Diffie-Hellman mod 630(427!+1)+1

Andrew Allen and Dan Brown, BlackBerry

CFRG, Prague, 2017 July 18

Gordon’s attack and current countermeasures

- D. M. Gordon, *Designing and detecting trapdoors for discrete log cryptosystems,* (CRYPTO conference), 1992.
  - A **backdoor** embedded into a Diffie-Hellman prime
  - Hidden vulnerability to special number field sieve (SNFS) attack
- J. Fried and P. Gaudry and N. Heninger and E. Thomé *A kilobit hidden SNFS discrete logarithm computation* http://eprint.iacr.org/2016/961
  - Realistic 1024-bit prime example
- Countermeasures that seem to work okay:
  - Derive p from pi or e [Gordon]
    - IPSec, TLS (e.g. RFC 7919): fixed DH primes use Gordon’s methods.
  - Derive p (and q) using pseudorandom hash [NIST]
  - Bonus: hash or pi looks random, reduces risk of other special weakness?

Benefits of p=630(427!+1)+1

- Compact description has only little room for **trapdoor**
  - Even **more compact** than using e, pi or hash
  - E.g. RFC 7919, ffdhe3072: $p=2^{3072}-2^{3008}+([e2^{2942}]+2625351)2^{64}-1$
    - (39 symbols by adding ^ for exponentiation, instead of 13).
- Diffie-Hellman **secure** as discrete log:
  - q-1 a product 1*2*3*…*427 of small numbers (p=hq+1)
  - den Boer proof nearly optimal (among SNFS-resistant primes)
  - Such a reduction (e.g. den Boer) **out of reach** for current primes?
- 3000+ bits: can **protect** 128-bit keys (AES, etc.)
- Small cofactor 630 **resists** small-subgroup attacks effectively

Heuristics about 630(427!+1)+1

- Heuristic: factorials are **special** in sense they are NOT small polynomials evaluated at small inputs
  - Else factoring would be easy
    - Write floor(sqrt(n))! as polynomial, evaluate mod n. Take gcd. [BBS?]
  - Weakly suggests that 630(427!+1)+1 not vulnerable to SNFS
- Heuristic: p has many zero bits in binary expansion
  - Suggests Diffie-Hellman using p ought to be a bit faster than random prime (due to faster **Barrett reduction**)

Extra slides

- On den Boer’s reductions
- Why use classic DH at all?
- General background review
  - Diffie-Hellman key exchange
  - Special number field sieve

Diffie-Hellman needs more than discrete log!

- DLP: g^x mod p ----> x
- DHP: g^x, g^y mod p ----> g^{xy} mod p
- If q-1 smooth (product of small numbers), then den Boer showed

Diffie-Hellman problem (DHP)

is nearly as hard as

discrete log problem (DLP)

- Gordon/NIST primes usually have q-1 random => not smooth
  - Factor of size q^(2/3) usually expected
  - den Boer proof does not apply
  - Alternatives: Maurer-Wolf, or Boneh-Lipton (looser, more complex)

The den Boer reduction

- Let G have prime order q mod p. (Note q|p-1.)
- Suppose DH(G^x, G^y)=G^(xy) was easy to compute.
- Let F be a field of size q.
- Represent x in F by G^x. Call this representation of the field G^F.
- Implement G^F: $G^{x+y}=G^xG^y$ and $G^{xy}=DH(G^x,G^y)$.
- To find x from G^x, try to solve discrete log in G^F.
- Log in G^F: given G^b and G^x, find t such that $G^x=G^{b^t}$.
- Since q-1 is smooth, use Pohlig-Hellman (PH) to quickly find t.
- Note: PH is group-generic, so it work in mult-group of G^F.

Why classic Diffie-Hellman in modern world?

- Older than elliptic curve (dhinosaurs of public-key crypto)
  - Older => safer (more studied)?
- If Alice and Bob have enough computing and communication power, they can use multiple public-key cryptographic algorithms, e.g.:
  - ECDH (multiple curves?)
  - Post-quanta algorithm(s)
  - RSA
  - **DH (classic DH – per this presentation)**
- I.e. sum independently established 128-bit keys
  - Secure if any 1 of the key establishments are secure.

Review: primes p,q in DH exchange

- Usually take $p = 2q+1$ for q prime
- Call p a safe prime (and q a Sophie Germaine prime)
- NIST, for digital signature algorithm (DSA), chooses a much smaller prime q with $p=hq+1$ for h large
  - Smaller signatures, risk of small-subgroup attack from large h
- Alice picks random a, Bob random b
- Alice compute $A=g^a \bmod p$, Bob $B=g^b \bmod p$. Exchange A, B.
- Shared secret is $A^b \bmod p = B^a \bmod p$.
- Usually: g has order q (or small multiple of q)

Special number field sieve (SNFS)

- Weak primes p of certain special form
  - Small-coefficient polynomials evaluated at a small input, e.g. sums of powers
  - Weaker than random primes due to SNFS
    - Random primes only vulnerable to general NFS (which is slower than SNFS)
- Unfortunately, the main faster-than-random primes
  - Mersenne primes (and like) are weaker for DH,
      - Side note: these types of primes okay for ECC <= no SNFS on ECC
  - Because they are also vulnerable to SNFS (sums of powers)
  - Note: Some DH systems use these special fast primes despite SNFS-risk
    - SNFS still infeasible at their key sizes,
    - Special form may avoid some other (hypothetical and unpublished) attack ???