*

*University of Pennsylvania
†University of Michigan
‡Adobe

February 28, 2017
This work

- Revisit decades-old small subgroup attacks in Diffie-Hellman
- Looked at hosts and implementations in the wild
- Punch line: Nobody implements the countermeasures!
- Emerged from Logjam [ABDGGHHSTVVWZZ 2015]
Textbook (Finite-Field) Diffie-Hellman Key Exchange
[Diffie Hellman 1976]

\[ p \text{ a prime } (\text{so } \mathbb{F}_p^* \text{ is a cyclic group}) \]

\[ g < p \text{ group generator (often 2 or 5)} \]

Images from XKCD
Textbook (Finite-Field) Diffie-Hellman Key Exchange

[Diffie Hellman 1976]

\[ p \text{ a prime (so } \mathbb{F}_p^* \text{ is a cyclic group)} \]

\[ g < p \text{ group generator (often 2 or 5)} \]

\[ g^a \mod p \]

\[ g^b \mod p \]

\[ \text{Enc}_{g^{ab}}(\text{data}) \]

\[ g^{ab} \mod p \]
Textbook (Finite-Field) Diffie-Hellman Key Exchange
[Diffie Hellman 1976]

$p$ a prime (so $\mathbb{F}_p^*$ is a cyclic group)

$g < p$ group generator (often 2 or 5)

\[
g^a \mod p \\
g^b \mod p \\
Enc_{g^{ab}}(data) \\
g^{ab} \mod p \\
g^{ab} \mod p
\]

Images from XKCD
NH: “There are dragons swimming under the placid surface of this beautiful mathematical lake.”
Background: groups, subgroups, and generators

Cyclic group

$$\text{Order} = \#\text{elements in group}$$
Background: groups, subgroups, and generators

Cyclic group

Order = \#elements in group

generator
Background: groups, subgroups, and generators

Cyclic group

Order = \#elements in group

generator
Background: groups, subgroups, and generators

Cyclic group

Order = \#elements in group

generator
Background: groups, subgroups, and generators

Cyclic group

Order = \#elements in group

generator
Background: groups, subgroups, and generators

Cyclic group

Order = \#elements in group

generator
Background: groups, subgroups, and generators

Subgroup

Order = \#elements in subgroup

generator
Background: groups, subgroups, and generators

Subgroup

Order = \#elements in subgroup

generator
Background: groups, subgroups, and generators

Subgroup

Order = \#elements in subgroup

generator
Background: groups, subgroups, and generators

Subgroup

Order = \#elements in subgroup

generator
Background: groups, subgroups, and generators

Small subgroup

Order = \# elements in subgroup

generator
Background: groups, subgroups, and generators

Small subgroup

Order = \#elements in subgroup

generator
Background: groups, subgroups, and generators

Small subgroup

Order = #elements in subgroup

generator
Background: groups, subgroups, and generators

Small subgroup

Order = #elements in subgroup

generator
Existence of small subgroups $\rightarrow$ small subgroup attacks.

\[ g \text{ generates correct subgroup of order } q \]
\[ g_3 \text{ generates subgroup of order 3} \]

[Lim Lee 1997]

\[ g^b, \operatorname{Enc}_{g_3^b}(data) \]

compute $b \mod 3$
Existence of small subgroups $\rightarrow$ small subgroup attacks.

- $g$ generates correct subgroup of order $q$
- $g_3$ generates subgroup of order 3

[Lim Lee 1997]

Repeat for many small subgroups $\Rightarrow$ find $b$ using Chinese Remainder Theorem
Small subgroup attacks

Made much worse with...

- Many small subgroups (i.e., p-1 has many small factors)
- Short secret exponents (common optimization)
- Reused Diffie-Hellman values (common optimization)
The countermeasures against these attacks are well known, and built into every DH standard:

- Use a “safe” prime $p = 2q + 1$, where $q$ is prime
  1. Verify $2 \leq y \leq p - 2$ (otherwise, may leak 1 bit)
- Use a subgroup of large prime order $q \mod p$
  1. Verify $2 \leq y \leq p - 2$
  2. Verify $1 = y^q \mod p$
Inspiration for our work

The attacks and defenses are known. Why is this work interesting?
Inspiration for our work

The attacks and defenses are known. Why is this work interesting?

“The Internet is vast, and filled with bugs.”

—Adam Langley, Crypto 2013
Inspiration for our work

The attacks and defenses are known. Why is this work interesting?

“The Internet is vast, and filled with bugs.”

—Adam Langley, Crypto 2013

Theorem (Murphy’s law)

*Anything that can go wrong, will go wrong.*

Corollary

*If it is possible for an implementation to have made a mistake, someone has.*
Standards mandate smaller subgroups
Leaves room for implementation mistakes

NIST SP800-56a: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography

Table 1: FFC parameter-size sets

<table>
<thead>
<tr>
<th>FFC parameter-size set name</th>
<th>FA</th>
<th>FB</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum security strength supported (in bits)</td>
<td>80</td>
<td>112</td>
<td>112</td>
</tr>
<tr>
<td>Bit length of field size $p$ (i.e., $\lceil \log_2 p \rceil$)</td>
<td>1024</td>
<td>2048</td>
<td>$2048^1$</td>
</tr>
<tr>
<td>Bit length of subgroup order $q$ (i.e., $\lceil \log_2 q \rceil$)</td>
<td>160</td>
<td>224</td>
<td>256</td>
</tr>
</tbody>
</table>

- No extra benefit from using small subgroups when already using short exponents
- DSA needs small subgroups, but not DH
Fast internet scanning lets us study behavior of publicly accessible hosts.

Widely deployed RFC5114 groups follow NIST recommendations*:

<table>
<thead>
<tr>
<th>Group</th>
<th>p (bits)</th>
<th>q (bits)</th>
<th>HTTPS</th>
<th>SMTP</th>
<th>IKEv1</th>
<th>IKEv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 22</td>
<td>1024</td>
<td>160</td>
<td>3%</td>
<td>≈ 0%</td>
<td>17%</td>
<td>13%</td>
</tr>
<tr>
<td>Group 23</td>
<td>2048</td>
<td>224</td>
<td>≈ 0%</td>
<td>33%</td>
<td>17%</td>
<td>13%</td>
</tr>
<tr>
<td>Group 24</td>
<td>2048</td>
<td>256</td>
<td>≈ 0%</td>
<td>≈ 0%</td>
<td>18%</td>
<td>14%</td>
</tr>
<tr>
<td>Total</td>
<td>—</td>
<td>—</td>
<td>40.6M</td>
<td>3.4M</td>
<td>1.9M</td>
<td>1.3M</td>
</tr>
</tbody>
</table>

Group 23: Can recover 201 bits of exponent in $\approx 2^{42}$ work

*: Scans from November 2016
Hosts don’t validate group order.

<table>
<thead>
<tr>
<th>DHE</th>
<th>Hosts</th>
<th>Non-Safe Primes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>11M</td>
<td>14%</td>
</tr>
<tr>
<td>IKEv1</td>
<td>2.6M</td>
<td>13%</td>
</tr>
<tr>
<td>IKEv2</td>
<td>1.3M</td>
<td>14%</td>
</tr>
<tr>
<td>SSH</td>
<td>11M</td>
<td>≈ 0%</td>
</tr>
</tbody>
</table>
Hosts don’t validate group order.

<table>
<thead>
<tr>
<th>DHE</th>
<th>Hosts</th>
<th>Non-Safe Primes</th>
<th>Hosts accepting...</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>11M</td>
<td>14%</td>
<td>0.6%</td>
</tr>
<tr>
<td>IKEv1</td>
<td>2.6M</td>
<td>13%</td>
<td>*</td>
</tr>
<tr>
<td>IKEv2</td>
<td>1.3M</td>
<td>14%</td>
<td>*</td>
</tr>
<tr>
<td>SSH</td>
<td>11M</td>
<td>≈ 0%</td>
<td>3%</td>
</tr>
</tbody>
</table>

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.
Hosts don’t validate group order.

<table>
<thead>
<tr>
<th>DHE Hosts</th>
<th>Non-Safe Primes</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>11M</td>
<td>14%</td>
<td>0.6%</td>
</tr>
<tr>
<td>IKEv1</td>
<td>2.6M</td>
<td>13%</td>
<td>*</td>
</tr>
<tr>
<td>IKEv2</td>
<td>1.3M</td>
<td>14%</td>
<td>*</td>
</tr>
<tr>
<td>SSH</td>
<td>11M</td>
<td>≈ 0%</td>
<td>3%</td>
</tr>
</tbody>
</table>

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.
Hosts don’t validate group order.

<table>
<thead>
<tr>
<th>DHE Hosts</th>
<th>Non-Safe Primes</th>
<th>Hosts accepting...</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>11M</td>
<td>14% 0.6% 3% 5%</td>
</tr>
<tr>
<td>IKEv1</td>
<td>2.6M</td>
<td>13% * 28% 27%</td>
</tr>
<tr>
<td>IKEv2</td>
<td>1.3M</td>
<td>14% * 0% 0%</td>
</tr>
<tr>
<td>SSH</td>
<td>11M</td>
<td>≈ 0% 3% 25% 33%</td>
</tr>
</tbody>
</table>

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.
Hosts don’t validate group order.

<table>
<thead>
<tr>
<th>DHE Hosts</th>
<th>Non-Safe Primes</th>
<th>0</th>
<th>1</th>
<th>p-1</th>
<th>$g_3/g_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS</td>
<td>11M</td>
<td>14%</td>
<td>0.6%</td>
<td>3%</td>
<td>5%</td>
</tr>
<tr>
<td>IKEv1</td>
<td>2.6M</td>
<td>13%</td>
<td>*</td>
<td>28%</td>
<td>27%</td>
</tr>
<tr>
<td>IKEv2</td>
<td>1.3M</td>
<td>14%</td>
<td>*</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>SSH</td>
<td>11M</td>
<td>$\approx 0%$</td>
<td>3%</td>
<td>25%</td>
<td>33%</td>
</tr>
</tbody>
</table>

*: Did not scan: 0 causes unpatched Libre/Openswan to restart IKE daemon.
Libraries don’t validate group order.
Similar findings to [DCE 2017 (up next!)]

<table>
<thead>
<tr>
<th>Library</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozilla NSS</td>
<td>$g \leq 2$</td>
</tr>
<tr>
<td>OpenJDK</td>
<td>$g \leq 2$</td>
</tr>
<tr>
<td>OpenSSL 1.0.2</td>
<td>None*</td>
</tr>
<tr>
<td>BouncyCastle</td>
<td>$g \leq 2$</td>
</tr>
<tr>
<td>Cryptlib</td>
<td>$g \leq 2$</td>
</tr>
<tr>
<td>libTomCrypt</td>
<td>None</td>
</tr>
<tr>
<td>CryptoPP</td>
<td>None</td>
</tr>
<tr>
<td>Botan</td>
<td>None</td>
</tr>
<tr>
<td>GnuTLS</td>
<td>$g \leq 2$</td>
</tr>
</tbody>
</table>

- “The server obtains the DH parameters via a PKCS#3 file which does not contain any subgroup information. This file format is the defacto standard across all crypto libraries.”

- OpenSSL vulnerable to full Lim-Lee key recovery attack for RFC 5114 primes

- Amazon Load Balancer vulnerable to partial key recovery attack

*: before CVE-2016-0701 in Jan ’16
Misconceptions

**Academics**

“There are many good reasons for using smaller subgroups, including **efficiency** and the fact that this setting matches the **theoretical security analyses of cryptosystems**.”

**Implementors**

“safe primes (...) have quite some undesirable properties. They don’t have a subgroup with size of the selected security parameter and that **requires them to use very large keys**.”

**Fact:** Short exponents with safe primes and with small subgroups are both well-studied
disconnects

academics

“(…) it is only necessary to validate cryptographic parameters properly - **but this is very well-known.**”

implementors

“I bet there are TLS clients (and other DH users) out there that use those values, and we would break them (…) **functionality trumps security every day, and twice on Tuesdays.**”

Countermeasures may be known, but are not always implemented
Takeaways

- Standards writers:
  - Software developers have different priorities
  - The fewer checks required, the better! (Murphy’s Law)

- Software developers:
  - Take care when it comes to cryptographic validation
  - Project Wycheproof: test crypto libraries against known attacks ([GitHub](https://github.com/google/wycheproof))

- Sysadmins:
  - Test your servers with our tools! ([GitHub](https://github.com/eniac/crypscan))

Questions?
Takeaways

▶ Standards writers:
  ▶ Software developers have different priorities
  ▶ The fewer checks required, the better! (Murphy’s Law)

▶ Software developers:
  ▶ Take care when it comes to cryptographic validation
  ▶ Project Wycheproof: test crypto libraries against known attacks (https://github.com/google/wycheproof)
Takeaways

▶ Standards writers:
  ▶ Software developers have different priorities
  ▶ The fewer checks required, the better! (Murphy’s Law)

▶ Software developers:
  ▶ Take care when it comes to cryptographic validation
  ▶ Project Wycheproof: test crypto libraries against known attacks (https://github.com/google/wycheproof)

▶ Sysadmins:
  ▶ Test your servers with our tools!
    (https://github.com/eniac/crypscan)
Takeaways

▶ Standards writers:
  ▶ Software developers have different priorities
  ▶ The fewer checks required, the better! (Murphy’s Law)

▶ Software developers:
  ▶ Take care when it comes to cryptographic validation
  ▶ Project Wycheproof: test crypto libraries against known attacks (https://github.com/google/wycheproof)

▶ Sysadmins:
  ▶ Test your servers with our tools! (https://github.com/eniac/crypscan)

Questions?
References

*Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice*

*Indiscreet Logs: Persistent Diffie-Hellman Backdoors in TLS*
Kristen Dorey, Nicholas Chang-Fong, Aleksander Essex. *NDSS 2017*
Measuring small subgroup attacks against Diffie-Hellman

Luke Valenta*, David Adrian†, Antonio Sanso‡, Shaanan Cohney*, Joshua Fried*, Marcella Hastings*, J. Alex Halderman†, Nadia Heninger*

*University of Pennsylvania
†University of Michigan
‡Adobe

February 28, 2017