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

`weakdh.org`

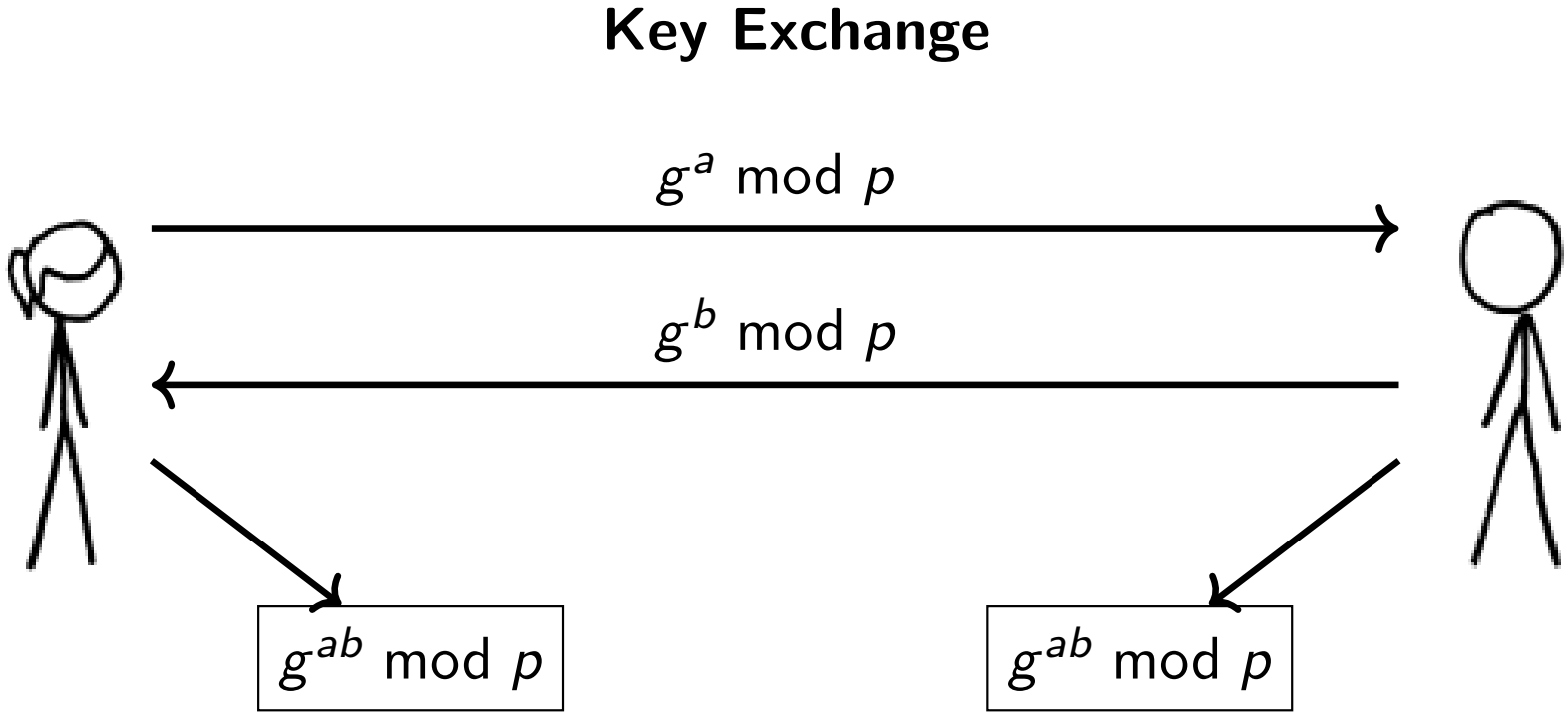
Textbook Diffie-Hellman

[Diffie Hellman 1976]

Public Parameters

p a prime

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



Diffie-Hellman is extremely common on the Internet

Protocol support for "mod p" Diffie-Hellman, spring 2015:

HTTPS Alexa Top 1M	68%
HTTPS Trusted cert	24%
SMTP StartTLS	41%
IMAPS	75%
POP3S	75%
SSH	100%
IPsec VPNs	100%

"Perfect Forward Secrecy"

"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."

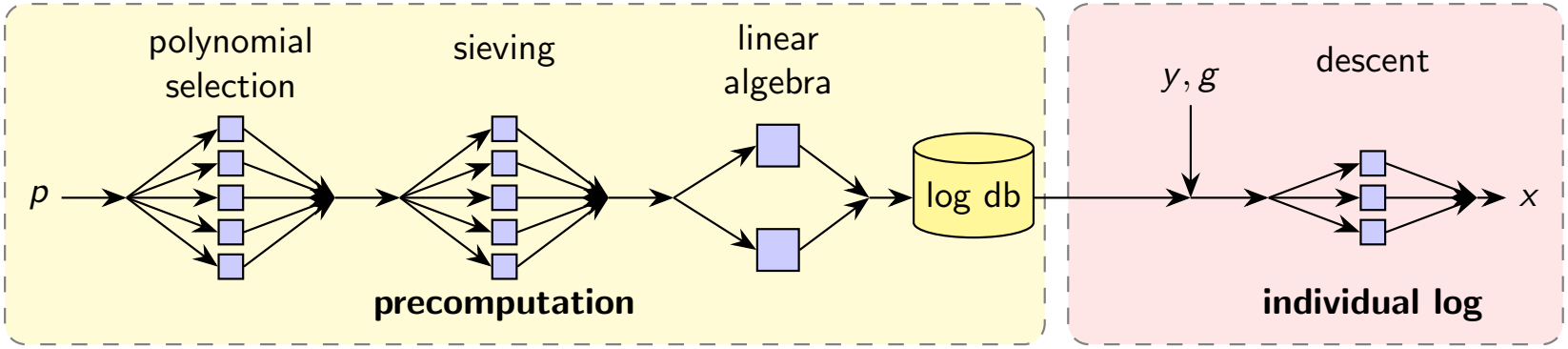
"With Perfect Forward Secrecy, anyone possessing the private key and a wiretap of Internet activity can decrypt nothing."

"Ideally the DH group would match or exceed the RSA key size but 1024-bit DHE is arguably better than straight 2048-bit RSA so you can get away with that if you want to."

"But in practical terms the risk of private key theft, for a non-ephemeral key, dwarfs out any cryptanalytic risk for any RSA or DH of 1024 bits or more; in that sense, PFS is a must-have and DHE with a 1024-bit DH key is much safer than RSA-based cipher suites, regardless of the RSA key size."

Cryptanalysis: number field sieve discrete log algorithm

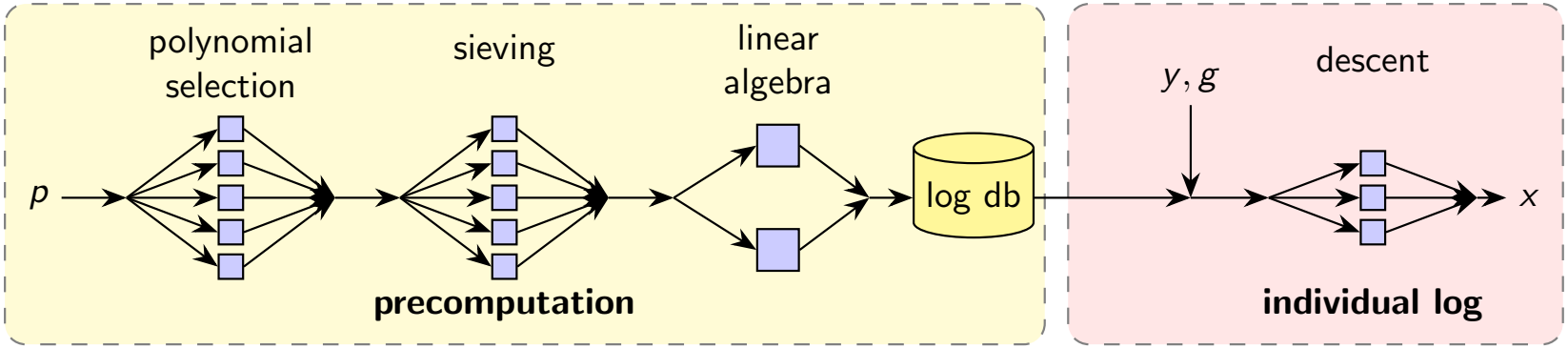
Goal: Given $g^x \equiv y \bmod p$, compute x.



$$L(1/3, 1.923) = \exp(1.923(\log p)^{1/3}(\log\log p)^{2/3})$$

Cryptanalysis: number field sieve discrete log algorithm

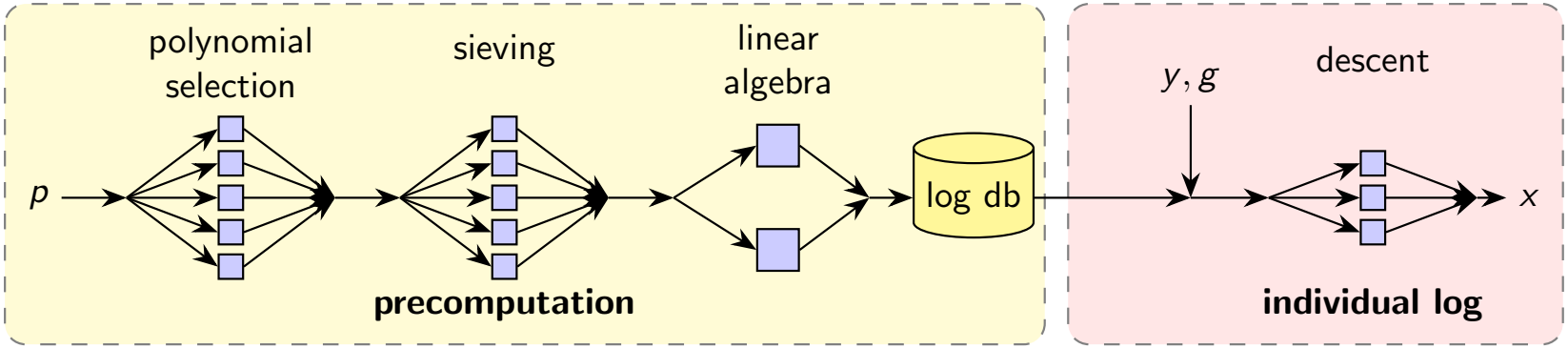
Goal: Given $g^x \equiv y \bmod p$, compute x.



$L(1/3, 1.923) = \exp(1.923(\log p)^{1/3}(\log\log p)^{2/3})$ $L(1/3, 1.232)$

Cryptanalysis: number field sieve discrete log algorithm

Goal: Given $g^x \equiv y \bmod p$, compute x.



$L(1/3, 1.923) = \exp(1.923(\log p)^{1/3}(\log\log p)^{2/3})$ $L(1/3, 1.232)$

	Sieving	Linear Algebra	Descent
RSA-512	0.5 core-years	0.33 core-years	
DH-512	2.5 core-years	7.7 core-years	10 core-mins

Precomputation can be done once and reused for many individual logs!

Our Results

Result #1: "Logjam": Active TLS MITM downgrade attack to 512-bit DHE export-grade cipher suites.

Diffie-Hellman TLS Handshake

hello, client random

list of cipher suites [. . . DHE . . .]

Diffie-Hellman TLS Handshake

hello, client random

list of cipher suites [. . . DHE . . .]

hello, server random, [DHE]

certificate = public RSA key + CA signatures

p, g, g^a, $\mathrm{Sign}_{\mathrm{RSAkey}}(p, g, g^a)$

Diffie-Hellman TLS Handshake

hello, client random

list of cipher suites [. . . DHE . . .]

hello, server random, [DHE]

certificate = public RSA key + CA signatures

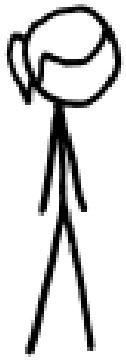
p, g, g^a, $\text{Sign}_{\text{RSAkey}}(p, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Diffie-Hellman TLS Handshake



hello, client random

list of cipher suites [. . . DHE . . .]

hello, server random, [DHE]

certificate = public RSA key + CA signatures

p, g, g^a, $\text{Sign}_{\text{RSAkey}}(p, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

client finished: $\text{Auth}_{k_{m_c}}(\text{dialog})$

server finished: $\text{Auth}_{k_{m_s}}(\text{dialog})$

$\text{Enc}_{k_e}(\text{request})$

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Export cipher suites in TLS

`TLS_RSA_EXPORT_WITH_RC4_40_MD5`
`TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5`
`TLS_RSA_EXPORT_WITH_DES40_CBC_SHA`

FREAK attack [BDFKPSZZ 2015]: Implementation flaw; use fast 512-bit factorization to downgrade modern browsers to broken export-grade RSA.

`TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA`
`TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA`
`TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA`
`TLS_DH_Anon_EXPORT_WITH_RC4_40_MD5`
`TLS_DH_Anon_EXPORT_WITH_DES40_CBC_SHA`

April 2015: 8.4% of Alexa top 1M HTTPS support `DHE_EXPORT`.

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

$[\ldots \text{DHE} \ldots]$

hello, server random, [DHE]

certificate = public RSA key + CA signatures

p, g, g^a, $\text{Sign}_{\text{RSAkey}}(p, g, g^a)$

g^b

$\text{KDF}(g^{ab}, \text{randoms}) \rightarrow k_{m_c}, k_{m_s}, k_e$

client finished: $\text{Auth}_{k_{m_c}}(\text{dialog})$

server finished: $\text{Auth}_{k_{m_s}}(\text{dialog})$

$\text{Enc}_{k_e}(\text{request})$

$\text{KDF}(g^{ab}, \text{randoms}) \rightarrow k_{m_c}, k_{m_s}, k_e$

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

~~[. . . DHE . . .]~~ [DHE_EXPORT]

hello, server random, [DHE]

certificate = public RSA key + CA signatures

p, g, g^a, $\text{Sign}_{\text{RSAkey}}(p, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

client finished: $\text{Auth}_{k_{m_c}}$(dialog)

server finished: $\text{Auth}_{k_{m_s}}$(dialog)

Enc_{k_e}(request)

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

~~[. . . DHE . . .]~~ [DHE_EXPORT]

hello, server random, [DHE_EXPORT]

certificate = public RSA key + CA signatures

p_{512}, g, g^a, $\text{Sign}_{\text{RSAkey}}(p_{512}, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

client finished: $\text{Auth}_{k_{m_c}}$(dialog)

server finished: $\text{Auth}_{k_{m_s}}$(dialog)

Enc_{k_e}(request)

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

~~[. . . DHE . . .]~~ [DHE_EXPORT]

hello, server random, ~~[DHE_EXPORT]~~[DHE]

certificate = public RSA key + CA signatures

p_{512}, g, g^a, $\text{Sign}_{\text{RSAkey}}(p_{512}, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

client finished: $\text{Auth}_{k_{m_c}}$(dialog)

server finished: $\text{Auth}_{k_{m_s}}$(dialog)

Enc_{k_e}(request)

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

~~[. . . DHE . . .]~~ [DHE_EXPORT]

hello, server random, ~~[DHE_EXPORT]~~[DHE]

certificate = public RSA key + CA signatures

p_{512}, g, g^a, $\text{Sign}_{\text{RSAkey}}(p_{512}, g, g^a)$

g^b

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

client finished: $\text{Auth}_{k_{m_c}}$(modified dialog)

server finished: $\text{Auth}_{k_{m_s}}$(modified dialog)

Enc_{k_e}(request)

KDF(g^{ab}, randoms) $\rightarrow$ k_{m_c}, k_{m_s}, k_e

Most hosts use the same parameters

Parameters hard-coded in implementations or built into standards.

97% of `DHE_EXPORT` hosts choose one of three 512-bit primes.

Hosts	Source	Year	Bits
80%	Apache 2.2	2005	512
13%	mod_ssl 2.3.0	1999	512
4%	JDK	2003	512

Top ten primes accounted for 99% of hosts.

Computing 512-bit discrete logs

- Carried out precomputation for Apache, mod_ssl primes.

	polysel	sieving	linalg	descent
	2000-3000 cores		288 cores	36 cores
DH-512	3 hours	15 hours	120 hours	70 seconds

- After 1 week precomputation, median individual log time 70s.
- Many ways attacker can work around delay.
- Logjam and our precomputations can be used to break connections to 8% of the HTTPS top 1M sites!

Daniel J. Bernstein @hashbreaker · Aug 12

@DLogBot m
5a2790dac75a8f9456da6f57ff117b1078f3a1472810a7bfdecb61ea8e43ce8fa16b
b019acf670ae98ed1cf9064b5a3f96fa5348ea5af7b949e10bf56b18f39f

4 5

CA Services
@DLogBot

Follow

.@hashbreaker
bada55ecc00031415926535897932 3

RETWEETS 8 FAVORITES 10

5:19 PM - 12 Aug 2015

Logjam mitigation

- Major browsers have raised minimum DH lengths:

  IE, Chrome, Firefox to 1024 bits; Safari to 768.

- TLS 1.3 draft includes anti-downgrade flag in client random.

Our Results

Result #1: "Logjam": Active TLS MITM downgrade attack to 512-bit DHE "export"-grade cipher suites.

Result #2: 1024-bit discrete log within range for governments. Parameter reuse allows wide-scale passive decryption.

Cost estimates for 768- and 1024-bit DHE and RSA

	Sieving core-years	Linear Algebra core-years	Descent core-time
RSA-512	0.5	0.33	
DH-512	2.5	7.7	10 mins
RSA-768	800	100	
DH-768	8,000	28,500	2 days
RSA-1024	1,000,000	120,000	
DH-1024	10,000,000	35,000,000	30 days

- Special-purpose hardware $\rightarrow \approx 80\times$ speedup
- $\approx$\$100Ms machine precomputes for one 1024-bit p every year
- Then, individual logs can be computed in close to real time

James Bamford, 2012, Wired

According to another top official also involved with the program, the NSA made an enormous breakthrough several years ago in its ability to cryptanalyze, or break, unfathomably complex encryption systems employed by not only governments around the world but also many average computer users in the US. The upshot, according to this official: "Everybody's a target; everybody with communication is a target."

[...]

The breakthrough was enormous, says the former official, and soon afterward the agency pulled the shade down tight on the project, even within the intelligence community and Congress. "Only the chairman and vice chairman and the two staff directors of each intelligence committee were told about it," he says. The reason? "They were thinking that this computing breakthrough was going to give them the ability to crack current public encryption."

Parameter reuse for 1024-bit Diffie-Hellman

- Precomputation for a single 1024-bit prime allows passive decryption of connections to 66% of VPN servers and 26% of SSH servers.

  (Oakley Group 2)

- Precomputation for a second common 1024-bit prime allows passive decryption for 18% of top 1M HTTPS domains.

  (Apache 2.2)

2013 NSA "Black Budget"

"Also, we are investing in groundbreaking cryptanalytic capabilities to defeat adversarial cryptography and exploit internet traffic."

This Exhibit is SECRET//NOFORN

Program	Expenditure Center	Project	FY 2011	FY 2012	FY 2013	FY 2012 - FY 2013 Change
	Computer Network Operations	Data Acquisition and Cover Support	56,949	100,987	117,605	16,618
		GENIE	615,177	636,175	651,743	15,568
		SIGINT Enabling	298,613	275,376	254,943	-20,433
	Computer Network Operations Total		**970,739**	**1,012,538**	**1,024,291**	**11,753**
	Cryptanalysis & Exploitation Services	Analysis of Target Systems	39,429	35,128	34,321	-807
		Cryptanalytic IT Systems	130,012	136,797	247,121	110,324
		Cyber Cryptanalysis	181,834	110,673	115,300	4,627
		Exploitation Solutions	90,024	59,915	58,308	-1,607
		Microelectronics	64,603	61,672	45,886	-15,786

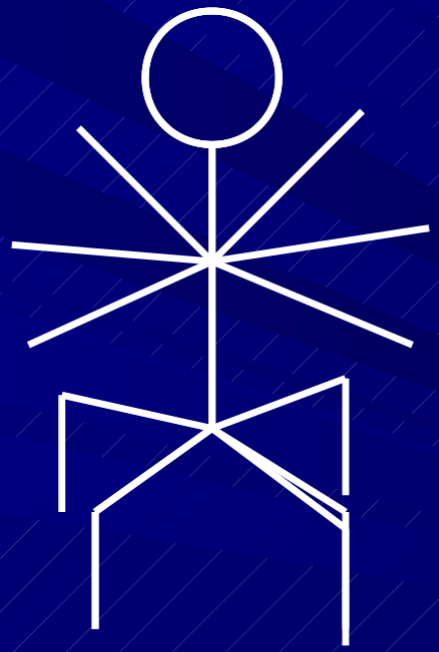
*numbers in thousands

4. Communicate Results

Can we decrypt the VPN traffic?

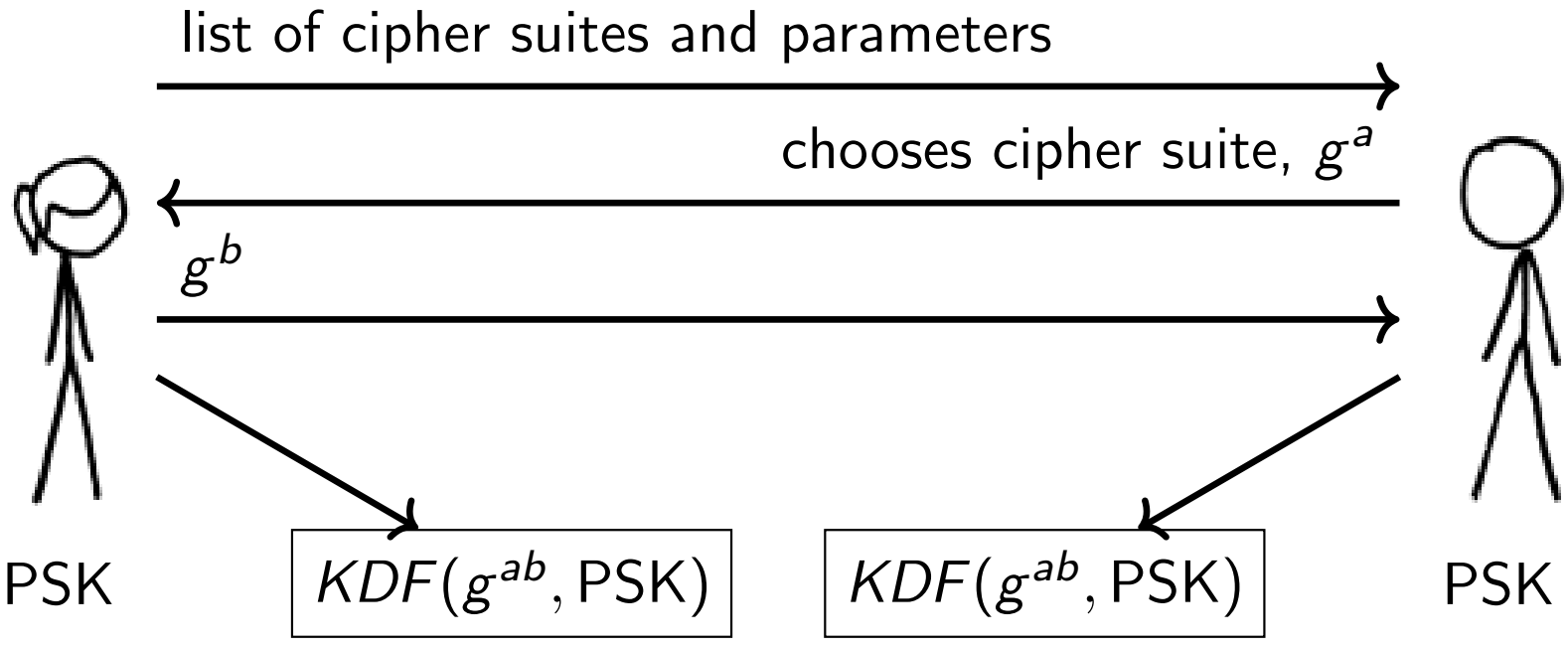
- If the answer is “No” then explain how to turn it into a “YES!”
- If the answer is “YES!” then…

Happy Dance!!



IKE Key Exchange for VPNs/IPsec

IKE chooses Diffie-Hellman parameters from standardized set.



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

- A 1024-bit DH break is a parsimonious explanation for NSA's large-scale passive decryption of VPN traffic.

NSA's on-demand IKE decryption requires:

- Known pre-shared key.
- Both sides of IKE handshake.
- Both IKE handshake and ESP traffic.
- IKE+ESP data is sent to HPC resources.

Discrete log decryption would require:

- Known pre-shared key.
- Both sides of IKE handshake.
- Both IKE handshake and ESP traffic.
- IKE data sent to HPC resources.

A well-designed "implant" would have fewer requirements.

Results and Mitigations

Result #1: "Logjam": Active TLS MITM downgrade attack to 512-bit DHE "export"-grade cipher suites.

Mitigations:

- Major browsers raised minimum DH lengths.
- TLS 1.3 draft anti-downgrade mechanism.

Result #2: 1024-bit discrete log within range for governments. Parameter reuse allows wide-scale passive decryption.

Mitigations:

- Move to elliptic curve cryptography
- If ECC isn't an option, use $\geq$ 2048-bit primes.
- If 2048-bit primes aren't an option, generate a fresh 1024-bit prime.

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

`weakdh.org`