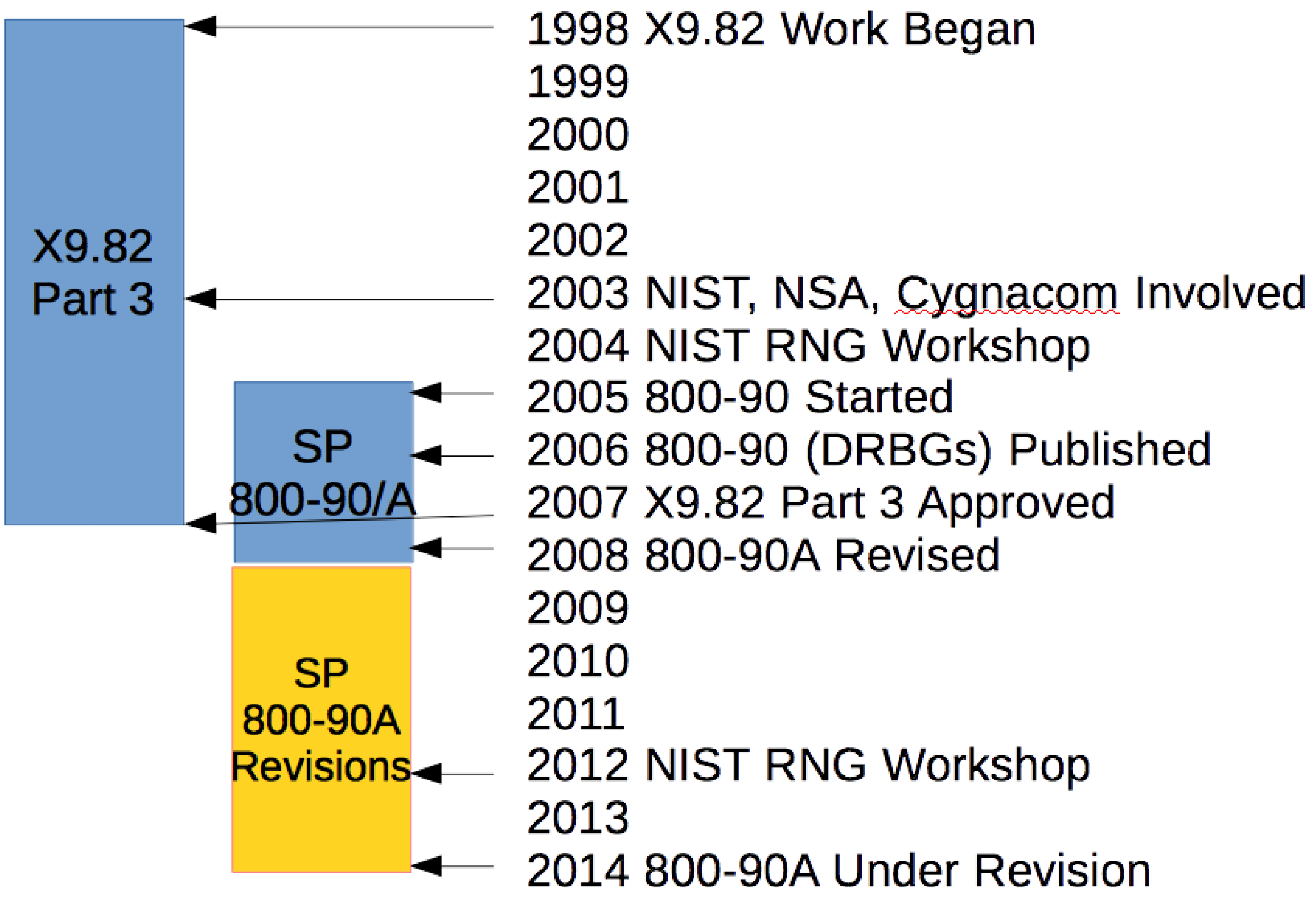
Dual EC in X9.82 and SP 800-90



1998 X9.82 Work Began
1999
2000
2001
2002
2003 NIST, NSA, Cygnacom Involved
2004 NIST RNG Workshop
2005 800-90 Started
2006 800-90 (DRBGs) Published
2007 X9.82 Part 3 Approved
2008 800-90A Revised
2009
2010
2011
2012 NIST RNG Workshop
2013
2014 800-90A Under Revision

John Kelsey, NIST, May 2014

Overview

- What Happened?
- Technical Background
- X9.82 and SP 800-90 Process
- Dual EC DRBG Postmortem
- What Went Wrong?
- Why

Summary: What’s the Issue

- NIST and NSA coauthored a set of standards on cryptographic random number generation.
- NSA provided Dual EC DRBG.
- *Many reasons* we should have rejected or modified Dual EC DRBG
  - Instead, we left it in.
- News stories based on Snowden disclosures came out.
  - Suggest that Dual EC DRBG has an *intentional backdoor*.

A Few Technical Details

- *What’s a DRBG?*
- *What’s Dual EC DRBG?*
- *What does it mean to say a DRBG is biased?*
- *What does it mean to say a DRBG has a backdoor?*

Random Numbers and Cryptography

- Random numbers are critical for cryptography
- Every time you use TLS (by going to a secure webpage), you’re using random numbers.
- NIST and NSA worked together on two different standards for cryptographic random number generation
  - X9.82 (1998-2007)
  - SP 800-90 (2005-Present)

DRBGs
Deterministic Random Bit Generators

- Cryptographic random number generators come in two parts:
  - Unpredictable processes used to generate a *seed*
  - Algorithm to generate *random bits* from seed.
- DRBG = Deterministic Random Bit Generator
  - *Algorithm* for generating random-looking bits.
  - Specified in X9.82 Part 3 and SP 800-90A.
  - Should produce outputs nobody can distinguish from random bits.
- Other parts of these standards describe how to produce the seed.

What Can Go Wrong?
Bias and Backdoors

- ***Bias***
  - DRBG outputs should be indistinguishable from random bits.
  - Slight deviation from random behavior = **bias**.
  - Like loaded dice or a weighted coin.
- ***Backdoor/Trapdoor***
  - Secret knowledge that lets you predict outputs = **backdoor**.
  - Without knowing seed, nobody should be able to predict DRBG outputs!

Dual EC DRBG
Dual *Elliptic Curve* DRBG

- DRBG provided by NSA
- Security based on number theory problem
- Defined for three curves (three security levels)
- For each curve, some public parameters (P,Q) defined as part of DRBG definition.
- Other DRBGs in standard:
  - Hash_DRBG (NSA design, modified by NIST)
    - SP 800-90/90A only
  - HMAC_DRBG, CTR_DRBG (NIST designs)

Dual EC DRBG: P and Q

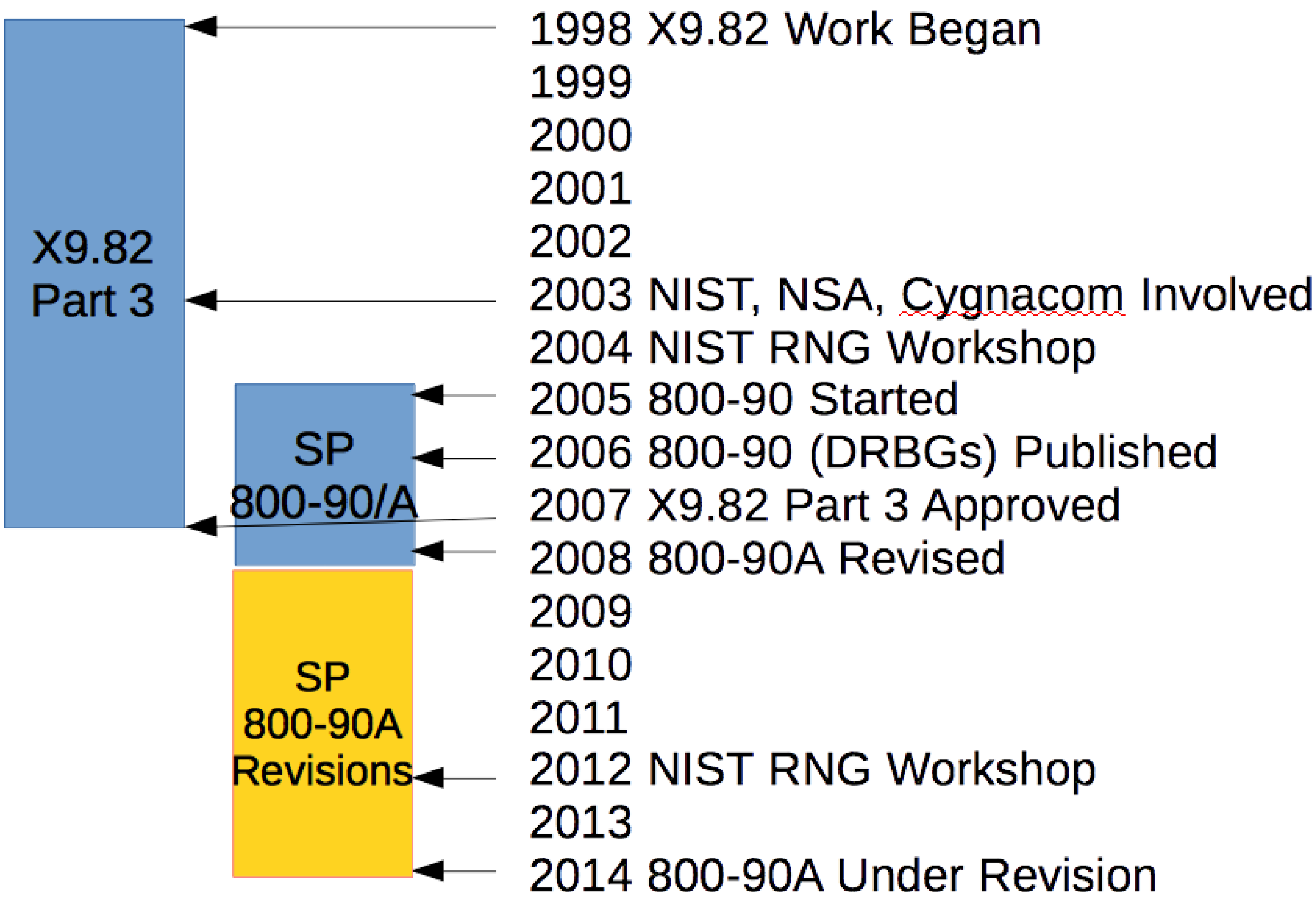
- Dual EC DRBG’s definition requires choosing some parameters: (P,Q)
  - Elliptic curve points.
- It is possible to choose (P,Q) so that you know a backdoor for the DRBG.
  - NSA is alleged to have done this.
- It is also possible to choose (P,Q) so that you can prove you don’t know a backdoor.
  - We have a mechanism to do this in our standards, but it seems never to have been used.

Issues with Dual EC DRBG

- **Bias** – Dual EC DRBG has a slight statistical bias
  - Theoretical weakness when DRBG is used to generate keys.
  - But it violates our requirements for DRBGs
- **Possible Backdoor** – (P,Q) may have been generated to allow NSA to know a backdoor.
  - This would be a practical (and very important) weakness

*Dual EC DRBG **should not** have been included in X9.82 or SP 800-90 in current form.*

History of the Standards



X9.82 Part 3

SP 800-90/A

SP 800-90A Revisions

1998 X9.82 Work Began
1999
2000
2001
2002
2003 NIST, NSA, Cygnacom Involved
2004 NIST RNG Workshop
2005 800-90 Started
2006 800-90 (DRBGs) Published
2007 X9.82 Part 3 Approved
2008 800-90A Revised
2009
2010
2011
2012 NIST RNG Workshop
2013
2014 800-90A Under Revision

Early Development (1998-2003)

- Why the project?
  - Crypto needs random numbers; no good guidance available
- Why X9F1?
  - Financial services industry was an early adopter of strong crypto.
  - Variety of members with crypto technical abilities
- Who to do the work?
  - NIST provided an editor
  - Expected technical participation from X9 members

Early Development (contd.)

- Early drafts included
  - RNGs from FIPS 186-2 and X9.31
  - Statistical tests from SP 800-22
  - Notions of DRBGs and NRBGs
  - Kind-of a grab bag of different algorithms and definitions.
- 2003: Started making progress
  - I joined NIST, got assigned to project.
  - Cygnacom and NSA agreed to help

Getting Outside Review

- NIST believed our work on the standard needed more outside review.
- Workshop at NIST in 2004
  - Obtained permission from X9 to hold the workshop
  - Obtained permission to post the drafts for a limited time (without charging)
- Expert review
  - Identified several experts
  - Very few responses

X9 Issues

- Doing standard in X9 made it harder to get feedback
  - Copies not available for review except by paying
  - Few universities are X9 members, so academic cryptographers usually can’t be involved
  - Limited number of members with the right background (e.g., RNGs)
    - Public review not very public
- Lengthy development and approval process: ballot through X9F1, X9F, X9 (≥ 1 year)
- X9 doesn’t deal with FIPS 140 validation issues

Moving Work to SP 800-90

- **SP 800-90 (initial draft in 2005)**
  - Mostly just DRBGs from X9.82 Part 3
  - Intended to get wider review: (beyond X9F1, without buying the document)
  - Can address validation issues
  - Faster approval and revision process as an SP than as an X9 standard
  - Contribution primarily by govt. employees
  - Worked on both SP 800-90 and X9.82 for a few years.
- Also X9.82, Part 3 (DRBGs) became an ISO Standard in 2007

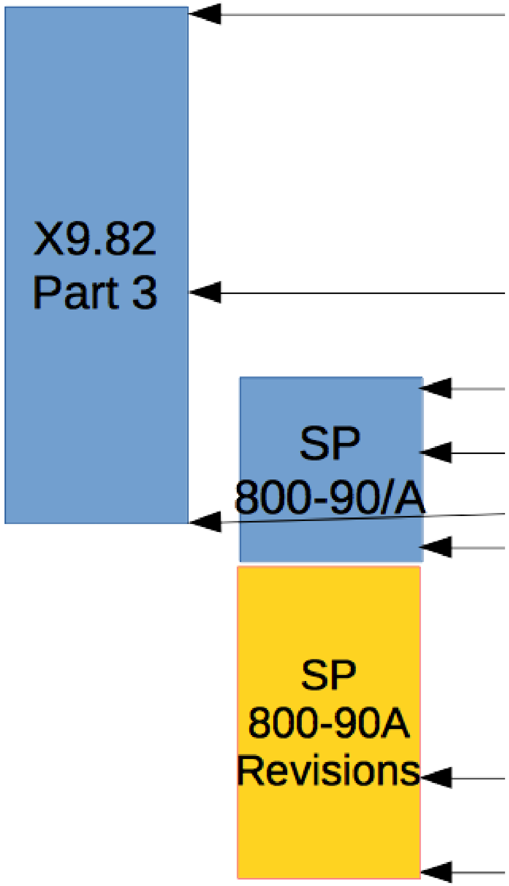
Completions

- SP 800-90/90A
  - Completed in 2006 (Mostly just DRBGs)
  - Revised ~2008 (SP 800-90A)
  - Revised in 2012
  - Revision provided for comment in 2014
  - Work ongoing on 800-90B and 800-90C
- X9.82, Part 3
  - Final ballot in 2007
  - Final version in 2007

Public Engagement

- X9.82 went through the normal process for an X9 document.
  - Discussed at many X9F1 meetings
  - Formal comments from members
- SP 800-90 also went through our normal process
  - Public comment periods
- Two public workshops: 2004, 2012
- Talk at RSA: 2006

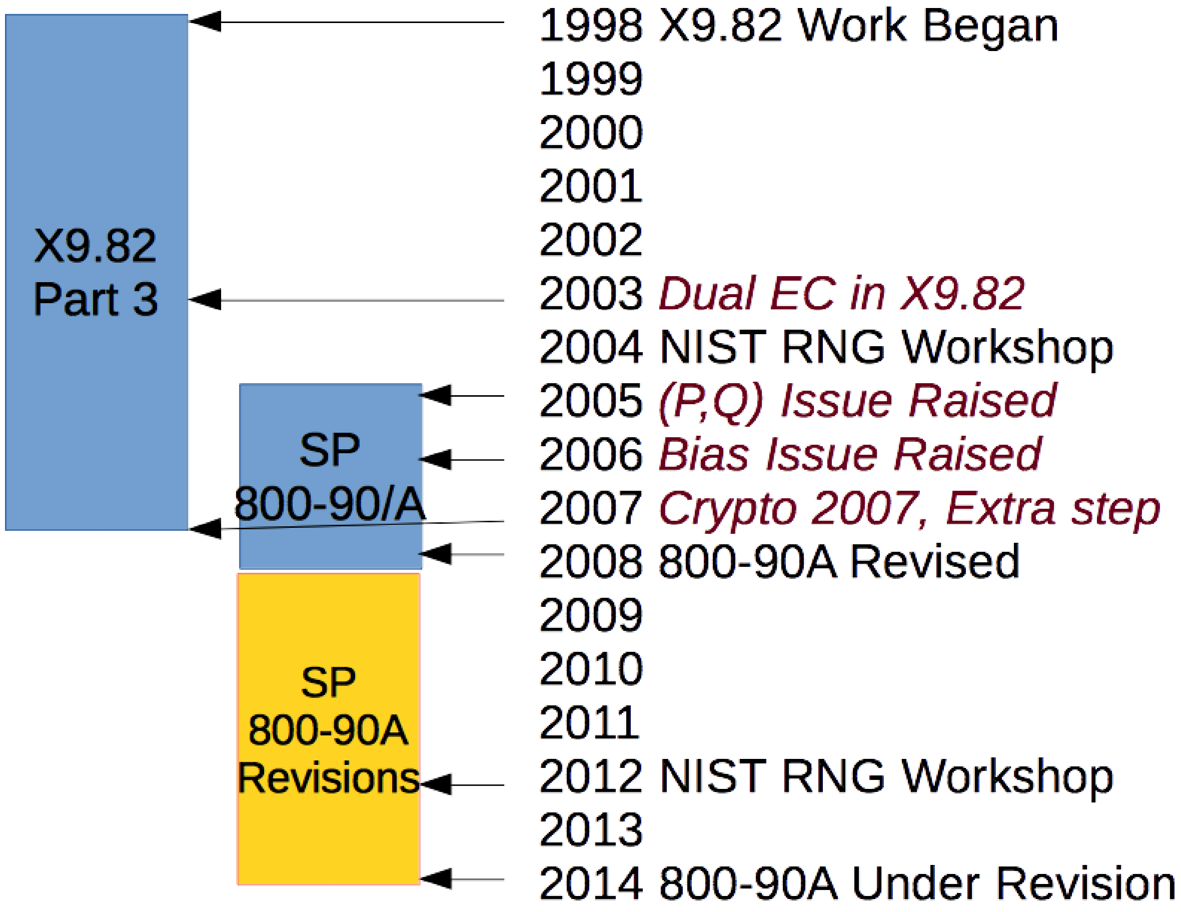
History Recap



1998 X9.82 Work Began
1999
2000
2001
2002
2003 NIST, NSA, Cygnacom Involved
2004 NIST RNG Workshop
2005 800-90 Started
2006 800-90 (DRBGs) Published
2007 X9.82 Part 3 Approved
2008 800-90A Revised
2009
2010
2011
2012 NIST RNG Workshop
2013
2014 800-90A Under Revision

NOTE: *The two standards had the same people working on them.*

Dual EC History
What Happened?



X9.82 Part 3

SP 800-90/A

SP 800-90A Revisions

1998 X9.82 Work Began
1999
2000
2001
2002
2003 *Dual EC in X9.82*
2004 NIST RNG Workshop
2005 *(P,Q) Issue Raised*
2006 *Bias Issue Raised*
2007 *Crypto 2007, Extra step*
2008 800-90A Revised
2009
2010
2011
2012 NIST RNG Workshop
2013
2014 800-90A Under Revision

Dual EC: What Went Wrong?

- Dual EC DRBG had security issues that should have kept it out of X9.82 and SP 800-90.
  - Bias (from not throwing away enough bits)
  - Possible backdoor in (P,Q)
- Both issues identified during standards development process.
- Changes made to the standards failed to adequately address them.

Reconstructing What Happened

- We’ve spent a lot of time trying to figure out what happened.
- Two standards spread out over more than a decade.
- Sources:
  - Our memories
  - X9 documents
  - Emails
  - Public comments
  - Intermediate versions of documents

10/27/04 Exchange with Cygnacom

- This was right after the RNG workshop
- I asked where (P, Q) came from.
- Said it could be generated randomly, but that NSA had told not to talk about it.
- I didn’t catch why this was significant then.

Oct 05: Microsoft Comment

- In 2005, we discussed the possibility of a trapdoor in the Dual EC DRBG in an X9 meeting. Niels Ferguson made a formal comment on the document in X9 pointing this out.
- The response (from NSA):
  - NSA generated (P, Q) in a secure, classified way.
  - DRBG was originally generated for national security community.
  - Wanted to protect their existing investment in it and allow devices using it to get FIPS validated.
  - It would be reasonable to allow other users to generate their own (P, Q).
- We basically agreed with this response.

How we responded

- We added an optional mechanism for generating a provably random (P, Q).
- Required original (P, Q) for validation.
- Recommended original (P, Q).
- Never explained why someone might generate their own (P, Q) in standard.

Why didn’t we respond more effectively?

- Dual EC DRBG is extremely slow, and seemed unlikely to see much use.
  - So putting a trapdoor in seemed kind-of pointless.
- NSA claimed to have existing customers who were using the DRBG, and wanted to be able to get FIPS validation.
  - We expected these customers would be the only users.
- We didn’t believe a backdoor in this algorithm was likely.

We answered the wrong question

- When this came up, we considered the question:
  - *Do we think there is a trapdoor into Dual EC DRBG?*
- We should have asked:
  - *Should we include an algorithm in our standards that **could** have a trapdoor?*

Feb 06: My RSA Talk

- I gave a talk at RSA in Feb 2006 about NIST's X9 standards, including X9.82.
- Kristian Gjosteen sent me a draft paper about a month later on **bias** in Dual EC DRBG
  - Need to throw away more bits to get rid of bias.
- I forwarded this to rest of editing committee.
  - Can't find any responses, but we did discuss this issue in X9.

NOTE: Matt Green has pointed out that addressing this bias would have made the trapdoor much harder to use.

Crypto 07 Rump Session Talk

On The Possibility of a Back Door in the NIST SP800-90 Dual EC Prng, by Niels Ferguson and Dan Shumow

- Explained the possible trapdoor.
- Explained the right way to fix it.
- Everything in the presentation had been discussed before in X9.
  - Called the community's attention to the issue.

Crypto 2007: My Email

- I sent an email to the editing committee immediately after Crypto 2007:
  - Statistical problems are probably a culture-clash between NSA's attack-oriented worldview and crypto community's proof-oriented worldview.
  - Potential trapdoor is a major issue and we should revisit how this is dealt with.
  - Apologized for not realizing before how big an issue this would be.
- I couldn't find any response to my email or any follow-up—this was one of our last chances to get this right.

Bruce Schneier’s Column

- Bruce Schneier pointed out that this looked really bad in a column in Wired in late 2007, and NIST wrote him a response.
- This is an email NIST sent back to Bruce at that time.

Nov 2007 Email to Bruce Schneier

In your November 14, 2007 Wired commentary (http://www.wired.com/politics/security/commentary/securitymatters/2007/11/securitymatters_1115), you suggested that the Dual_EC_DRBG random number generator published in NIST Special Publication 800-90 has a property "that can only be described as a back door." We have no evidence that anyone has, or will ever have, the "secret numbers" for the back door that were hypothesized by mathematicians Dan Shumow and Neils Ferguson, that would provide advance information on the random numbers generated by the algorithm. For this reason, we are not withdrawing the algorithm at this time. NIST Special Publication 800-90, which includes a method for randomly generating points if there is a concern about a back door, underwent a rigorous review process that included a public comment period before it was published,. All NIST algorithms, including the Dual_EC_DRBG, undergo continual review throughout their lifetime. If successful attacks are found on an algorithm, the algorithm is withdrawn.

X9F1 Chair, Late 2007

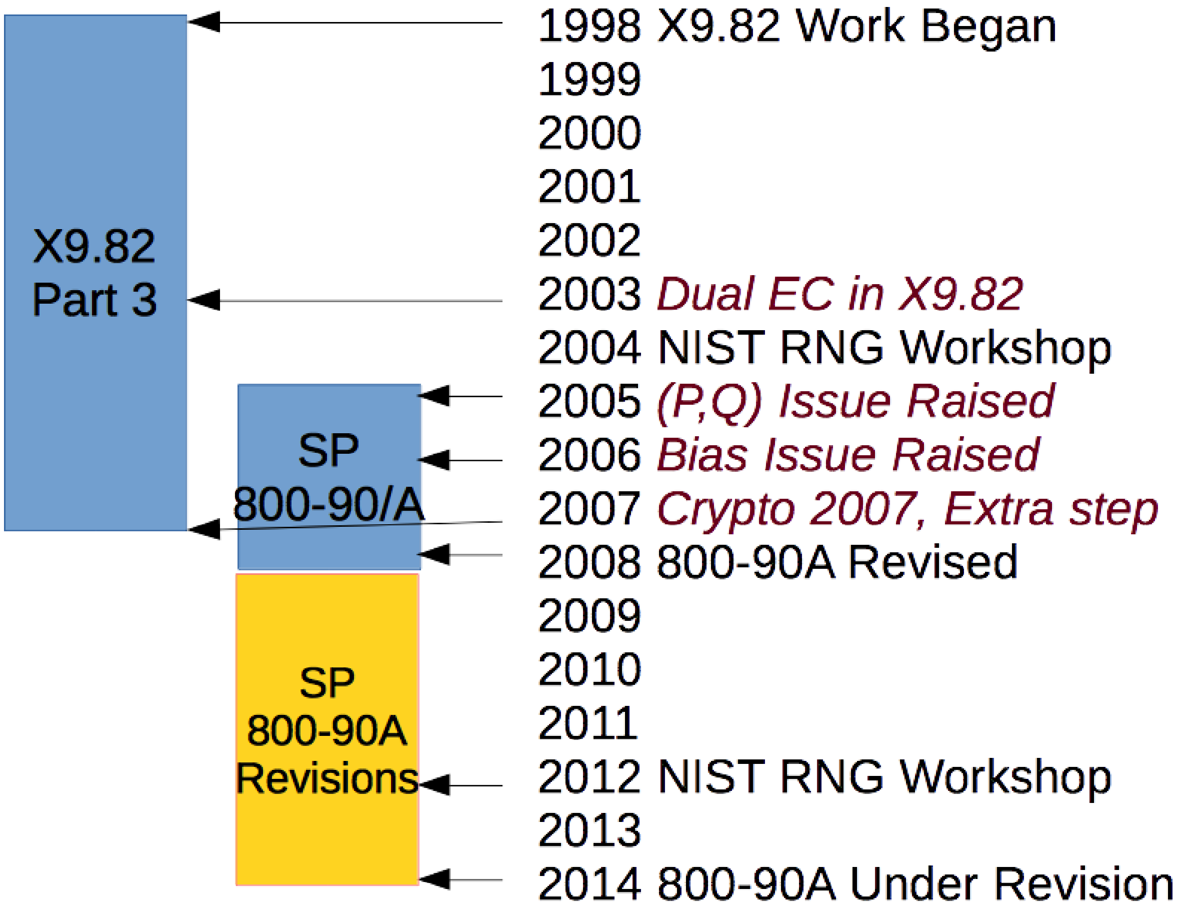
- Miles Smid, the X9F1 chair, emailed to ask Elaine whether we should remove or alter Dual EC in light of the Crypto rump session talk.
  - This would have involved reopening a standard that was already done.
- The result was a discussion on the next X9F1 teleconference and the next X9F1 meeting

Jan 08 X9F1 Meeting in Ft Lauderdale

Note summarizing decision of X9F1 group

- Because of Crypto 2007 Rump Session, they revisited the decision to include Dual EC with the “legacy” elliptic curve points.
- In the discussion, several points were made:
  - This concern had been raised in the X9F1 group
  - (P,Q) generated to avoid this issue.
  - Vendors could generate their own (P,Q).
  - Known constituency that was using the DRBG with the old (P,Q) already.
  - NIST was planning to validate the DRBG using the original (NSA generated) (P,Q).
  - X9.82 had been balloted through X9 with no objections to the current wording.
- Their conclusion was that Dual EC DRBG should stay in X9.82 as specified.

Problems With Dual EC DRBG in X9.82 and SP 800-90



X9.82 Part 3

SP 800-90/A

SP 800-90A Revisions

1998 X9.82 Work Began
1999
2000
2001
2002
2003 *Dual EC in X9.82*
2004 NIST RNG Workshop
2005 *(P,Q) Issue Raised*
2006 *Bias Issue Raised*
2007 *Crypto 2007, Extra step*
2008 800-90A Revised
2009
2010
2011
2012 NIST RNG Workshop
2013
2014 800-90A Under Revision

Dual EC Issue #1: Statistical Bias

- **Issue:** Dual EC DRBG outputs have a small statistical **bias**
  - There is a step in the DRBG where some bits are thrown away (“truncated”)
  - Too few bits are truncated, leaving a detectable bias in the output statistics
  - Note: throwing away more bits would make trapdoor harder to use!
- **History:**
  - This was discussed during the X9 process
  - Papers and comments by many people in 2006-2007

Dual EC Issue #1: Statistical Bias (2)

- **What We Did:**
  - We left Dual EC DRBG in with too little truncation and a statistical bias.
- **What We Should Have Done:**
  - Either removed Dual EC DRBG, or changed it to throw away more bits.
- **Why:**
  - Some internal discussion in X9 that said this was just a theoretical issue.
  - Some efficiency concerns raised (but already slow!)
  - It was NSA's algorithm, we largely let them respond to comments on it.

Dual EC Issue #2: Potential Trapdoor

- **Issue:** (P,Q) of Dual EC DRBG could have been generated to insert a trapdoor.
  - Ideally should have been generated in a verifiably random way.
  - Different (P,Q) needed for each curve supported.
- **History:**
  - (P,Q) present in X9.82 for three curves from 2003 on.
  - Issue raised in 2005 and again a few times through 2008

Dual EC Issue #2: Potential Trapdoor(2)

- **What We Should Have Done:**
  - Generate a new (P,Q) in a verifiable way for each included curve.
  - Either allow the original (P,Q) for limited time, or remove the original (P,Q) from the document.
- **What We Did:**
  - Added an optional way for users of standard to generate their own verifiably secure (P,Q).
  - Did not explain why anywhere in the document.

Dual EC Issue #2: Potential Trapdoor(3)

- **Why?:**
- NSA claimed to have customers using Dual EC DRBG with the original points.
  - Wanted to be able to obtain FIPS validation
  - Requiring new points would render existing implementations unable to get validated
- Expected little use outside of DoD and national security community
- A backdoor in Dual EC DRBG seemed very unlikely to us.
  - Answered the wrong question.

Dual EC Issue #3: (P,Q) Generation

- **Issue:** Our solution to the potential trapdoor was to let users generate their own (P,Q) parameter.
- As far as we know nobody used it.
  - Not even organizations with a lot of cryptographic expertise.

Dual EC Issue #3: (P,Q) Generation(2)

- **What We Did:**
  - Original (P,Q) was default, and was required for FIPS validation.
  - Had to go to another unrelated standard (X9) to get full algorithm for generating your own (P,Q)
  - No explanation of why to generate your own (P,Q)
- **What We Should Have Done:**
  - Give a default (P,Q) generated in a verifiable way,.
  - Specify method used to generate (P,Q) with enough detail to allow readers to verify points.
  - Not require original (P,Q)—at most leave it as an option for already fielded implementations.

Why did this happen?

- NIST Relationship with NSA
- Insularity
- Standards Group Dynamics
- Recordkeeping and Continuity

Issue #1
NIST Relationship with NSA

- Relied on NSA for expertise we lacked on elliptic curve cryptography
- Trusted NSA input where we would have been much more skeptical of anyone else
  - We trust their technical expertise.
  - …and we've gotten very useful technical feedback from them on many occasions.

Issue #2
Insularity

- Ignored or minimized feedback from outside editing committee that could have saved us
- Trusted other members of editing committee
- NSA owned Dual EC, we owned symmetric DRBGs

Issue #3
Standards Group Dynamics

- Many reasons why Dual EC DRBG should not have been in standard….
  - Performance, Bias, Potential Trapdoor
- …but it had a champion on X9.82 editing committee
  - Common way for weak algorithms to get into standards.
- Wanted to allow existing implementations to comply with future standard.
  - Not a good idea when it requires weakening standard!

Issue #4: **Recordkeeping and Continuity**

- Long-running project with lots of interruptions.
  - Lots of opportunities for issues to get dropped or quietly forgotten.
  - Saw many examples in emails.
  - Contributed to incomplete reviews of documents and forgotten issues
- Recordkeeping for the project was informal
  - Largely in email or current drafts of documents

Acknowledgements

- Lots of cryptographers outside of NIST have commented on these two standards, before and after the news stories based on Snowden’s leaks, and I’ve drawn heavily on their comments and analyses.

Dan Bernstein, Dan Brown, Niels Ferguson, Kristian Gjosteen, Matt Green, Tanja Lange, Bruce Schneier, Berry Schoenmakers, Dan Shumow, Andrey Sidorenko

With apologies to anyone I’ve left out.