Review of

Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice

David Adrian et. al.

What is the problem?

Diffie-Hellman key exchange which is widely used in Internet Protocol is less secure than widely believed. Due to a protocol flaw in TLS, a man-in-the-middle attacker can downgrade a connection to export-grade cryptography, can also be construed as a backwards compatibility attack.

Summary

Exploiting a novel flaw in TLS, Logjam is able to actively downgrade connections to export-grade Diffie-Hellman and using NSF discrete log algorithm and extensive precomputations, Logjam is able to compute the discrete logs for the particular group and is able to take control of the communication between the victim clients and servers.

Key Insights

- Explanations about the different delays they needed to create to keep the connection alive while discrete log computation is going on in the background gave new insights into how modern browsers handle TLS connections.
- Extensive research about the compute power needed to compute the 784-bit and 1024-bit groups gave insights about state-of-the-art machines of the time.

Strengths

- The paper does a good job in explaining the steps they had to take and the compute power and time it takes for each of the steps. They also explained the complexity equation well. They have done extensive studies about their measurements and have all the data to back them up.
- Logjam is versatile and the paper identifies such instances where even if the DHE_EXPORT is not supported there are other flaws that they can exploit.

Weaknesses

- The only weakness is that there could have been more to the section of disclosure and response because a vulnerability of this magnitude should have elicited a stronger response.

Summary of Key Results

- Paper recommends the use of ECDHE and to be suspect about the NIST parameters due to the involvement of NSA in their design, to increase minimum key strengths and to generate fresh groups if maintaining compatibility with DHE is necessary.
- To bridge the apparent gap between cryptographers (theoreticians), and practical system designers will help alleviate a lot of the problems that come with flaws in implementation.

Open Questions

- What is the current support for these weak protocols, has the disclosure done enough to ward off servers and clients from using and supporting these protocols?