800-90 and Dual EC DRBG

John Kelsey, NIST
RNG Standardization

• Random numbers needed for cryptography

• **X9.82**: Standards effort in X9F1 (banking standards org)
  • Started around 1998 (I came onboard in 2003)
  • Made very little progress early on
  • Eventually became mainly a US government effort
  • NIST and NSA, with some participation from CSE
Moving to NIST Special Publications

- X9 Documents not available to public
  - Hard to get feedback from academics
- X9 process was slow
- X9 not tuned to needs of FIPS validation

Most of work on standards done by US federal employees (NIST and NSA, with some help from CSE)
Three Documents

• **SP 800-90A: Deterministic Random Bit Generators**

• SP 800-90B: Entropy Sources

• SP 800-90C: Putting it All Together

*Derived partially from the work done in X9.82.*
Algorithms in 800-90A

- CTR-DRBG = block cipher based
- HMAC-DRBG = HMAC (hash function) based
- Hash-DRBG = hash function based
- Dual-EC-DRBG = elliptic curve based

*Other than Hash-DRBG, same algorithms in X9.82*
Dual EC DRBG

- Start from $P$ and $Q$ (system parameters)
- To generate an output:
  - $z = sQ$ (convert to integer)
  - Output $z$, with some bits truncated
  - $s = sP$ (Convert to integer)
Dual EC DRBG has two parameters, P and Q.

• Can be public and shared with all users

  …but that isn’t necessary.

• Where do these come from?

  • Provided in standard

  • Ultimately from designers of Dual EC DRBG at NSA.

  • What if you don’t trust the people who generated P and Q?
Tusting P and Q

• If P and Q are randomly generated, Dual EC secure.
  
• *P and Q can be generated to insert a backdoor.*

• Issue was first raised in an X9 meeting

• Later, issue was described at Crypto 2007 rump session.
The Possible Trapdoor

0. Attacker knows \( a \) such that \( aQ = P \)

\[ s_1 \xrightarrow{a} Q \xrightarrow{\phi} \text{Truncate} \xrightarrow{\text{Drbg output}} \]

1. Invert these steps \( 2^{16} \) work.

2. Use knowledge of \( a \) to derive next seed value

- Attacker generates \( P \) and \( Q \) with trapdoor!

- \( z = sQ \) (convert to integer)
- Output \( z \), with some bits truncated
  - Guess truncated bits to get back to \( sQ \).
- \( s = sP \) (Convert to integer)
  - Use trapdoor: new \( s = asQ \)
Discussed in X9 Meeting

• Didn’t seem like a real threat

• Obvious choice would have been to generate P and Q in a verifiably random way, make those the new system parameters.
  
  • At least one vendor had implemented with original P,Q.

• Instead, we allowed implementers to generate their own P and Q in a verifiably random way.
  
  • As far as we know, nobody actually did this..
Snowden Disclosures

• News stories came out strongly suggesting that Dual EC had a trapdoor inserted by NSA

• This put the previous discussions in an entirely new light.

• We responded by:

  • Issuing an ITL bulletin telling everyone to stop using Dual EC DRBG until further notice.
  
  • Putting all three 800-90 documents up for public comment
Future of 800-90A

• Our current plan is to remove Dual EC DRBG
  • Its performance is pretty slow
  • Many vendors already have scrambled to remove or disable it in their products.
• Phase-out period
Questions / Lessons Learned

• Developing standards in an adversarial world?

• Transitive trust?