Development of FIPS 186

Digital Signatures
(and Elliptic Curves)

Dustin Moody
Digital Signatures

• 1978 – RSA signature algorithm
• 1982 – Federal Register Notice soliciting digital signature algorithms
  – RSA paper only suggested algorithm
  – Patent issues
• 1989 – NIST/NSA Technical Working Group (TWG)
  – Issue #1: public key crypto, including signatures,
The Digital Signature Standard – FIPS 186

- NIST considered several algorithms
  - RSA, already deployed in industry
  - Other algorithms from academic literature
  - NSA-designed algorithms
- Although favored by industry, RSA was not selected
  - Issues over exportability and patents
- FIPS 186 standardized the NSA-designed Digital Signature Algorithm (DSA) in 1991
Public Concerns

- DSA selection process not public
- Not enough cryptanalysis
- Parameter sizes
  - 512 bit modulus, 160 bit subgroup
- Performance concerns

- 1992 NIST report on comments and adjudications
  - Increased parameter sizes allowed
FIPS 186-1, FIPS 186-2

• Background
  – DSA not widely adopted
  – Interest in RSA and elliptic curve DSA schemes

• 1997 – NIST requested comments on adding new signature schemes to FIPS 186
  – Overwhelmingly positive response for both schemes

• NIST worked with ASC X9
  – X9.31 for RSA and X9.62 for ECDSA

• 1998 – FIPS 186-1, approves X9.31

• 2000 – FIPS 186-2, approves X9.62
Continued Development

• Before FIPS 186-1, industry implemented RSA signatures following PKCS#1 standard
  – When FIPS 186-1 was developed, NIST assumed the public would switch to ANS X9.31, but this didn’t happen
  – NIST moved to allow PKCS#1 version of RSA signatures (FIPS 186-2)
• 2009 – FIPS 186-3 increased key sized for DSA and added additional requirements for ECDSA and RSA
  – NSA collaborated on FIPS 186-3
• 2013 – FIPS 186-4 corrected errors
NIST Curves

• 1985 – Elliptic Curve Cryptography proposed

• 2000 – NIST standardized the Elliptic Curve DSA in FIPS 186-2
  – NIST recommended 15 elliptic curves of varying security levels, called *NIST curves*
  – The NIST curves are also used for key agreement (SP 800-56A)

• 2013 – some concerns about NIST curves
Curve Concerns

- **Efficiency**
  - NIST curves chosen to be efficient
  - New curves with more efficient implementations have since been found

- **Security**
  - The addition operation for the NIST curves has special cases which can allow for side-channel attacks
  - New curves have been found which avoid this pitfall

- Do the NIST curves have hidden weakness?
Types of Curves

• Two different kinds of curves:
  – *Pseudo-random curves* - coefficients are generated from the output of a seeded cryptographic hash
  – *Special curves* - coefficients and underlying field have been selected to optimize efficiency

• Concern expressed over provenance of the parameters of pseudo-random curves
  – Where do NIST curve coefficients come from?
Pseudorandom Curve Generation

• Each pseudo-random curve has a parameter $b$
  – The parameter $b$ is the output of a one-way function generated from a seed
    • i.e. $H(\text{seed})=b$

• Given the seed, it is easily verified that $b$ was generated by this method

• Ensures the elliptic curve cannot be predetermined
Curve Selection

- In general, a pseudorandom curve was chosen by:
  1) Select a seed and generate the elliptic curve
  2) Check if curve is secure against known attacks. If vulnerable, go to step 1 and repeat
     Note: Very likely need to choose many seeds

- The curves were generated by the NSA
- The seeds and curve parameters are published
Security of NIST Curves

• Assuming that SHA-1 cannot be inverted, generation process provides assurance NIST curves not intentionally constructed with hidden weaknesses

• In particular, the NIST curves do NOT belong to any known class of elliptic curves with weak security properties
  – No sufficiently large classes of weak curves are known

• There are NO known attacks of cryptographic significance which lessen the claimed security levels of the NIST curves