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-Béguelink, Paul Zimmermann

Facultad de Ciencias Exactas, Ingeniería y Agrimensura,
Universidad Nacional de Rosario.

Ines Cipullo - Diciembre 2024

Protocolo Diffie-Hellman

- Alice y Bob eligen un número primo p y un generador g
- Alice $\rightarrow$ Bob: $A = g^a \bmod p$
- Bob $\rightarrow$ Alice: $B = g^b \bmod p$
- Alice y Bob calculan un **shared secret** a partir de A y B: $g^{ab} \bmod p$
- Este **shared secret** se utilizará como clave para el resto de la comunicación

Cómo vulnerar Diffie-Hellman?

Un atacante que observa el intercambio tiene acceso a p, g, A y B. Para poder calcular la clave, su mejor opción es averiguar uno de los componentes privados, a o b. Eso lo logra calculando el logaritmo discreto del valor público correspondiente: A o B.

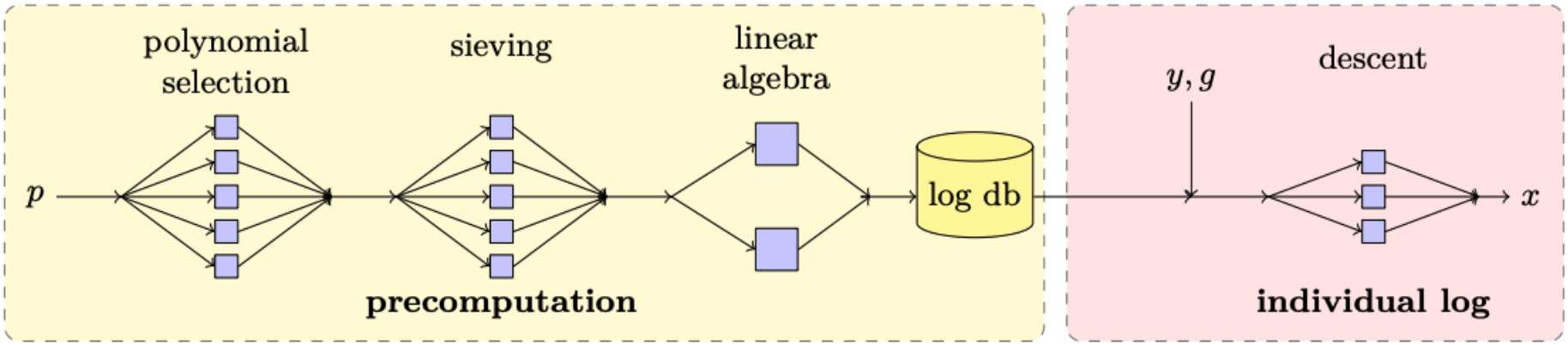


Figure 1: **The number field sieve algorithm for discrete log** consists of a precomputation stage that depends only on the prime p and a descent stage that computes individual logs. With sufficient precomputation, an attacker can quickly break any Diffie-Hellman instances that use a particular p.

TLS y Diffie-Hellman

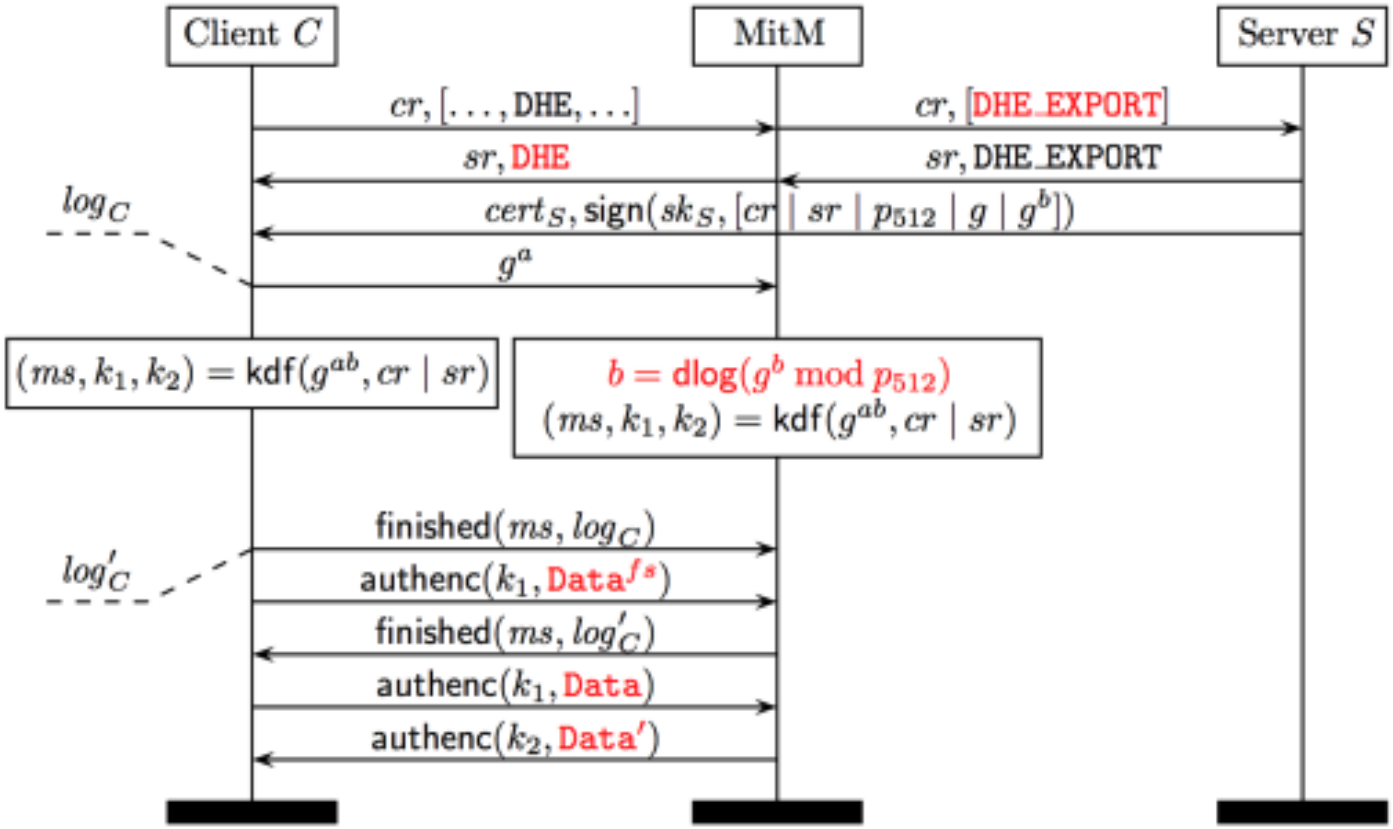
TLS es un protocolo criptográfico para comunicaciones entre aplicaciones, y su uso más usual es en HTTPS.
TLS Handshake:

- El cliente envía al servidor un mensaje *ClientHello* que incluye la lista de algoritmos criptográficos que soporta y un fresco *cr*
- El servidor elige un algoritmo y se lo comunica en un mensaje *ServerHello*, también incluyendo un fresco *sr*

TLS soporta multiples variantes de Diffie-Hellman:

- Ephemeral Diffie-Hellman (DHE): es el definido originalmente y soporta números de cualquier cantidad de bits.
- Export-grade Diffie-Hellman (DHE_Export): soporta números de un máximo de 512 bits, igual a DHE en el resto. La mayoría de clientes ya no lo aceptan.

Ataque LogJam



Computaciones de logaritmo discreto

- Se aplicó el algoritmo a 3 primos de 512 bits
- Las precomputaciones tardaron 7 días por cada número primo
- Seleccion polinomial y sieving (o tamizado) se pueden paralelizar, el algebra lineal no tanto. Se decidio hacer más sieving, lo que permite que el paso de algerba lineal sea más corto, y que el paso de descenso sea más rápido.
- Se logra computar el último paso en un promedio de 70 segundos.

Reemplazar al servidor en la comunicación

Para poder completar el ataque, se debe poder computar el secreto en tiempo real, pero el algoritmo que se obtuvo tarda en promedio 70 segundos. Tecnicas para que el cliente espere nuestra respuesta:

- Si el usuario usa un **cliente de consola**, como git o curl, suelen tener un tiempo de espera mayor o no tener tiemout.
- **TLS warning alerts**: los navegadores ignoran estos mensajes pero se reinicia el contador del timeout, por lo que el atacante los puede utilizar para demorar el timeout.
- **Caching de claves efímeras**: algunos servidores no utilizan un valor fresco de b para cada nueva conexión, sino que lo reutilizan para evitar tener que recalcular g^b. Si se obtiene la clave, se puede atacar futuras conexiones en tiempo real.
- **TLS false start**: algunos clientes envían datos sensibles antes de concluir el handshake para reducir la latencia de la comunicación, de esta forma el atacante luego puede desencriptarlo sin restricciones de tiempo.

Amenazas estatales

Como vimos, para poder llevar a cabo el ataque es necesario degradar la conexión para utilizar primos de 512 bits, ya que es lo que nos permite el poder de cómputo, y esto depende de la falla de TLS.
Qué pasaria si esto no fuese necesario?

	Sieving			Linear Algebra		Descent	
	I	$\log_2 B$	core-years	rows	core-years	core-time	
RSA-512	14	29	0.5	4.3M	0.33		Timings with default CADO-NFS parameters.
DH-512	15	27	2.5	2.1M	7.7	10 mins	For the computations in this paper; may be suboptimal.
RSA-768	16	37	800	250M	100		Est. based on [29] with less sieving.
DH-768	17	35	8,000	150M	28,500	2 days	Est. based on [8, 29] and our own experiments.
RSA-1024	18	42	1,000,000	8.7B	120,000		Est. based on complexity formula.
DH-1024	19	40	10,000,000	5.2B	35,000,000	30 days	Est. based on complexity formula and our experiments.

Table 2: **Estimating costs for factoring and discrete log**. For sieving, we give two important parameters: the number of bits of the smoothness bound B and the sieving region parameter I. For linear algebra, all costs for DH are for safe primes; for DSA primes with q of 160 bits, this should be divided by 6.4 for 1024 bits, 4.8 for 768 bits, and 3.2 for 512 bits.

La NSA ya rompió DH 1024 bits?

Documentos clasificados que fueron publicados indicarían que sí.

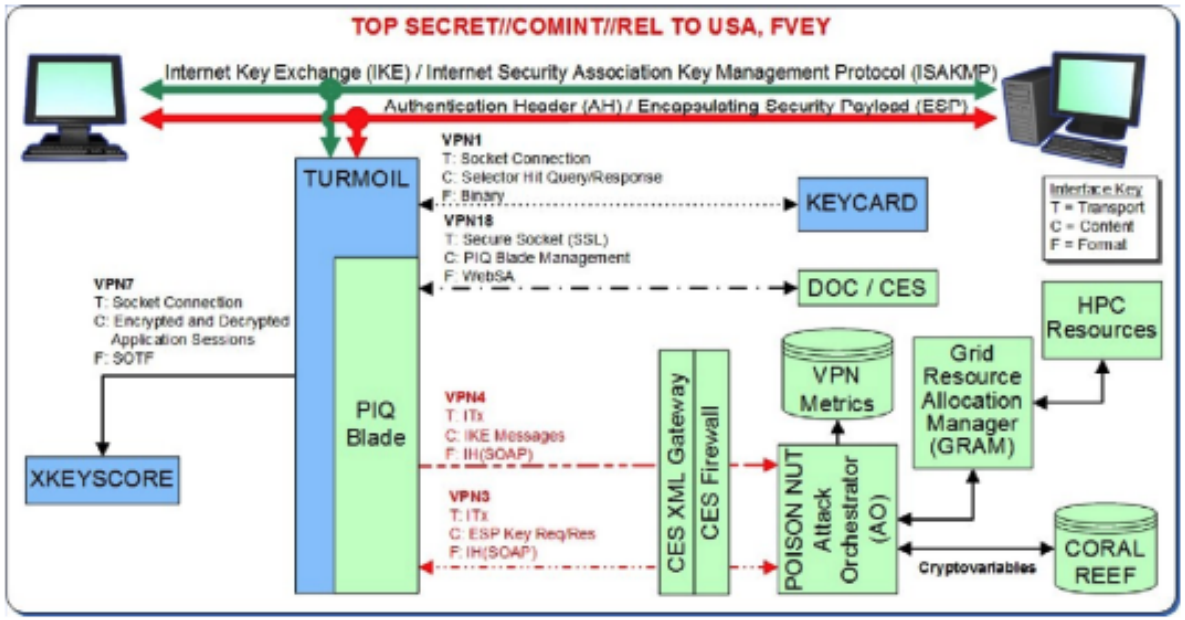


Figure 4: **NSA's VPN decryption infrastructure.** This classified illustration published by Der Spiegel [67] shows captured IKE handshake messages being passed to a high-performance computing system, which returns the symmetric keys for ESP session traffic. The details of this attack are consistent with an efficient break for 1024-bit Diffie-Hellman.

Qué consecuencias tiene esto?

Asumiendo los recursos para poder precomputar una cantidad pequeña de grupos de 1024 bits, se analizó el uso de Diffie-Hellman en los protocolos más populares: IKE, HTTPS, SSH.

IKE:

- 31,8% de IKEv1 soportan Oakley 1 (2,6% lo prefiere)
- 86,1% de IKEv1 soportan Oakley 2 (66,1% lo prefiere)
- 19,7% de IKEv2 soportan Oakley 1 (5,8% lo prefiere)
- 91% de IKEv2 soportan Oakley 2 (63,9% lo prefiere)

SSH:

- 98.9% soporta Oakley 2 (21% lo prefiere)

HTTPS (sample: Alexa Top 1M):

- 24% de la conexiones utilizaron DH
- 84% usa grupos de 1024 bits o menos (94% usa 1 de los mismos 5 grupos)

Qué consecuencias tiene esto? (Cont.

TLS también se puede utilizar para asegurar comunicaciones entre servidores de email. Se evaluó un sample de todas las IPs publicas para IMAPS, POP3S y SMTP.

- Entre un 8% y un 15% soportan DHE_EXPORT
- Alrededor del 75% soporta DHE

Si se comprometieran los 10 primos de 1024 bits más usados por cada protocolo, se comprometerian las conexiones de más de 2M de servidores de mail.

Alternativas de defensa

- Utilizar Diffie-Hellman elíptico:
  - Actualmente no se conocen ataques
  - Es menos costoso computacionalmente para los usuarios
- Incrementar el largo mínimo de las claves:
  - 2048 bits como estándar
  - No se debería aceptar menos de 1024 bits
- Evitar utilizar los mismos grupos de primos:
  - Evita que afecten las precomputaciones existentes
  - No amortiza costos del atacante
- No permitir debilitar deliberadamente protocolos criptográficos:
  - DHE_EXPORT se debería haber deprecado hace bastante