Cryptography and Encryption Algorithms
CS 239
Computer Security
January 26, 2005

Outline

- Uses of cryptography
- Symmetric cryptography
- Asymmetric cryptography

Uses of Cryptography

- What can we use cryptography for?
- Lots of things
  - Secrecy
  - Authentication
  - Prevention of alteration

Cryptography and Secrecy

- Pretty obvious
- Only those knowing the proper keys can decrypt the message
  - Thus preserving secrecy
- Used cleverly, it can provide other forms of secrecy

Cryptography and Zero-Knowledge Proofs

- With really clever use, cryptography can be used to prove I know a secret
  - Without telling you the secret
- Seems like magic, but it can work
- Basically, using multiple levels of cryptography in very clever ways

Cryptography and Authentication

- How can I prove to you that I created a piece of data?
- What if I give you the data in encrypted form?
  - Using a key only you and I know
- Then only you or I could have created it
  - Unless one of us told someone else the key . . .

Some Limitations on Cryptography and Authentication

- If both parties cooperative, cryptography can authenticate
  - Problems with non-repudiation, though
- What if three parties want to share a key?
  - No longer certain who created anything
  - Public key cryptography can solve this problem
- What if I want to prove authenticity without secrecy?

Cryptography and Non-Alterability

- Changing one bit of an encrypted message completely garbles it
  - For many forms of cryptography
- If a checksum is part of encrypted data, that's detectable
- If you don't need secrecy, can get the same effect
  - By just encrypting the checksum

Symmetric and Asymmetric Cryptosystems

- Symmetric - the encrypter and decrypter share a secret key
  - Used for both encrypting and decrypting
- Asymmetric – encrypter has different key than decrypter

Description of Symmetric Systems

- $C = E(K,P)$
- $P = D(K,C)$
- *E()* and *D()* are not necessarily symmetric operations

Advantages of Symmetric Key Systems

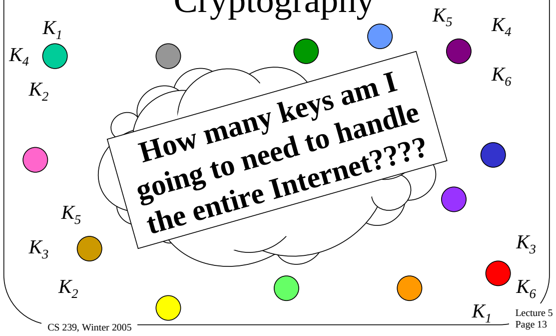
+ Encryption and authentication performed in a single operation
+ Well-known (and trusted) ones perform faster than asymmetric key systems
+ Doesn't require any centralized authority
  - Though key servers help a lot

Disadvantage of Symmetric Key Systems

– Encryption and authentication performed in a single operation
  - Makes signature more difficult

– Non-repudiation hard without servers
– Key distribution can be a problem
– Scaling

Scaling Problems of Symmetric Cryptography



CS 239, Winter 2005

Lecture 5
Page 13

Sample Symmetric Key Ciphers

- The Data Encryption Standard
- The Advanced Encryption Standard
- There are many others

CS 239, Winter 2005

Lecture 5
Page 14

The Data Encryption Standard

- Probably the best known symmetric key cryptosystem
- Developed in 1977
- Still much used
  - Which implies breaking it isn’t trivial
- But showing its age

CS 239, Winter 2005

Lecture 5
Page 15

History of DES

- Developed in response to National Bureau of Standards studies
- Developed by IBM
- Analyzed , altered, and approved by the National Security Agency
- Adopted as a federal standard
- One of the most widely used encryption algorithms

CS 239, Winter 2005

Lecture 5
Page 16

Overview of DES Algorithm

- A block encryption algorithm
  - 64 bit blocks
- Uses substitution and permutation
  - Repeated applications
    - 16 cycles worth
- 64 bit key
  - Only 56 bits really used, though

CS 239, Winter 2005

Lecture 5
Page 17

More On DES Algorithm

- Uses substitutions to provide confusion
  - To hide the set of characters sent
- Uses transpositions to provide diffusion
  - To spread the effects of one plaintext bit into other bits
- Uses only standard arithmetic and logic functions and table lookup

CS 239, Winter 2005

Lecture 5
Page 18

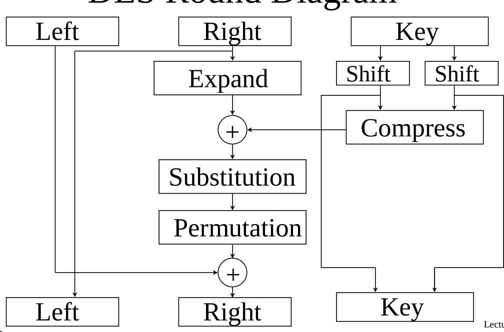
Description of DES Algorithm

- Alternate applications of two different ciphers
  - A *product cipher*
- Starts by breaking block in half
- The algorithm goes through 16 *rounds*
- Each round consists of a substitution followed by a permutation

One DES Round

- Select 48 bits from the key
- Expand right half of block to 48 bits
- XOR with key bits
- Look up result in an S-box
  - Resulting in 32 bits
- Perform a permutation using a P-box
- XOR with left half of block
- Result is new right half
- Old right half becomes new left half

DES Round Diagram



S-Boxes

- Table lookups to perform substitutions
- Permanently defined for DES
- Eight different S-boxes
  - Six bits out of 48 bits go to each
  - Four bits come out of each
- Choice of contents of S-boxes believed to strongly impact security of DES

P-Box

- Maps 32 input bits to 32 output bits
- A single, straight permutation
  - Unlike S-boxes, which are table lookups

Decrypting DES

- For DES, *D()* is the same as *E()*
- You decrypt with exactly the same algorithm
- If you feed ciphertext and the same key into DES, the original plaintext pops out

Is DES Secure?

- Apparently, reasonably
- No evidence NSA put a trapdoor in
  - Alterations believed to have increased security against differential cryptanalysis
- Some keys are known to be weak with DES
  - So good implementations reject them
- To date, only brute force attacks have publicly cracked DES

Key Length and DES

- Easiest brute force attack is to try all keys
  - Looking for a meaningful output
- Cost of attack proportional to number of possible keys
- Is 2^{56} enough keys?

DES Cracking Experiments

- RSA Data Security issued challenge to crack a DES-encrypted message
- Various people got together to do so
  - Harnessing computers across the Internet
  - Using a brute-force approach
- Done in 1998

How the DES Message Was Cracked

- Required use of tens of thousands of computers
- Took four months
- The searchers “got lucky”
  - Only one quarter of key space searched
  - On average, brute force requires searching one half of key space
- Done over six years ago
  - So it would presumably take 1/16 as much time today

DES and Differential Cryptography

- Research has shown that DES is somewhat susceptible to differential cryptography
- NSA alterations to original DES seem to have strengthened it against this attack
- Only relevant for chosen-plaintext attack scenarios

Does This Mean DES is Unsafe?

- Depends on what you use it for
- In how many cases will tens of thousands of machines apply spare cycles for several days to break one message?
- On the other hand, computers will continue to get faster
- And motivated opponents can harness vast resources
- Those who care seriously about security don’t tend to use DES any more

Triple DES

- Simple way of increasing security of DES
- Apply DES three times iteratively to each block
  - Thus, 1/3 as fast as DES
- Use different key for each encryption
- Effectively doubles the key length of DES
- Approved by NIST
  - Which recommends using in in preference to DES

The Advanced Encryption Standard

- A relatively new cryptographic algorithm
- Intended to be the replacement for DES
- Chosen by NIST
  - Through an open competition
- Chosen cipher was originally called Rijndael
  - Developed by Dutch researchers
  - Uses combination of permutation and substitution

Increased Popularity of AES

- Appears to be gradually replacing DES
  - As was intended
- Various RFCs describe using AES in IPSEC
- FreeS/WAN IPSEC (for Linux) includes AES
- Commercial VPNs that use AES are available

Public Key Encryption Systems

- The encrypter and decrypter have different keys

$C = E(K_E,P)$

$P = D(K_D,C)$

- Often, works the other way, too

$C' = E(K_D, P)$

$P = D(K_E, C')$

History of Public Key Cryptography

- Invented by Diffie and Hellman in 1976
- Merkle and Hellman developed Knapsack algorithm in 1978
- Rivest-Shamir-Adelman developed RSA in 1978
  - Most popular public key algorithm
- Many public key cryptography advances secretly developed by British and US government cryptographers earlier

Practical Use of Public Key Cryptography

- Keys are created in pairs
- One key is kept secret by the owner
- The other is made public to the world
- If you want to send an encrypted message to someone, encrypt with his public key
  - Only he has private key to decrypt

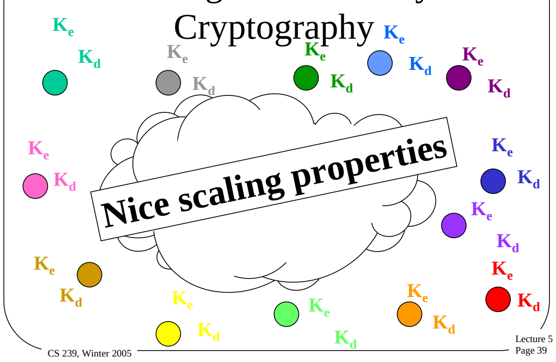
Authentication With Shared Keys

- If only two people know the key, and I didn't create a properly encrypted message -
  - The other guy must have
- But what if he claims he didn't?
- Or what if there are more than two?
- Requires authentication servers

Authentication With Public Keys

- If I want to "sign" a message, encrypt it with my private key
- Only I know private key, so no one else could create that message
- Everyone knows my public key, so everyone can check my claim directly

Scaling of Public Key Cryptography



Key Management Issues

- To communicate via shared key cryptography, key must be distributed
  - In trusted fashion
- To communicate via public key cryptography, need to find out each other's public key
  - "Simply publish public keys"

Issues of Key Publication

- Security of public key cryptography depends on using the right public key
- If I am fooled into using the wrong one, that key's owner reads my message
- Need high assurance that a given key belongs to a particular person
- Which requires a *key distribution infrastructure*

RSA Algorithm

- Most popular public key cryptographic algorithm
- In wide use
- Has withstood much cryptanalysis
- Based on hard problem of factoring large numbers

RSA Keys

- Keys are functions of a pair of 100-200 digit prime numbers
- Relationship between public and private key is complex
- Recovering plaintext without private key (even knowing public key) is supposedly equivalent to factoring product of the prime numbers

Comparison of DES and RSA

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

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 2003, a 576 bit RSA key was successfully factored
  - Using supercomputers at three major German universities and other hardware
- Research 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

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 (or another PK algorithm) to authenticate and establish a session key
- Use DES/Triple DES/AES using session key for the rest of the transmission

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
  – But recently attacks found on "cousins" of SHA-1

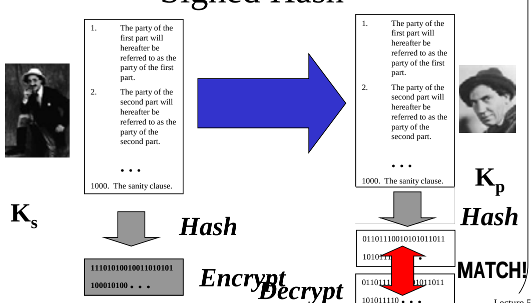
Use of Cryptographic Hashes

- 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?
- And what (if anything) would secure hashes actually be useful for?

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

Checking a Document With a Signed 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

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 cryptography at all
- The US laws less restrictive than they used to be

The US Government Position on Cryptography

- All forms of cryptography are legal to use in the US
- **<u>BUT</u>**
  - Some minor restrictions on exporting cryptography to other countries
- The NSA used to try 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

Key Recovery Cryptosystems

- An attempt to balance:
  - Legitimate societal security needs
    - Which require strong encryption
  - And legitimate governmental and law enforcement needs
    - Which require 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 (for widespread use)
- 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
- Some businesses run key recovery internally
  - More to avoid losing important data when keys lost than for any other reason