CSci 4271W Development of Secure Software Systems Day 22: Cryptography: public key primitives, failures Stephen McCamant

University of Minnesota, Computer Science & Engineering

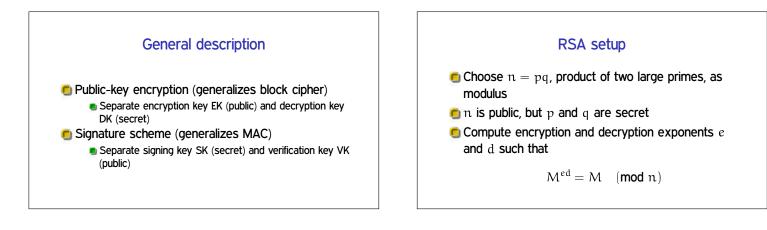
Outline

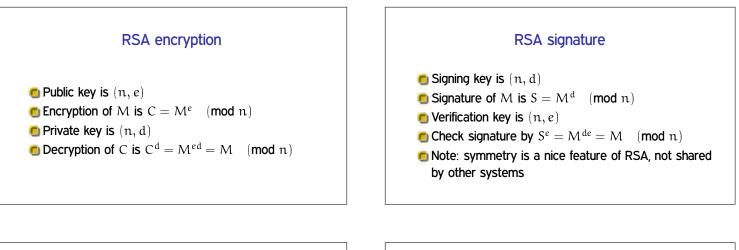
Public key encryption and signatures

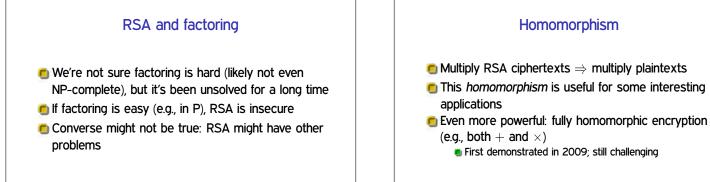
Announcements intermission

Cryptographic protocols

More causes of crypto failure







Problems with vanilla RSA

Homomorphism leads to chosen-ciphertext attacks
 If message and *e* are both small compared to n, can compute M^{1/e} over the integers
 Many more complex attacks too

Hybrid encryption

- Public-key operations are slow
- In practice, use them just to set up symmetric session keys
- + Only pay RSA costs at setup time
- Breaks at either level are fatal

Padding, try #1

- Need to expand message (e.g., AES key) size to match modulus
- PKCS#1 v. 1.5 scheme: prepend 00 01 FF FF .. FF
- Surprising discovery (Bleichenbacher'98): allows adaptive chosen ciphertext attacks on SSL Variants recurred later (c.f. "ROBOT" 2018)

Modern "padding"

- Much more complicated encoding schemes using hashing, random salts, Feistel-like structures, etc.
- Common examples: OAEP for encryption, PSS for signing
- Progress driven largely by improvement in random oracle proofs

Simpler padding alternative

- "Key encapsulation mechanism" (KEM)
- For common case of public-key crypto used for symmetric-key setup

 Also applies to DH
- Choose RSA message r at random mod n, symmetric key is H(r)
- Hard to retrofit, RSA-KEM insecure if e and r reused with different n

Post-quantum cryptography

- One thing quantum computers would be good for is breaking crypto
- Square root speedup of general search Countermeasure: double symmetric security level
- Factoring and discrete log become poly-time
 - DH, RSA, DSA, elliptic curves totally broken
 Totally new primitives needed (lattices, etc.)
- Not a problem yet, but getting ready

Box and locks revisited

- Alice and Bob's box scheme fails if an intermediary can set up two sets of boxes
 - Middleperson (man-in-the-middle) attack
- Real world analogue: challenges of protocol design and public key distribution

Outline

Public key encryption and signatures

Announcements intermission

Cryptographic protocols

More causes of crypto failure

Midterm 2 is Tuesday

- Similar in format to midterm 1 Any paper materials OK, but no electronics
- Covers OS security, web security, and crypto but before public-key
- Past exams and 3/4 solutions on public site

Anderson reading quiz

- There will be a reading quiz on the Anderson cryptography chapter
- 🖲 Won't be due until after next Thursday
- But we'll post on Piazza when it's available
 Might use as part of your midterm studying

Outline

Public key encryption and signatures

Announcements intermission

Cryptographic protocols

More causes of crypto failure

A couple more security goals

Non-repudiation: principal cannot later deny having made a commitment

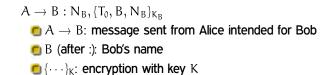
I.e., consider proving fact to a third party

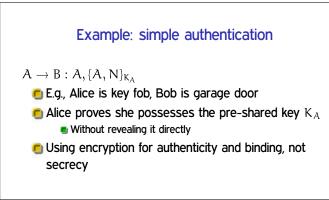
- Forward secrecy: recovering later information does not reveal past information
 - Motivates using Diffie-Hellman to generate fresh keys for each session

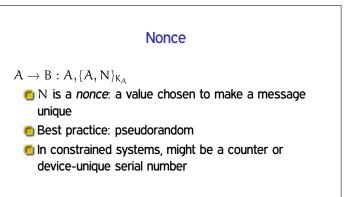
Abstract protocols

- Outline of what information is communicated in messages
 - Omit most details of encoding, naming, sizes, choice of ciphers, etc.
- Describes honest operation
 - But must be secure against adversarial participants
- Seemingly simple, but many subtle problems

Protocol notation







Replay attacks

- A nonce is needed to prevent a verbatim replay of a previous message
- Garage door difficulty: remembering previous nonces
 Particularly: lunchtime/roommate/valet scenario
- Or, door chooses the nonce: challenge-response authentication

Middleperson attacks

- Older name: man-in-the-middle attack, MITM
- Adversary impersonates Alice to Bob and vice-versa, relays messages
- Powerful position for both eavesdropping and modification
- No easy fix if Alice and Bob aren't already related

Chess grandmaster problem

- Variant or dual of middleperson
- Adversary forwards messages to simulate capabilities with his own identity
- How to win at correspondence chess
- 🖲 Anderson's MiG-in-the-middle

Anti-pattern: "oracle"

- Any way a legitimate protocol service can give a capability to an adversary
- Can exist whenever a party decrypts, signs, etc.
- "Padding oracle" was an instance of this at the implementation level

Outline

Public key encryption and signatures

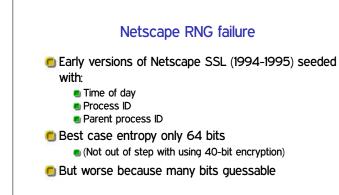
Announcements intermission

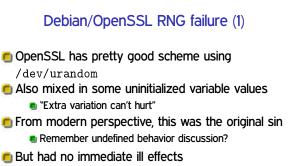
Cryptographic protocols

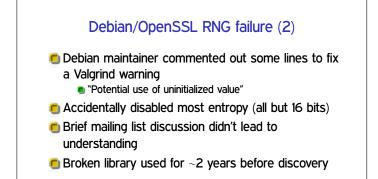
More causes of crypto failure

Random numbers and entropy

- Cryptographic RNGs use cipher-like techniques to provide indistinguishability
- But rely on truly random seeding to stop brute force Extreme case: no entropy — always same "randomness"
- Modern best practice: seed pool with 256 bits of entropy
 - Suitable for security levels up to 2²⁵⁶



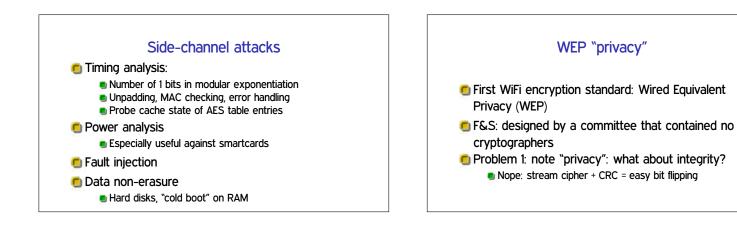




Detected RSA/DSA collisions

2012: around 1% of the SSL keys on the public net are breakable

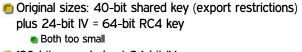
- Some sites share complete keypairs
- RSA keys with one prime in common (detected by large-scale GCD)
- One likely culprit: insufficient entropy in key generation
 - Embedded devices, Linux /dev/urandom vs. /dev/random
- DSA signature algorithm also very vulnerable



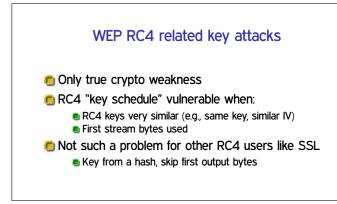
WEP shared key

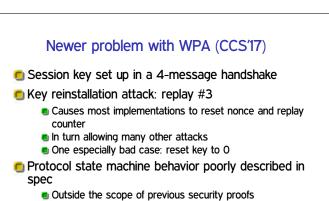
- Single key known by all parties on network
- Easy to compromise
- Hard to change
- Also often disabled by default
- 🖲 Example: a previous employer

WEP key size and IV size



- 🖲 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





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