Miscellaneous Topics in Cryptography
CS 239
Security for System Software
April 19, 2000

Outline

- More on RSA and asymmetric crypto
- Digital signatures
- Digital hashes
- Cryptographic APIs
- Key recovery systems

Using RSA

- Public key consists of two numbers, e and n
- Private key consists of one number, d
- $C = m^e \bmod n$
- $m = C^d \bmod n$
- Works equally well in reverse

Security of RSA

- Conjectured that security depends on factoring large numbers
  - But never proven
  - Some variants proven equivalent to factoring problem
- Probably the conjecture is correct

Attacks on Factoring RSA Keys

- In 1994, a 129 bit RSA key was successfully factored
  - 1600 computers, several months
- Recent paper (fall 2001) on integer factorization suggests keys up to 2048 bits may be insecure
- Size will keep increasing
- The longer the key, the more expensive the encryption and decryption

Comparison of DES and RSA

- DES is much more complex
- However, DES uses only simple arithmetic, logic, and table lookup
- RSA uses exponentiation to very large powers
  - Computationally 1000 times more expensive in hardware
  - 100 times in software
- Key selection also more expensive

Combined Use of Symmetric and Asymmetric Cryptography

- Very common to use both in a single session
- Asymmetric cryptography essentially used to “bootstrap” symmetric crypto
- Use RSA to authenticate and establish a session key
- Use DES/Triple DES/AES using session key for the rest of the transmission

Other Common Asymmetric Crypto Algorithms

- ElGamal
  - Based on calculating discrete logarithms in a finite field
- DSA
  - Also based on discrete logarithms
- Ultimately, cracking discrete log algorithms also boils down to factoring

Elliptic Curve-Based Algorithms

- Typically variants of discrete logarithm algorithms
- Conjectured that they may be more secure
  - Allowing a shorter key length
- Techniques used to factor in traditional algorithms don’t work on elliptic curve algorithms
- So far, no known “replacement” techniques have been discovered for elliptic curves

Asymmetric Cryptography and Crypto Systems

- Asymmetric cryptography (in the large) is highly dependent on supporting system
- Particular a key infrastructure
- We’ll discuss those in more detail later

Digital Signature Algorithms

- In some cases, secrecy isn’t required
- But authentication is
- The data must be guaranteed to be that which was originally sent
- Especially important for data that is long-lived

Desirable Properties of Digital Signatures

- Unforgeable
- Verifiable
- Non-repudiable
- Cheap to compute and verify
- Non-reusable
- No reliance on trusted authority
- Signed document is unchangeable

Encryption and Digital Signatures

- Digital signature methods are based on encryption
- Encryption can be used as a signature

Signatures With Shared Key Encryption

- Requires a trusted third party
- Signer encrypts document with secret key shared with third party
- Receiver checks validity of signature by consulting with trusted third party
- Third party required so receiver can't forge the signature

Signatures With Public Key Cryptography

- Signer encrypts document with his private key
- Receiver checks validity by decrypting with signer's public key
- Only signer has the private key
  - So no trusted third party required
- But receiver must be certain that he has the right public key

Problems With Simple Encryption Approach

- Computationally expensive
  - Especially with public key approach
- Document is encrypted
  - Must be decrypted for use
  - If in regular use, must store encrypted and decrypted versions

Secure Hash Algorithms

- A method of protecting data from modification
- Doesn't actually prevent modification
- But gives strong evidence that modification did or didn't occur
- Typically used with digital signatures

Idea Behind Secure Hashes

- Apply a one-way cryptographic function to data in question
- Producing a much shorter result
- Attach the cryptographic hash to the data before sending
- When necessary, repeat the function on the data and compare to the hash value

Secure Hash Algorithm (SHA)

- Endorsed by NIST
- But produced by the NSA . . .
- Reduces input data of up to 2^{64} bits to 160 bit digest
- Doesn't require secret key
- Generally felt to be reasonably secure

Use of Cryptographic Hashes

- Must assume opponent also has hashing function
- And it doesn't use secret key
- So opponent can substitute a different message with a different hash
- How to prevent this?

Hashing and Signatures

- Use a digital signature algorithm to sign the hash
- But why not just sign the whole message, instead?
- Computing the hash and signing it may be faster than signing the document
- Receiver need only store document plus hash

The Birthday Attack

- How many people must be in a room for the chances to be greater than even that two of them share a birthday?
- Answer is 23
- The same principle can be used to attack hash algorithms

Using the Birthday Attack on Hashes

- For a given document, find a different document that has the effect you want
- Trivially alter the second document so that it hashes to the same value as the target document
  - Using an exhaustive attack

How Hard Is the Birthday Attack?

- Depends on the length of the hash
  - And the quality of the hashing algorithm
- Essentially, looking for hashing collisions
- So long hashes are good
  - SHA produces 2^{80} random hashes

Cryptographic Application Programming Interfaces

- Programs that require cryptographic security should not be written for one particular algorithm
- They should use general methods to interface with cryptography
- Which leads to the need for cryptographic APIs

Some Example CAPIs

- NSA CAPI
- X/Open Generic Cryptographic Services API
- RSA Labs Cryptoki
- HP ICF
- Microsoft CryptoAPI
- GSS-API
- GCS-API

Issues in Designing CAPIs

- Level of cryptographic awareness
- Algorithm independence
- Application independence
- Other typical API issues
- Other typical security issues

Cryptographic Awareness

- General purpose applications should merely call cryptographic services
  - Knowing nothing of their specifics
- Special purpose applications must be cryptographically aware
  - E.g., certificate authority applications
  - Can they also benefit from CAPIs?

GSS-API

- Generic Security Services API
- Supports cryptographically unaware applications
- High level interface to authentication, integrity, non-repudiation, and confidentiality services

Microsoft CryptoAPI

- Supports cryptographically aware applications
- Utilizing both hardware and software cryptographic modules
  - Cryptographic Service Providers
- More application control over algorithms, keys, modes of operation

Legal and Political Issues in Cryptography

- Cryptography is meant to help keep secrets
- But should all secrets be kept?
- Many legal and moral issues

Societal Implications of Cryptography

- Criminals can conceal communications from the police
- Citizens can conceal taxable income from the government
- Terrorists can conceal their activities from governments trying to stop them

Problems With Controlling Cryptography

- Essentially, it's mostly algorithms
- If you know the algorithm, you can have a working copy easily
- At which point, you can conceal your secrets from anybody
  - To the strength the algorithm provides

Governmental Responses to Cryptography

- They vary widely
- Some nations require government approval to use cryptography
- Some nations have no laws governing it at all
- The US laws on the subject are particularly relevant

The US Government Position on Cryptography

- All forms of cryptography are legal to use in the US
- **BUT**
  - Some restrictions on exporting cryptography to other countries
- The NSA also has tried to keep a lid on cryptographic research

US Restrictions on Cryptographic Exports

- Rules changed in 2000
- Greatly liberalizing cryptographic exports
- Almost all cryptography is exportable
- Exception is for government use by a handful of countries
  - Those the US government currently doesn't like

Cryptographic Source Code and Free Speech

- US government took Phil Zimmermann to court over PGP
- Court ruled that he had a free-speech right to publish PGP source
- Eventually, appeals courts also found in favor of Zimmermann

Other Nations and Cryptography

- Generally, most nations have few or no restrictions on cryptography
- A group of treaty signatories have export restrictions similar to US's
- Some have strong restrictions
  - China, Russia, Vietnam, a few others
- A few have laws on domestic use of crypto
  - E.g., Australia, UK, India have laws that demand decryption with court order

Efficacy of These Restrictions

- The world's full of bright programmers
- And the algorithms are publicly available almost anywhere
- So the US regulations don't work, right?
- Wrong!

Key Recovery Cryptosystems

- An attempt to balance:
  - Legitimate societal security needs
    - Requiring strong encryption
  - And legitimate governmental and law enforcement needs
    - Requiring access to data
- How can you have strong encryption and still satisfy governments?

Idea Behind Key Recovery

- Use encryption algorithms that are highly secure against cryptanalysis
- But with mechanisms that allow legitimate law enforcement agency to:
  - Obtain any key with sufficient legal authority
  - Very, very quickly
  - Without the owner knowing

Proper Use of Data Recovery Methods

- All encrypted transmissions (or saved data) must have key recovery methods applied
- Basically, the user must cooperate
  - Or his encryption system must force him to cooperate
  - Which implies everyone must use this form of cryptosystem

Methods to Implement Key Recovery

- Key registry method
  - Register all keys before use
- Data field recovery method
  - Basically, keep key in specially encrypted form in each message
  - With special mechanisms to get key out of the message

Problems With Key Recovery Systems

- Requires trusted infrastructures
- Requires cooperation (forced or voluntary) of all users
- Requires more trust in authorities than many people have
- International issues
- Performance and/or security problems with actual algorithms

The Current Status of Key Recovery Systems

- Pretty much dead
- US tried to convince everyone to use them
  - Skipjack algorithm, Clipper chip
- Very few agreed
- US is moving on to other approaches to dealing with cryptography