CSci 427IW Development of Secure Software Systems Day 20: 'S' protocols and crypto failures

Stephen McCamant University of Minnesota, Computer Science & Engineering

Outline

SSH

SSL/TLS

More causes of crypto failure

DNSSEC

Short history of SSH

- 5 Started out as freeware by Tatu Ylönen in 1995
- Original version commercialized
- Fully open-source OpenSSH from OpenBSD
- Protocol redesigned and standardized for "SSH 2"



Newer crypto vulnerabilities

IV chaining: IV based on last message ciphertext

- Allows chosen plaintext attacks
- Better proposal: separate, random IVs
- 🖲 Some tricky attacks still left
 - Send byte-by-byte, watch for errors
 - Of arguable exploitability due to abort
- Now migrating to CTR mode

SSH over SSH

- SSH to machine 1, from there to machine 2 Common in these days of NATs
- Better: have machine 1 forward an encrypted connection
- 1. No need to trust 1 for secrecy
- 2. Timing attacks against password typing

SSH (non-)PKI

When you connect to a host freshly, a mild note
When the host key has changed, a large warning

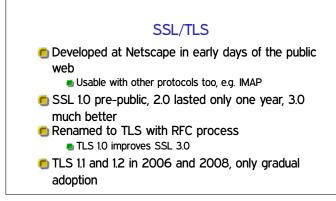
Outline

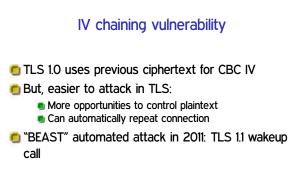
SSH

SSL/TLS

More causes of crypto failure

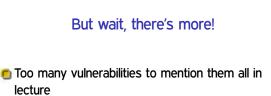
DNSSEC





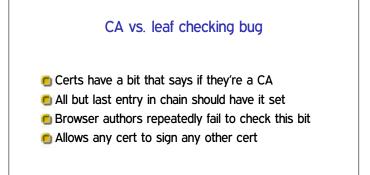
Compression oracle vuln.

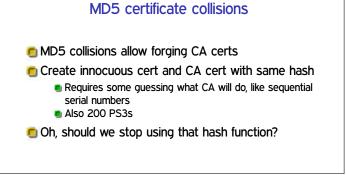
- O Compr(S \parallel A), where S should be secret and A is attacker-controlled
- Attacker observes ciphertext length
- If A is similar to S, combination compresses better
- Compression exists separately in HTTP and TLS

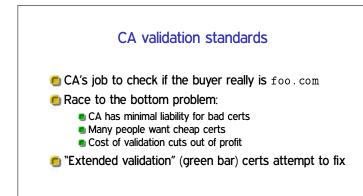


- Kaloper-Meršinjak et al. have longer list "Lessons learned" are variable, though
- Meta-message: don't try this at home

HTTPS hierarchical PKI Browser has order of 100 root certs Not same set in every browser Standards for selection not always clear Many of these in turn have sub-CAs Also, "wildcard" certs for individual domains Hierarchical trust? No. Any CA can sign a cert for any domain A couple of CA compromises recently Most major governments, and many companies you've never heard of, could probably make a google.com cert Still working on: make browser more picky, compare notes

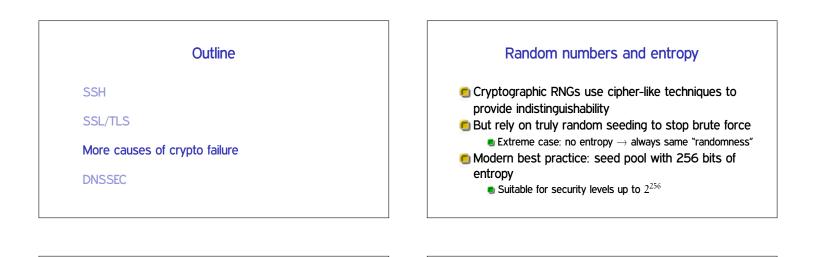






HTTPS and usability

- Many HTTPS security challenges tied with user decisions
- Is this really my bank?
- Seems to be a quite tricky problem
 - Security warnings often ignored, etc.
 - We'll return to this as a major example later

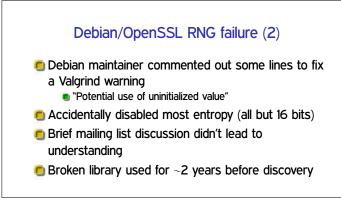


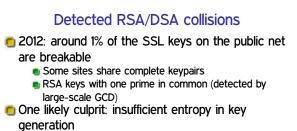
Netscape RNG failure

- Early versions of Netscape SSL (1994-1995) seeded with:
 - Time of day
 - Process ID
 - Parent process ID
- Best case entropy only 64 bits
 - (Not out of step with using 40-bit encryption)
- But worse because many bits guessable

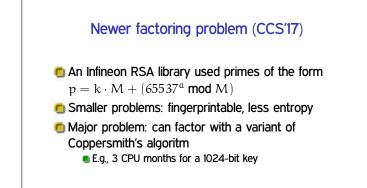
Debian/OpenSSL RNG failure (1)

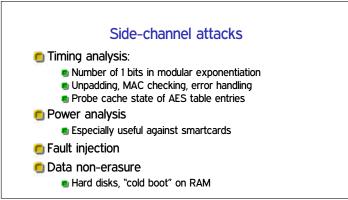
- OpenSSL has pretty good scheme using /dev/urandom
- Also mixed in some uninitialized variable values "Extra variation can't hurt"
- From modern perspective, this was the original sin Remember undefined behavior discussion?
- 🖲 But had no immediate ill effects

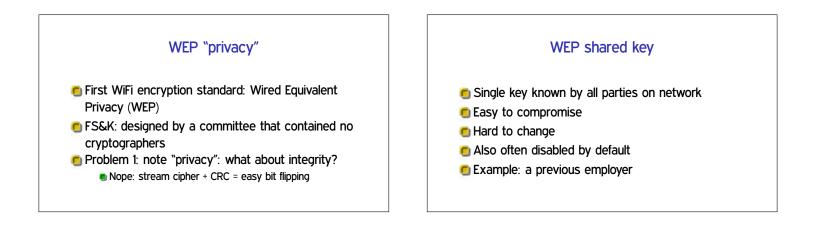




- Embedded devices, Linux /dev/urandom vs. /dev/random
- DSA signature algorithm also very vulnerable









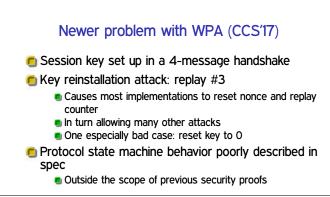
Original sizes: 40-bit shared key (export restrictions) plus 24-bit IV = 64-bit RC4 key

Both too small

- 🖲 128-bit upgrade kept 24-bit IV
 - Vague about how to choose IVs
 - Least bad: sequential, collision takes hours
 - Worse: random or everyone starts at zero

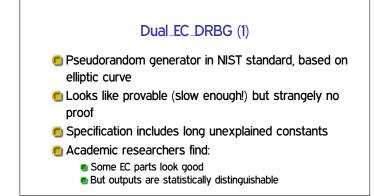
WEP RC4 related key attacks

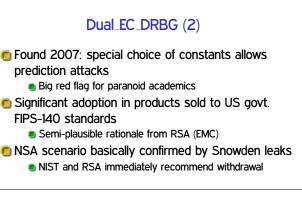
- Only true crypto weakness
- RC4 "key schedule" vulnerable when:
 - RC4 keys very similar (e.g., same key, similar IV)
 - First stream bytes used
- Not such a problem for other RC4 users like SSL
 - Key from a hash, skip first output bytes

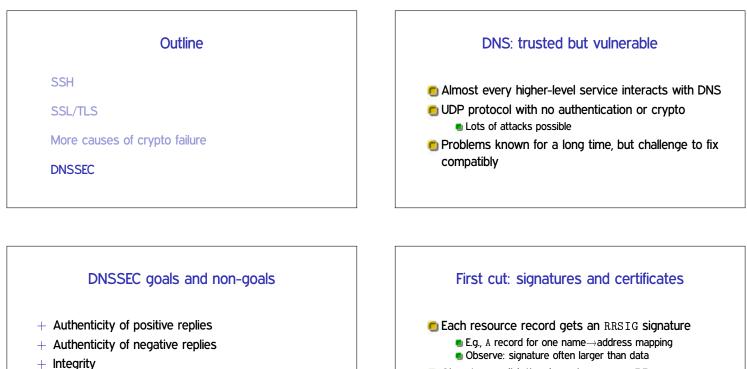


Trustworthiness of primitives

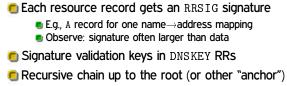
- Classic worry: DES S-boxes
- Obviously in trouble if cipher chosen by your adversary
- In a public spec, most worrying are unexplained elements
- Best practice: choose constants from well-known math, like digits of π

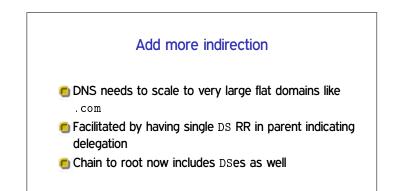


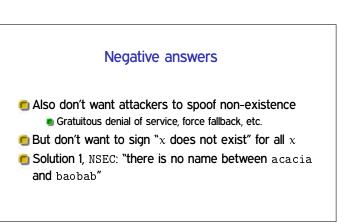


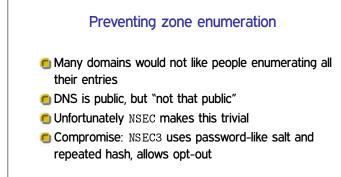


- Confidentiality
- Availability









DANE: linking TLS to DNSSEC

"DNS-based Authentication of Named Entities"

- DNS contains hash of TLS cert, don't need CAs
- How is DNSSEC's tree of certs better than TLS's?

Signing the root

- Political problem: many already distrust US-centered nature of DNS infrastructure
- Practical problem: must be very secure with no single point of failure
- Finally accomplished in 2010
 - Solution involves 'key ceremonies', international committees, smart cards, safe deposit boxes, etc.

Deployment

- Standard deployment problem: all cost and no benefit to being first mover
- Servers working on it, mostly top-down
- Clients: still less than 20%
- Will probably be common for a while: insecure connection to secure resolver

What about privacy?

- Users increasingly want privacy for their DNS queries as well
- Older DNSCurve and DNSCrypt protocols were not standardized
- More recent "DNS over TLS" and "DNS over HTTPS" are RFCs
- DNS over HTTPS in major browsers might have serious centralization effects