CSci 5271 Introduction to Computer Security More crypto protocols and failures

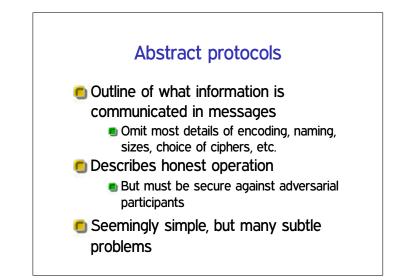
Stephen McCamant University of Minnesota, Computer Science & Engineering

Outline

More crypto protocols

Announcements intermission

More causes of crypto failure



Protocol notation

- $A \rightarrow B : N_B, \{T_0, B, N_B\}_{K_B}$ $\blacksquare A \rightarrow B$: message sent from Alice intended for Bob
 - B (after :): Bob's name
 - ${\color{black} \bullet} {\color{black} \bullet}_{\mathsf{K}}: {\color{black} \bullet} {\color{black$

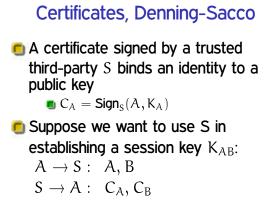
Needham-Schroeder

Mutual authentication via nonce exchange, assuming public keys (core):

 $\begin{array}{ll} A \rightarrow B: \ \{N_A,A\}_{E_B} \\ B \rightarrow A: \ \{N_A,N_B\}_{E_A} \\ A \rightarrow B: \ \{N_B\}_{E_B} \end{array}$

Needham-Schroeder MITM

 $\begin{array}{ll} A \rightarrow C: \ \{N_A, A\}_{E_C} \\ C \rightarrow B: \ \{N_A, A\}_{E_B} \\ B \rightarrow C: \ \{N_A, N_B\}_{E_A} \\ C \rightarrow A: \ \{N_A, N_B\}_{E_A} \\ A \rightarrow C: \ \{N_B\}_{E_C} \\ C \rightarrow B: \ \{N_B\}_{E_B} \end{array}$



 $A \to B: \ C_A, C_B, \{\text{Sign}_A(K_{AB})\}_{K_B}$

Attack against Denning-Sacco

 $\begin{array}{rcl} A \rightarrow S: & A, B \\ S \rightarrow A: & C_A, C_B \\ \hline A \rightarrow B: & C_A, C_B, \{ \text{Sign}_A(K_{AB}) \}_{K_B} \\ \hline B \rightarrow S: & B, C \\ S \rightarrow B: & C_B, C_C \\ B \rightarrow C: & C_A, C_C, \{ \text{Sign}_A(K_{AB}) \}_{K_C} \\ \end{array}$ By re-encrypting the signed key, Bob can pretend to be Alice to Charlie



Implementation principles

- Ensure unique message types and parsing
- Design for ciphers and key sizes to change
- Limit information in outbound error messages
- Be careful with out-of-order messages

Outline

More crypto protocols

Announcements intermission

More causes of crypto failure

Note to early readers

- This is the section of the slides most likely to change in the final version
- If class has already happened, make sure you have the latest slides for announcements

Outline

More crypto protocols

Announcements intermission

More causes of crypto failure

Random numbers and entropy

Cryptographic RNGs use cipher-like techniques to provide indistinguishability

- But rely on truly random seeding to stop brute force
 - \blacksquare Extreme case: no entropy \rightarrow always same "randomness"
- Modern best practice: seed pool with 256 bits of entropy
 - Suitable for security levels up to 2²⁵⁶

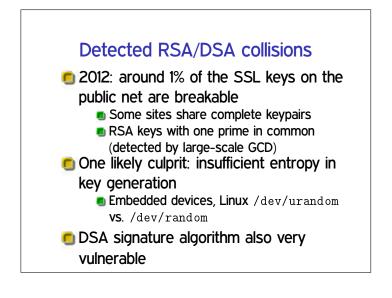
Netscape RNG failure Early versions of Netscape SSL (1994-1995) seeded with: Time of day Process ID Parent process ID Best case entropy only 64 bits (Not out of step with using 40-bit encryption) But worse because many bits

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

Debian/OpenSSL RNG failure (2)

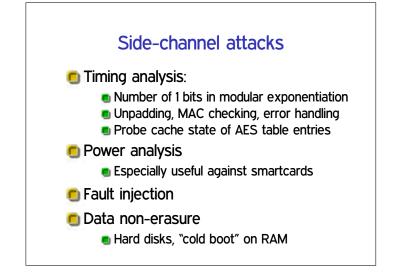
quessable

- Debian maintainer commented out some lines to fix a Valgrind warning "Potential use of uninitialized value"
- Accidentally disabled most entropy (all but 16 bits)
- Brief mailing list discussion didn't lead to understanding
- Broken library used for ~2 years before discovery



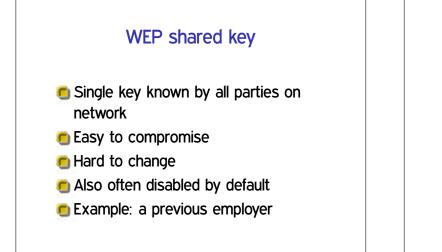
New factoring problem (CCS'17)

- An Infineon RSA library used primes of the form $p = k \cdot M + (65537^a \mod M)$
- Smaller problems: fingerprintable, less entropy
- Major problem: can factor with a variant of Coppersmith's algoritm E.g., 3 CPU months for a 1024-bit key



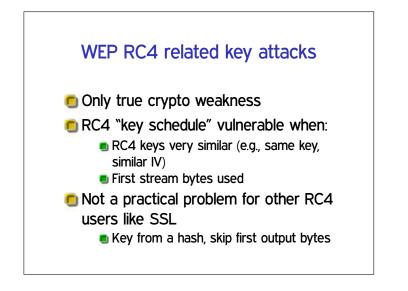


- First WiFi encryption standard: Wired Equivalent Privacy (WEP)
- F&S: designed by a committee that contained no cryptographers
- Problem 1: note "privacy": what about integrity?
 - Nope: stream cipher + CRC = easy bit flipping



WEP key size and IV size

- Original sizes: 40-bit shared key (export restrictions) plus 24-bit IV = 64-bit RC4 key
 Both too small
 - 120 bit un anno de la cast 24
- 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



New problem with WPA (CCS'17)

- Session key set up in a 4-message handshake
- Key reinstallation attack: replay #3
 - Causes most implementations to reset nonce and replay counter
 - In turn allowing many other attacks
 - One especially bad case: reset key to 0

Protocol state machine behavior poorly described in spec

Outside the scope of previous security proofs

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

Dual_EC_DRBG (1)

- Pseudorandom generator in NIST standard, based on elliptic curve
- Looks like provable (slow enough!) but strangely no proof
- Specification includes long unexplained constants
- Academic researchers find:
 - Some EC parts look good
 - But outputs are statistically distinguishable

Dual_EC_DRBG (2)

Found 2007: special choice of constants allows prediction attacks
 Big red flag for paranoid academics
 Significant adoption in products sold to US govt. FIPS-140 standards
 Semi-plausible rationale from RSA (EMC)
 NSA scenario basically confirmed by Snowden leaks
 NIST and RSA immediately recommend withdrawal