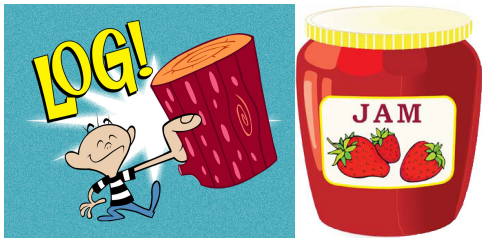


Diffie-Hellman, discrete logs, the NSA, and you

J. Alex Halderman
University of Michigan

Based on joint work:

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

22nd ACM Conference on Computer and Communications Security, CCS '15, October 2015. **Best paper award!**

https://weakdh.org

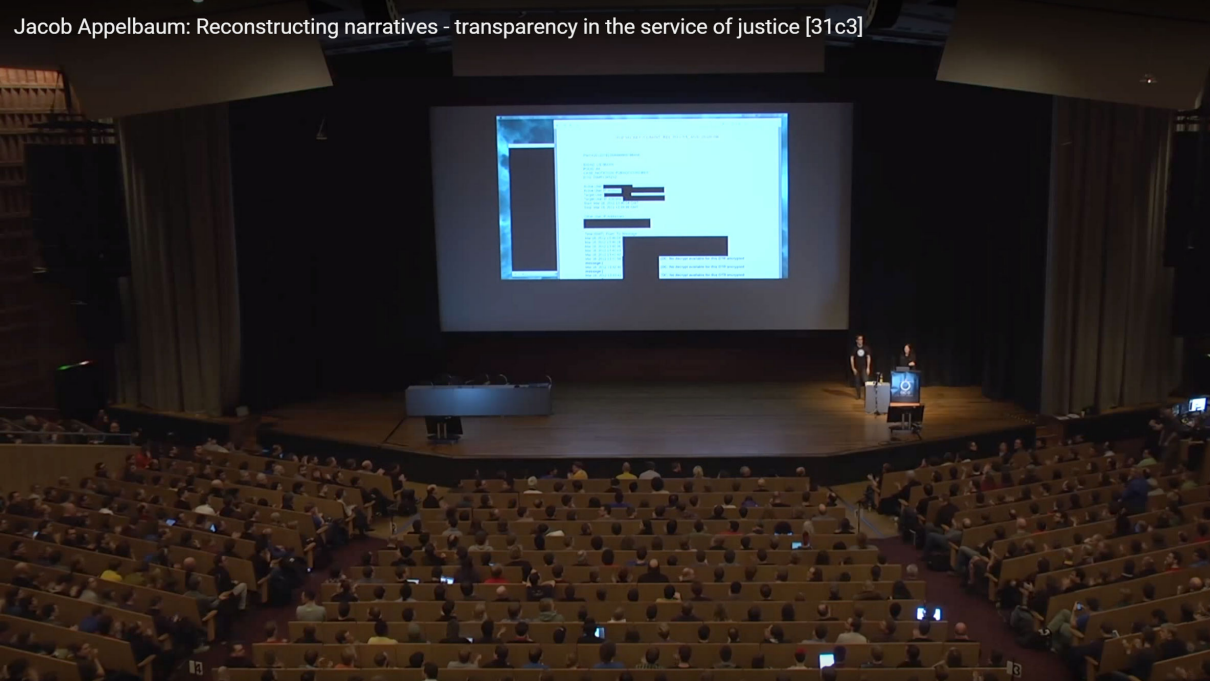
Jacob Appelbaum: Reconstructing narratives - transparency in the service of justice [31c3]

#6258

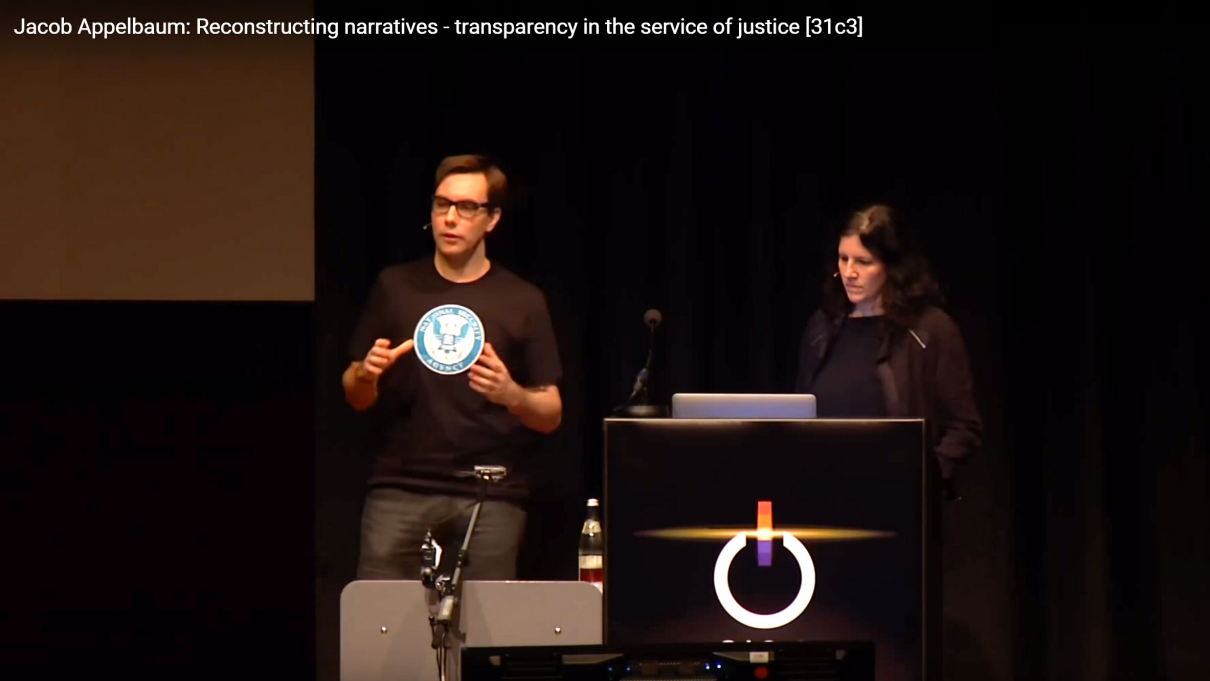
Jacob, Laura Poitras

Reconstructing narratives

transparency in the service of justice


Jacob Appelbaum: Reconstructing narratives - transparency in the service of justice [31c3]

Jacob Appelbaum: Reconstructing narratives - transparency in the service of justice [31c3]

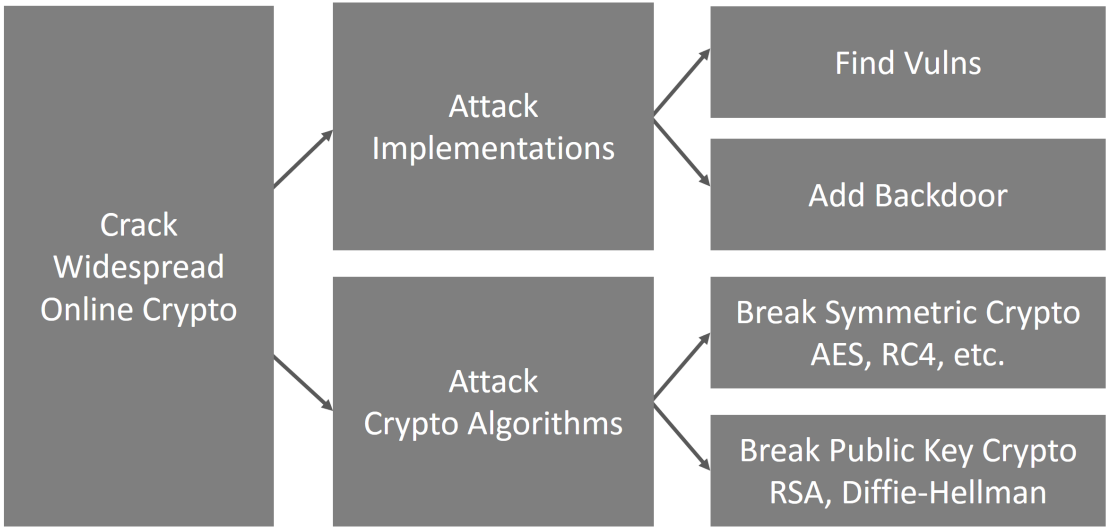


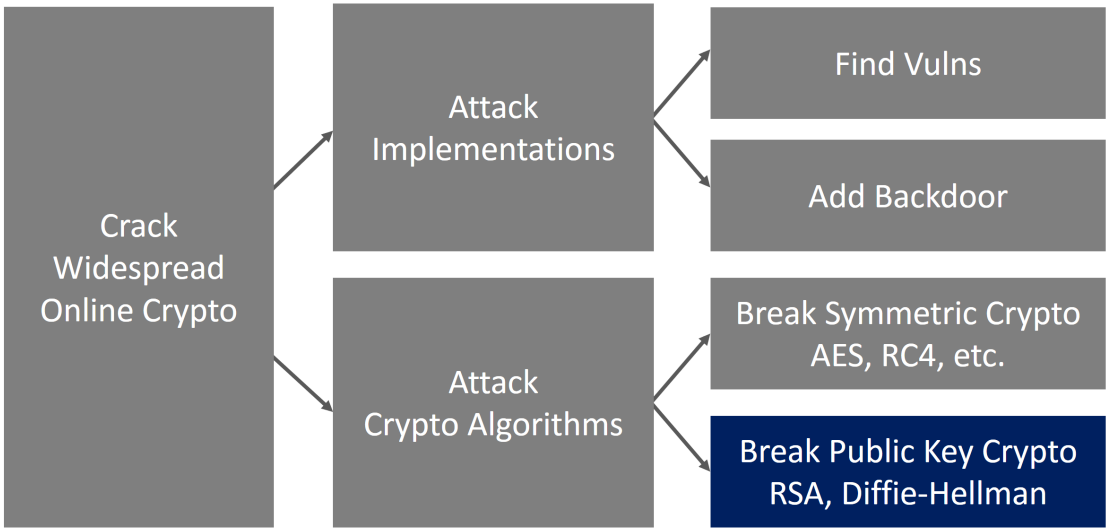
Prying Eyes: Inside the NSA's War on Internet Security

By SPIEGEL Staff

AP/dpa

US and British intelligence agencies undertake every effort imaginable to crack all types of encrypted Internet communication. The cloud, it seems, is full of holes. The good news: New Snowden documents show that some forms of encryption still cause problems for the NSA.


Crack Widespread Online Crypto
Attack Implementations
Find Vulns
Add Backdoor
Attack Crypto Algorithms
Break Symmetric Crypto AES, RC4, etc.
Break Public Key Crypto RSA, Diffie-Hellman


Crack Widespread Online Crypto
Attack Implementations
Find Vulns
Add Backdoor
Attack Crypto Algorithms
Break Symmetric Crypto AES, RC4, etc.
Break Public Key Crypto RSA, Diffie-Hellman

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Textbook RSA Encryption

[Rivest Shamir Adleman 1977]

Public Key

$N = pq$ modulus

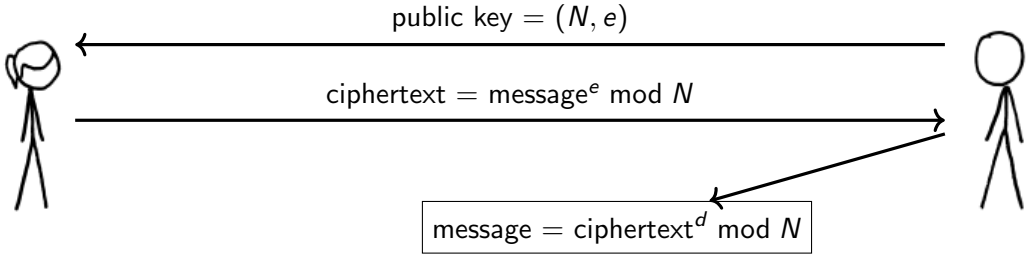
e encryption exponent

Private Key

p, q primes

d decryption exponent

$(d = e^{-1} \bmod (p-1)(q-1))$



RSA cryptanalysis

Factoring

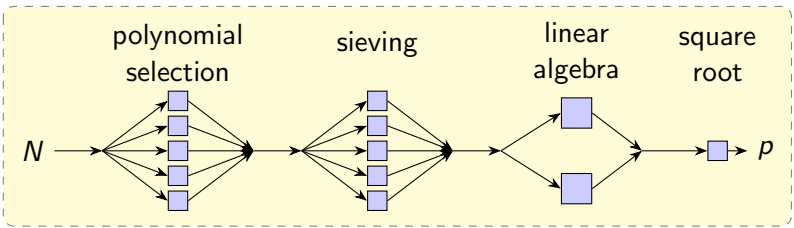
Problem: Factor N into p and q.

- Lets an attacker compute the private key.
- Factoring is much harder than multiplication.
- Best known algorithm: number field sieve.

Factoring with the number field sieve

Algorithm

1. **Polynomial selection** Choose a good number field.
2. **Relation finding** Factor many small-ish integers.
3. **Linear algebra** Use the factorizations to construct squares.
4. **Square root** Take square roots and check if factor N.



How long does it take to factor using the number field sieve?

Answer 1:

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

How long does it take to factor using the number field sieve?

Answer 1:

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

Answer 2:

512-bit RSA: $<$ 1 core-year. (4 hours + \$75 on EC2! seclab.upenn.edu/projects/faas/)

768-bit RSA: $<$ 1,000 core-years. ($<$ 1 calendar year)

1024-bit RSA: $\approx$ 1,000,000 core-years.

2048-bit RSA: Minimum recommended key size today.

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

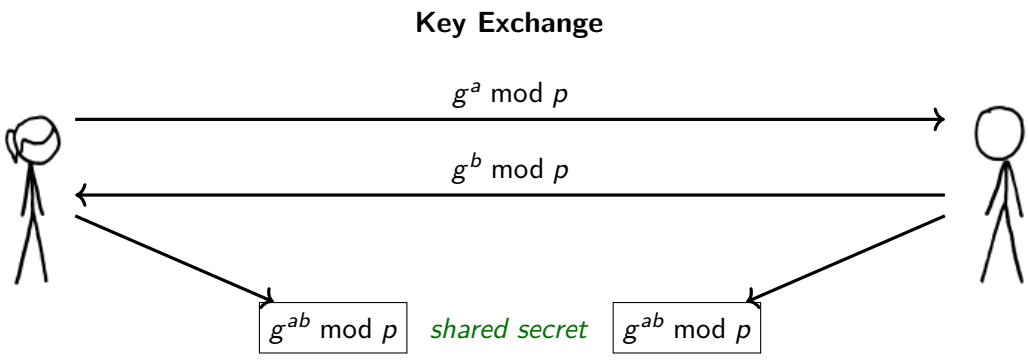
"We stand today on the brink of a revolution in cryptography." – November 1976

Textbook Diffie-Hellman

Public Parameters

p a prime

$g < p$ (often 2 or 5)



Textbook Diffie-Hellman

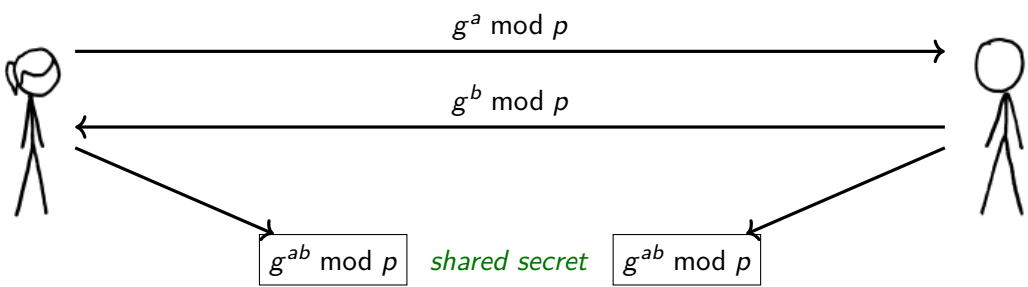
Public Parameters

p a prime

$g < p$ (often 2 or 5)

Provides perfect forward secrecy:
Can't later hack Alice or Bob to decrypt connections intercepted today.*

Key Exchange



Advocating Diffie-Hellman over RSA for 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."

We were wrong. We're sorry. :(

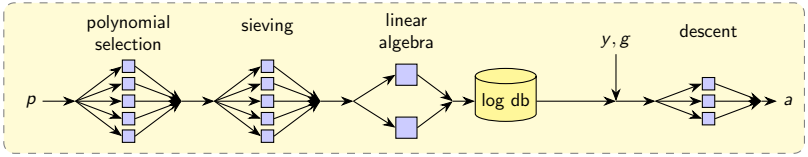
Diffie-Hellman cryptanalysis

Discrete Log

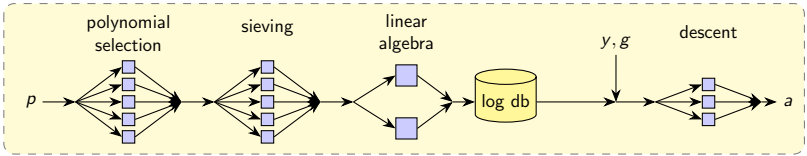
Problem: Given $y = g^a \bmod p$, compute a.

- Allows attacker to compute shared key.
- Discrete log is much harder than modular exponentiation.
- Best known algorithm: number field sieve for discrete log.

Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm



Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm

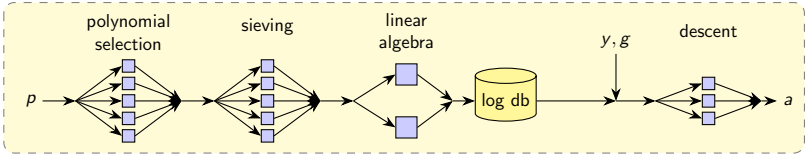


How long does the number field sieve take?

Answer 1:

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

Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm



How long does the number field sieve take?

Answer 2:

512-bit DH: $\approx$ 10 core-years.

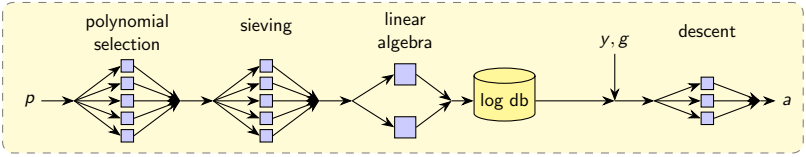
768-bit DH: $\approx$ 35,000 core-years.

1024-bit DH: $\approx$ 45,000,000 core-years.

2048-bit DH: Minimum recommended key size today.

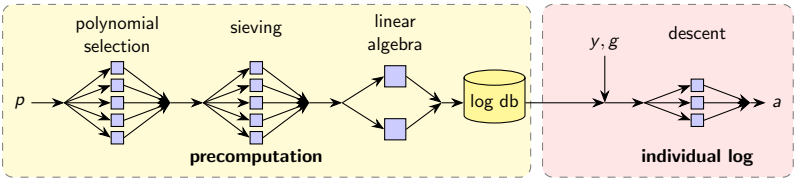
Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm

But... What if you want to break many connections that use the same public parameter p?



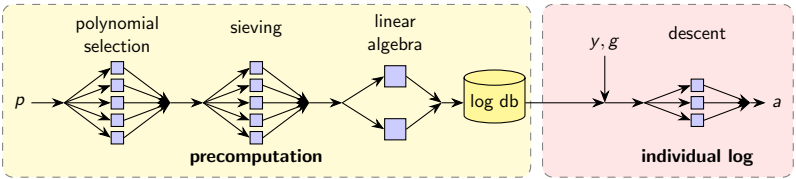
Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm

But... What if you want to break many connections that use the same public parameter p?



Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm

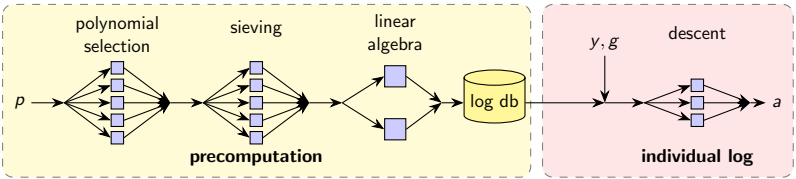
But... What if you want to break many connections that use the same public parameter p?



Uh oh!

Diffie-Hellman cryptanalysis: number field sieve discrete log algorithm

But... What if you want to break many connections that use the same public parameter p?



Uh oh!

	Precomputation	Individual Log
DH-512	10 core-years	10 core-minutes
DH-768	35,000 core-years	2 core-days
DH-1024	45,000,000 core-years	30 core-days

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

Exploiting Diffie-Hellman

Logjam attack:
Anyone can use HTTPS backdoors from '90s crypto war to pwn modern browsers.

International Traffic in Arms Regulations

April 1, 1992 version

Category XIII--Auxiliary Military Equipment ...

(b) Information Security Systems and equipment, cryptographic devices, software, and components specifically designed or modified therefore, including:

(1) Cryptographic (including key management) systems, equipment, assemblies, modules, integrated circuits, components or software with the capability of maintaining secrecy or confidentiality of information or information systems, except cryptographic equipment and software as follows:

(i) Restricted to decryption functions specifically designed to allow the execution of copy protected software, provided the decryption functions are not user-accessible.

(ii) Specially designed, developed or modified for use in machines for banking or money transactions, and restricted to use only in such transactions. Machines for banking or money transactions include automatic teller machines, self-service statement printers, point of sale terminals or equipment for the encryption of interbanking transactions.

Commerce Control List: Category 5 - Info. Security

a.1.a. A symmetric algorithm employing a key length
in excess of 56-bits; or

a.1.b. An asymmetric algorithm where the security of the
algorithm is based on any of the following:

a.1.b.1. Factorization of integers in excess of 512 bits (e.g., RSA);

a.1.b.2. Computation of discrete logarithms in a multiplicative
group of a finite field of size greater than 512 bits (e.g., Diffie-Hellman
over Z/pZ); or

a.1.b.3. Discrete logarithms in a group other than mentioned
in 5A002.a.1.b.2 in excess of 112 bits (e.g., Diffie-Hellman
over an elliptic curve);

a.2. Designed or modified to perform cryptanalytic functions;

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
```

```
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
```

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. Affected most browsers and 9.6% of Alexa top million HTTPS sites.

```
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
```

Logjam attack: Protocol flaw; use fast 512-bit discrete log to downgrade modern browsers to broken export-grade DH. Affected all browsers and 8.4% of Alexa top million HTTPS sites.

Logjam: Active downgrade attack to export Diffie-Hellman

Protocol flaw: Server does not sign chosen cipher suite.

hello, client random

[. . . 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

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

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.

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

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

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

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

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

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

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

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

Enc_{k_e}(request)

KDF(g^{ab}, randoms) $\rightarrow$ k_{mc}, k_{ms}, k_e

KDF(g^{ab}, randoms) $\rightarrow$ k_{mc}, k_{ms}, k_e

How widely shared are Diffie-Hellman public parameters?

How widely shared are Diffie-Hellman public parameters?

We used Internet-wide scanning to find out:

- Parameters hard-coded in many implementations or built into standards.
- 97% of hosts that support `DHE_EXPORT` chose 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.

Attacking the most common 512-bit primes

- Carried out precomputation for Apache, mod_ssl, OpenSSL 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.
- Logjam and our precomputations can be used to break connections to 8% of the HTTPS top 1M sites!

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.

CA Services Test @CAServicesBot · 8s

@DLogBot What's up bot?

1

CA Services

@DLogBot

Following

@CAServicesBot Thank you for using the CA Services discrete logarithm bot. Your request should be <group (a/m/o)> <ephemeral key in hex>

3:58 PM - 12 Aug 2015

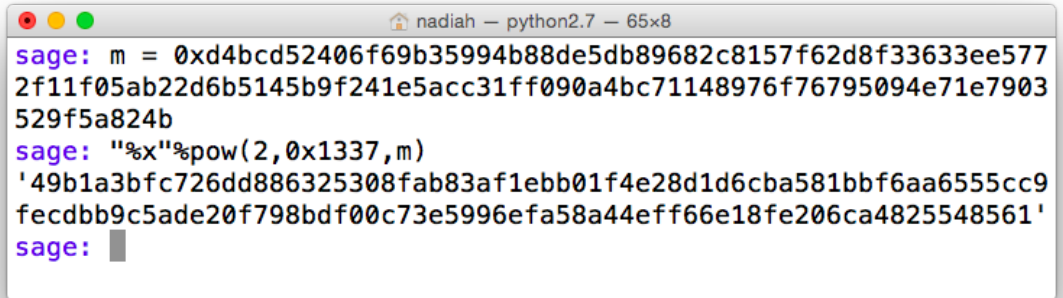
Reply to @DLogBot

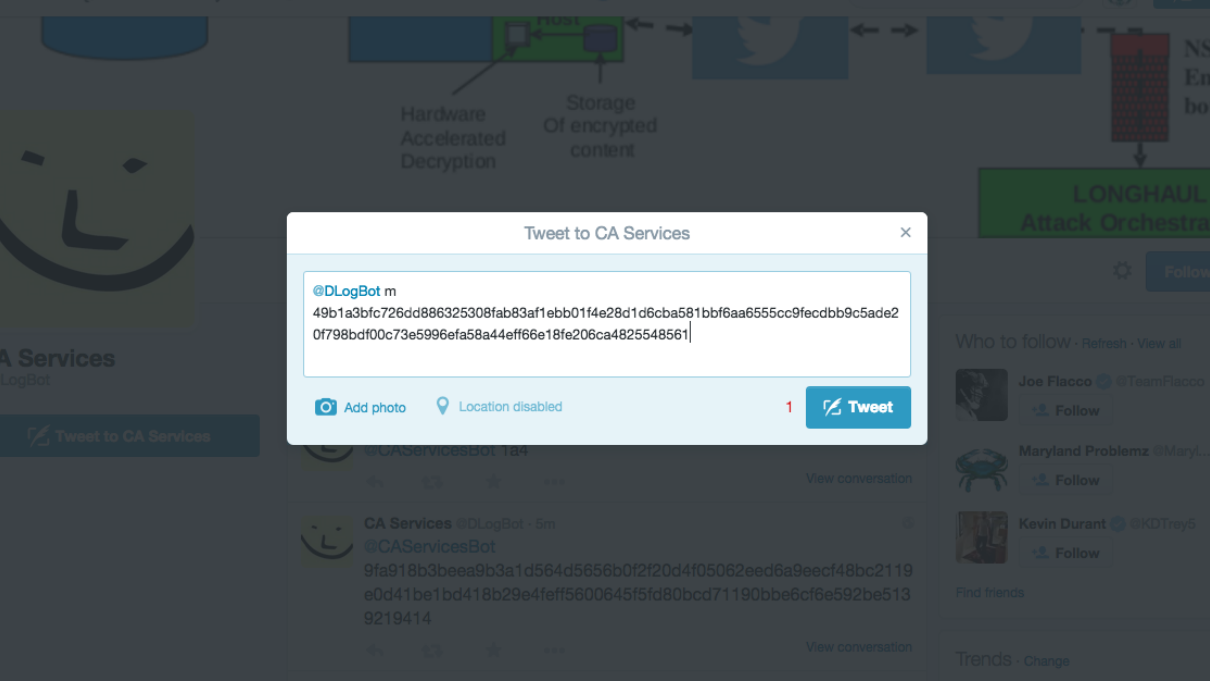
g = 2

apache:
9fdb8b8a004544f0045f1737d0ba2e0b274cdf1a9f588218fb43
5316a16e374171fd19d8d8f37c39bf863fd60e3e300680a3030c
6e4c3757d08f70e6aa871033

openssl:
da583c16d9852289d0e4af756f4cca92dd4be533b804fb0fed94e
f9c8a4403ed574650d36999db29d776276ba2d3d412e218f4dd1e
084cf6d8003e7c4774e833

mod_ssl:
d4bcd52406f69b35994b88de5db89682c8157f62d8f33633ee577
2f11f05ab22d6b5145b9f241e5acc31ff090a4bc71148976f7679
5094e71e7903529f5a824b


nadiah — python2.7 — 65×8
sage: m = 0xd4bcd52406f69b35994b88de5db89682c8157f62d8f33633ee5772f11f05ab22d6b5145b9f241e5acc31ff090a4bc71148976f76795094e71e7903529f5a824b
sage: "%x"%pow(2,0x1337,m)
'49b1a3bfc726dd886325308fab83af1ebb01f4e28d1d6cba581bbf6aa6555cc9fecdbb9c5ade20f798bdf00c73e5996efa58a44eff66e18fe206ca4825548561'
sage:


Hardware
Accelerated
Decryption
Storage
Of encrypted
content
LONGHAUL
Attack Orchestra
Tweet to Tweet to CA Services
@DLogBot m
49b1a3bfc726dd886325308fab83af1ebb01f4e28d1d6cba581bbf6aa6555cc9fecdbb9c5ade2
0f798bdf00c73e5996efa58a44eff66e18fe206ca4825548561
Add photo
Location disabled
1
Tweet
@CAServicesBot 1a4
View conversation
CA Services @DLogBot · 5m
@CAServicesBot
9fa918b3beea9b3a1d564d5656b0f2f20d4f05062eed6a9eecf48bc2119
e0d41be1bd418b29e4feff5600645f5fd80bcd71190bbe6cf6e592be513
9219414
View conversation
Who to follow · Refresh · View all
Joe Flacco @TeamFlacco
Follow
Maryland Problemz @MaryL
Follow
Kevin Durant @KDTrey5
Follow
Find friends
Trends · Change

```
90075569388022478418182684628050789
Wed Aug 12 18:45:33 2015 Deduced log of (470366623, 338113733, 1) from rel: 132375484352441692441431001820948733803608386541927990642258338281320222181965758481072535570446848729782370195553151
2602180142026876687466257886567740811
Wed Aug 12 18:45:33 2015 Deduced log of (25962401291, 20249792848, 0) from rel: 50563330737415960415049575328637460098249698959575974061877011070448070162405342698195719255905794579609010675528
95341537108850795747444342377377252424012
Wed Aug 12 18:45:33 2015 Deduced log of (152250701, 93107759, 0) from rel: 3686228138501582831528267989299074122849023771854691788298703712891447715918764923496597009693500652769104860220785305
619414614604451268227923135417596522
Wed Aug 12 18:45:33 2015 Deduced log of (143560981, 107192704, 1) from rel: 307860298231873673521041039296250709974237070003579136705638665216739899169781455078181073498093656533016603801659935
2969318604395616575043296456952989875
Wed Aug 12 18:45:33 2015 Deduced log of (278147959, 4410170, 1) from rel: 25056849574856392499222308446576938841269118670947370888901048538967385274262761282032832317465502707565100993756859432
82967400011055224573218135260330208
Wed Aug 12 18:45:33 2015 Deduced log of (577776097, 143348514, 1) from rel: 442338900520785256843955526485021247124215862664804946530253545141243702700937987405857426583528963422304799351630803
7919138940654137304654435108007690916
Wed Aug 12 18:45:33 2015 Deduced log of (418914989, 316721060, 0) from rel: 367207948543056444750000809717878390413543347607219059542512916467040252344925831757084814255582491568962667272894312
8947454323548300302225949416351771128
Wed Aug 12 18:45:33 2015 Deduced log of (338071753, 301777175, 1) from rel: 107520094080132598180876691823882310946126491164966924787589553472432468656483029744406591626920263170944975959228432
8146328557329068755741596486570153269
Wed Aug 12 18:45:33 2015 Deduced log of (334373141, 196775704, 0) from rel: 295924098567118682024550377212916862876453441769461426617303202625502622200306437293865652865355209952938669043254101
9564883269282155862438329453684513068
Wed Aug 12 18:45:33 2015 Deduced log of (85126349747, 78086389814, 1) from rel: 27200889150733468645844292420323315569132323223885332400187747723487106724374323444054210030340495981561444935649
29239372773533486332215126820403976503407
Wed Aug 12 18:45:33 2015 Deduced log of (61692766831901, 45737121448216, 0) from rel: 95038570194681056849283585551975517533527177793714128774971758086342076709285287095228231851435952966745662
5541155358927946669460742386945831281675936023
Wed Aug 12 18:45:33 2015 Deduced log of (206206811, 193918773, 1) from rel: 150129281584625653670608607473569366162828241135446571302979567882130921962371540905953102043599924796633201952837388
9419796552335493328120146276606866566
Wed Aug 12 18:45:33 2015 Deduced log of (226880903, 205068533, 1) from rel: 536845220976247216070823275266198080149544131379113941538069691920397796698875020561335789576215785486392872542467451
47897740996481184405739629902957951
Wed Aug 12 18:45:33 2015 Deduced log of (292902821, 134190188, 1) from rel: 439445138364582159970955146631236342487943352438465485528919566060271737879221218557114979606253028169269932103194756
5471325659574647516602447049908705083
Wed Aug 12 18:45:33 2015 Deduced log of (1307702399, 951504975, 0) from rel: 38841298453945470805423616555117535234792510892661146266849568329271564902616188183902560240858545567709177219978557
80328963175738364980433171357469136010
Wed Aug 12 18:45:33 2015 Deduced log of (202009439, 132696313, 1) from rel: 586746728022223145407986515759301518747360666540921337722240732928530330678243336945997816779047558926178822228470304
876293629942892685459127281244685422
Wed Aug 12 18:45:33 2015 Deduced log of (277410803, 9791152, 1) from rel: 22590019166737223971750805375787300510697656341358256117345964220989351226937717614398389638816685131456089057250849854
08190370647197782845966543502798213
Wed Aug 12 18:45:33 2015 Deduced log of (5077939123, 4652063990, 1) from rel: 4700476107471785387546249836018003783883820730918635794744673497035689116025731296286088760225494958896998096151611
592457429278405080150250145357601977064
Wed Aug 12 18:45:33 2015 Deduced log of (570419228219, 397375869245, 0) from rel: 325565485301564285408441912676148829488912833584028431503330529912489939420417223758408743360854620751701701700
2433773924631799147022786959354304587524964
Wed Aug 12 18:45:33 2015 Deduced log of (149460671, 53992188, 1) from rel: 2651811832073407132611177858759137310517057082625851781864665609907080436715896016549107955625430256337426585266782281
625402584417872997921002003028988191
Wed Aug 12 18:45:33 2015 Deduced log of (1004549233, 320473819, 1) from rel: 43645924758722777996430259828303303626139844034933653422361856992233309077823314341678196465942192928524784534230875
36704765635181511493534233116851018079
Wed Aug 12 18:45:33 2015 Deduced log of (500400001, 21453686, 0) from rel: 1824485543248714996126611031082238658686942161900511996641160312138508104939395699652788432709468236780128943689748040
801275264279266440000948459876830721
Wed Aug 12 18:45:33 2015 # p=11141973616799305182672125953821539621789863864652082189484418383755797726500394212227314338509410225240621127512913527972129628050063918116985983184044619
Wed Aug 12 18:45:33 2015 # ell=55709868083996525913360629769107698108949319323260410947422091918778988632501971061136571692547051126203105637564567639860648140250319590584991592022309
Wed Aug 12 18:45:33 2015 log(2)=47197552632340568686371246544797334661553247954679670451641942805650991309645074077087810506576181574948593419547569918183935151116719990414766766997927305
Wed Aug 12 18:45:33 2015 log(3)=2049575714412193487722204790266300362401149174396124002943447498737082565025944343998728517979935372577736658606725714791522464109941469995680192346636733
Wed Aug 12 18:45:33 2015 log(7)=560851705539270178596366566826552536899471430269845959055017872321665733107029846170341120694186605686700691326115179252433760356477128350339490874729626
Wed Aug 12 18:45:33 2015 # target=3859662976628548511009844324443395384136719293219082073451893466257682376593532665837044091024437857496590340632262194167030009888816048652618833300063585
Wed Aug 12 18:45:33 2015 log(target)=2174109246973388728641750598631118018861306904316653371885963544518062050840597343884563900467512428378983902294307224745620816837459453497430898847451692
Wed Aug 12 18:45:33 2015 Final consistency check ok!
```

CA Services Test @CAServicesBot · 3m

@DLogBot m
49b1a3bfc726dd886325308fab83af1ebb01f4e28d1d6cba581bbf6aa6555cc9fe
cdbb9c5ade20f798bdf00c73e5996efa58a44eff66e18fe206ca4825548561

1

CA Services
@DLogBot

Following

@CAServicesBot 1337

3:45 PM - 12 Aug 2015

Reply to @DLogBot

@DLogBot m
5a2790dac75a8f9456da6f57ff117b1078f3a1472810a7bfdecb61ea8e43ce8fa16b
b019acf670ae98ed1cf9064b5a3f96fa5348ea5af7b949e10bf56b18f39f

4 5 •••

CA Services
@DLogBot

Follow

.@hashbreaker
bada55ecc00031415926535897932 3

RETWEETS 8 FAVORITES 10

5:19 PM - 12 Aug 2015

•••

Exploiting Diffie-Hellman

Logjam attack:
Anyone can use backdoors from '90s crypto war to pwn modern browsers.

Mass surveillance:
Governments can exploit 1024-bit discrete log for wide-scale passive decryption.

Is breaking 1024-bit Diffie-Hellman within reach of governments?

	Precomputation core-years	Individual Log core-time
RSA-512	1	—
DH-512	10	10 mins
RSA-768	1,000	—
DH-768	35,000	2 days
RSA-1024	1,000,000	—
DH-1024	45,000,000	30 days

Is breaking 1024-bit Diffie-Hellman within reach of governments?

	Precomputation core-years	Individual Log core-time
RSA-512	1	—
DH-512	10	10 mins
RSA-768	1,000	—
DH-768	35,000	2 days
RSA-1024	1,000,000	—
DH-1024	45,000,000	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."

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

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.

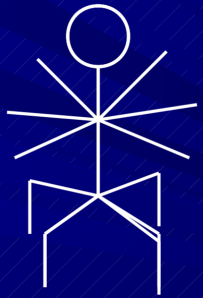
  (Apache 2.2)

4. Communicate Results

Can we decrypt the VPN traffic?

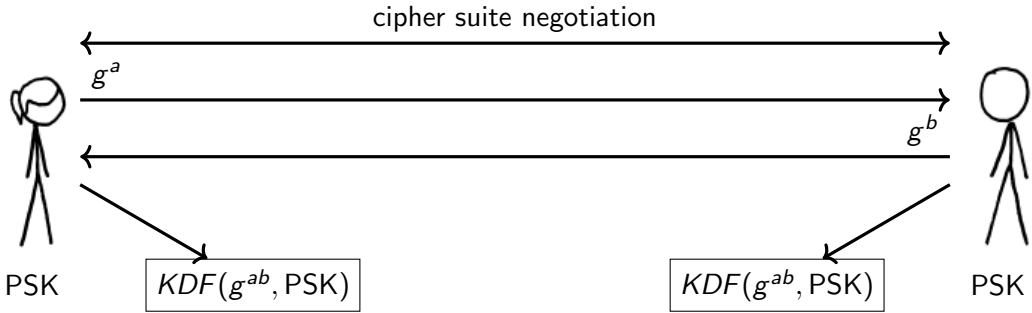
- If the answer is “No” then explain how to turn it into a “YES!”
- If the answer is “YES!” then…

Happy Dance!!



IKE Key Exchange for IPsec VPNs

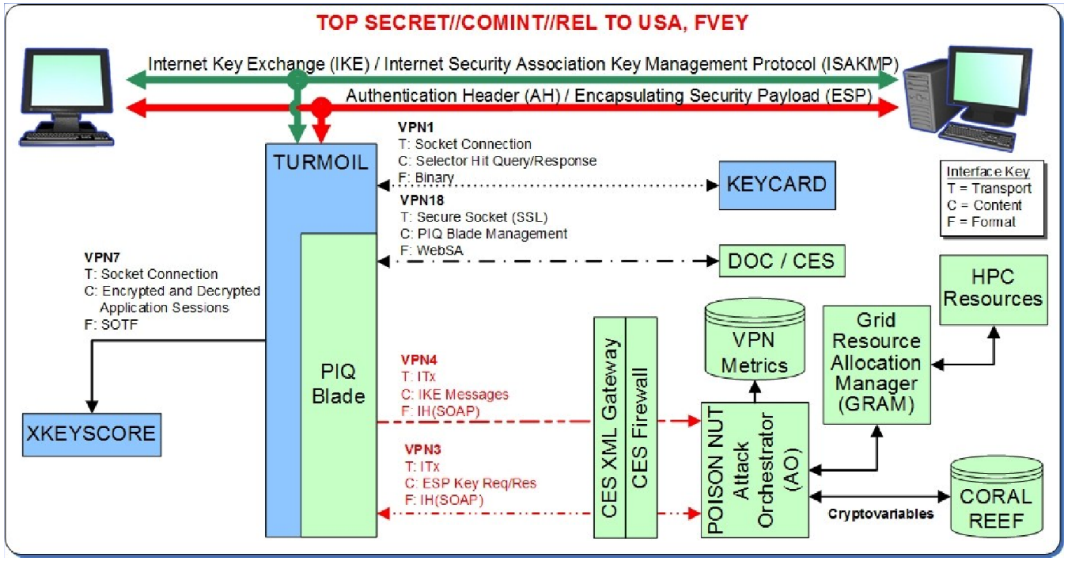
IKE chooses Diffie-Hellman parameters from standardized set.



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

NSA VPN Attack Orchestration



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.

	Vulnerable servers, if the attacker can precompute for . . .			
	all 512-bit p	all 768-bit p	one 1024-bit p	ten 1024-bit p
HTTPS Top 1M MITM	45K (8.4%)	45K (8.4%)	205K (37.1%)	309K (56.1%)
HTTPS Top 1M	118 (0.0%)	407 (0.1%)	98.5K (17.9%)	132K (24.0%)
HTTPS Trusted MITM	489K (3.4%)	556K (3.9%)	1.84M (12.8%)	3.41M (23.8%)
HTTPS Trusted	1K (0.0%)	46.7K (0.3%)	939K (6.56%)	1.43M (10.0%)
IKEv1 IPv4	–	64K (2.6%)	1.69M (66.1%)	1.69M (66.1%)
IKEv2 IPv4	–	66K (5.8%)	726K (63.9%)	726K (63.9%)
SSH IPv4	–	–	3.6M (25.7%)	3.6M (25.7%)

Diffie-Hellman Attacks and Mitigations

Logjam attack:

Anyone can use backdoors from '90s crypto war to pwn modern browsers.

Mitigations:

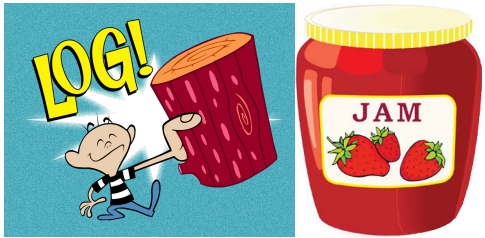
- Major browsers raised minimum DH lengths.
- TLS 1.3 draft anti-downgrade mechanism.
- Recommendation: Don't backdoor crypto!

Mass surveillance:

Governments can exploit 1024-bit discrete log for 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.



Diffie-Hellman, discrete logs, the NSA, and you

J. Alex Halderman
University of Michigan

https://weakdh.org