The Future of Security Standards

John Kelsey, NIST, Dec 2016

Overview

- My background
- Security standards are different
- How to fail
- Designing better security standards
- Building public confidence in standards
- Wrapup

My background

- NIST cryptographer
- Worked on several NIST standards
  - SP800-90A, B, C
  - X9.82 Parts 2 and 3
  - VVSG 1.0, 2.0
  - FIPS202 (SHA3)
  - SP 800-185 (cSHAKE)
- And one more, before coming to NIST
  - syslog-sign

Security standards are like other standards…

- Usually go through consensus-based standards processes
- Same organizations as non-security or non-security-relevant standards

ISO	IETF
X9	IEEE
NIST	etc.

- Similar issues with intellectual property, limited review, slow progress, etc.

…but also NOT like other standards

- Ways security standards can fail that don’t apply to other standards
  - Failures are invisible
  - More options, backward compatibility, etc. can weaken standard
  - Consensus standards processes don’t always play well with security
  - Adversarial participants in the process
- Problems with security standards that others don’t face
  - Security adds cost and hassle
  - Often imposed on users instead of demanded by users.
  - Public confidence in standards is critical
  - Powerful entities often want weaker security

Why are security standards hard?

Security failures are usually invisible

- Security failures are invisible
  - Product works fine
  - Interoperates with other products fine
  - Only problem: all your secrets are leaking to the guy in the van outside.
- Failures become visible all at once
  - High-profile attack or widely publicized academic attack...
  - ...then they’re repaired in crisis mode

- Example: 802.11 WEP, pretty much any other security failure

Security adds cost and hassle

- Most standards are responding to something users want
  - Wireless internet
  - Standard port types that work across vendors
  - Showing video on the web
- Security standards are often *imposed* on users or organizations
  - Minimum password requirements
  - PCI standards for companies handling credit card data
  - FISMA standards imposed on government agencies
  - *Result: pushback on requirements, doing the minimum required*

- Example: VVPAT requirements in VVSG 2.0

More options = worse security (usually)

- Common to have one or two main options everyone actually uses, but then lots of options in the standard
  - “Everyone’s a winner”
  - Every company has their own stuff that’s in the standard
- This is terrible for security
  - The more options, the more likely one is weak
  - If I can force you into supporting the weak option, I get to attack you—even though the usual stuff everyone does is secure.
- Example: Heartbleed in OpenSSL
  - Heartbeat was an almost-never-used option
  - Implementation error turned it into a huge security hole in millions of computers

Backward compatibility hurts security

- Common to update standards and leave support for anything in previous versions...
- ...even when the update is intended to improve security.
- This is usually fine for functionality—the old stuff just doesn't include the new features.
- It's a disaster for security
  - Downgrade attacks!
  - Some people choose the cheapest (weak, old) option
- Examples: TLS attacks on export-controlled ciphersuites (Logjam)

Backfilling to existing practice

- In new standards, common to try to backfill the new standard to fit what people are doing in the field
  - Justified by cost of changeover
  - Companies with stuff in the field often on standards committee
- Problem: New stuff gets built with same bad security model as old.
- This makes it very difficult to improve security with new standard.

- Example: 3DES and SHA1 in TLS and new NIST standards
- Example: SWEET32 (attacking use of 3DES because of its small block)

Algorithm agility—the wrong way

- Standards groups often justify having lots of different crypto options for the sake of *algorithm agility*.
  - ”Everyone is a winner”
- Common situation: One mandatory-to-implement algorithm, ten seldom-used, poorly-analyzed one.
  - If the mandatory algorithm is broken, **you can’t turn it off**!
- Real algorithm agility means *you can turn any algorithm off without breaking things.*
- Ideal situation: Two strong, mandatory-to-implement algorithms.
  - **If one is broken, you CAN turn it off!**

Standardization process vs security

Closed standards process

- Many standards processes don’t allow outsiders to comment
- This excludes many people who might give useful reviews
  - Academics
  - Grad students
  - New researchers wanting to make a name
- Commonly standards cost a lot of money!
- Result: only people on standards committee can see documents
  - Attacks don’t get found for many years

Procedural issues with standards

- Design by committee
  - Usually not a great idea
  - Really really bad for security and especially crypto
  - Need one coherent security model in mind
- Long process with many participants
  - Opportunity for bad things to get slipped in or good things to get broken
  - Editing committee may change over time—easy for important knowledge to get lost.
- Insularity
  - Ignoring external feedback
  - The “Not Invented Here” syndrome

Adversarial participants

- Normal standards processes have some adversarial elements
  - Competitors trying to spike each other's stuff
  - People trying to slip IP into standard so they can collect royalties
- Security standards have much uglier potential adversaries, who may want to...
  - generically weaken security
  - delay use of security that would make their jobs harder
  - install a specific backdoor for their own access

Who might want to weaken a security standard?

- Intelligence agencies (foreign and domestic)
- Law enforcement agencies (many different countries)
- Companies that use exploits in their business
- Companies whose business model is threatened by security
- Even criminals

- Example: Dual EC DRBG in SP 800-90 and X9.82

What’s needed for future security standards?

1. Getting the technical details right
2. Gaining public confidence

Getting the right expertise

- Very important to get the science/math/technology right in the design
- This requires expertise which isn't always available in standards committee
- ...also requires *time* from high-value people.
  - Example: SHA3 competition, CAESAR competition
- Making a good selection from outside designs requires expertise
- Building something new requires even ***more*** expertise

Getting expert feedback on technical details

- Standards often involve technical details from a variety of fields
- Example: SP 800-90
  - Symmetric crypto
  - Asymmetric crypto (not anymore)
  - Statistics
  - Information theory
- Important to get feedback from experts in all those fields
- Not so easy to get experts to read a whole standard!
- Alternatives
  - Summaries of narrow technical issues that need review
  - Academic papers and presentations

How meaningful are your review comments?

- Any security standard needs review by people with the right expertise
  - NOT just the designers!
  - Limited expertise available in standards organization/committee
- Solutions: Public comment period, internal comments by other participants in standards group
- **Question: How do you know how much depth of review you've gotten?**
- Lots of nitpicky comments << a few careful analyses
- Is there someone whose job is to do a thorough review?
  - Do they know it's their job?
  - Do they have enough time and resources to do it?

How will it be used? [Implementations]

- How will this standard be used in practice?
- What errors will implementers make?
- What errors will users of standard make?
- Error-prone?
  - How hard is it to mess up implementation or use?
- Misuse resilience
  - How much security do they retain if they mess something up?

- Examples: DSA and random numbers, GCM and nonce reuse

How will it be used? (2) [Enforcement]

- How will standard be enforced and applied?
- Testing labs?
  - Can labs test all the critical security requirements?
  - Do they have expertise and incentives to do so?
- Auditors?
  - How will auditors know whether your standard is being followed or implemented correctly?

- Example: SP 800-90B, GCM

What’s needed for future security standards?

1. Getting the technical details right
2. **Gaining public confidence**

Confidence is critical for success

- Failures of security are invisible...
- ...so conspiracy theories and FUD (Fear, Uncertainty, Doubt) can run wild.
- Lack of confidence means security standards aren't adopted widely—leads to
  - Balkanization (everyone does their own thing)
  - Snake-oil (don't trust the standard, trust my million-bit-key cryptosystem)
  - No security (people don't use anything because they're scared)
- Example: Use of Intel RNG

Where did your constants come from?

- Lots of crypto standards have some constants
  - S-boxes
  - Bit permutations
  - Matrices
  - Initial values
- Need to be transparent about where these came from
- ..and need to show they weren't chosen to weaken standard.
- ***Rigidity*** means choosing constants in such a way that designer had few (or maybe only one) plausible choices for them.
- Example: NIST elliptic curves

Participation by the Community

- More community participation -> more trust

- Competitions are great for this
  - Demanding to run
  - Probably only so many can be going at a time
- For any standard, public engagement is a must
  - Talks/Papers
  - Public comment periods
  - Workshops
  - Methods to enable feedback

Transparency of Process

- Standards are more trustworthy if the process used to generate them is transparent.
- Full disclosure of who worked on standard and any conflicts
- Public participation
  - Workshops, public comment periods
- Transparent handling of public comments
  - Publish comments and responses
- Make reasoning for decisions as open as possible

Wrapup

Wrapup

- Standards are hard, security standards are ***especially*** hard.
  - Many normal parts of standards process play badly with security.
  - Adversarial in ways other standards aren't.
- Security standards require specialized expertise
  - Hard to get good reviews
  - Hard to even know how much depth reviewers considered
- Both design and process of standard need to get technology right AND encourage public confidence
  - Rigidity, simplicity of design, transparency of process all important

Questions?