

Cryptology

introduction

Björn Victor

Spring 2008

Administrativia

Teachers Björn Victor and Frédéric Haziza

Lab assistant Ke Jiang

Web <http://www.it.uu.se/edu/course/homepage/secure/vt08> (or search for “crypto” on the front page).

Language in English, men prata svenska om ni tycker det är besvärligt!

Litterature Stinson: *Cryptology - theory and practise* (3rd ed)

Examination

- Two labs: breaking and implementing ciphers
- PM + presentation: write a short PM on an interesting subject, present to class.
- Final exam.

Plus exercises, mathematical and practical.

Overview

- introduction & motivation
- basic theory
- symmetric, shared-key, ciphers
- asymmetric, public-key, ciphers
- key management & agreement
- digital signatures, data integrity

Theory throughout, mixed with practical examples and exercises.

Laborations

Labs to be done in groups of two, preferred language C/C++ (can be discussed).

- 1 *Vigenère cryptanalysis*: break a cipher which was considered unbreakable for hundreds of years
- 2 *Public-key cryptography*: implement the RSA algorithm for public-key crypto

Computer INsecurity: concepts

Asset: (tillgång) hardware, software, data, information, reputation. . .

Vulnerability: (svaghet) weaknesses in design/implementation/procedure which may be exploited to cause loss/harm to *assets*

Threat: (hot) set of circumstances/actions which potentially cause loss/harm to assets

Attack: exploit of vulnerability

Control: action, device, procedure, technique which removes (or reduces) vulnerability

A threat is *blocked* by control of a vulnerability.

Basic threats

- interception: unauthorized access to information
- interruption: unavailability of authorised access (delete, destruct etc)
- modification: unauthorised (or not)
- fabrication: unauthorised, (inject data, falsifications, etc)

Give examples of attacks on computer security, try to classify the threats.

Aspects/goals of computer security

Confidentiality: prevent unauthorised *disclosure* of information

Integrity: prev. unauth. *modification* of info

Availability: prevent unauthorised withholding of info (or resources)

plus accountability, authenticity, trust, risk, privacy, anonymity, reliability, dependability. . .

Policy and mechanism

Policy

- *specifies* what security we want to achieve, e.g., “only teachers can set the grades of students”, “only the owner of an object can grant permissions for others to that object”
- can be used to formalise the goals

Mechanism

- the *methods* which can be used to fulfill the policy requirements; the “implementation” of security
- can also be formalised

The mechanisms should enforce the policy. Validate formally if/when possible.

Mechanisms

Example: direct access control.

- capability lists: what may a given *subject* do to which *objects*? Cf. tickets, memory protection
- access control lists: which subjects may do what to a given object? Cf. invitation list, file protection

Mechanisms

Example: information flow control.

Direct access control does not handle everything: (example).

- Need to analyse how information flows, and
- to control the information flows.

Examples: Bell-LaPadula system, Chinese Wall system.

Mechanisms

Example: cryptography.

Can be used to encode data so that

- it can not be read by the wrong subject (confidentiality)
- it can not be sent by the wrong subject (authenticity)
- it can not have been modified (integrity)

Cryptography

- symmetric:** the same (secret) key is used for both encryption and decryption
- asymmetric:** two different keys are used for encryption and decryption
- protocols:** e.g. for authentication, confidentiality, key distribution, etc.

Symmetric cryptography

Encryption and decryption use the same key.

- the key must be known to both sender and receiver
- key distribution problems to solve

Can be implemented very efficiently (e.g. in hardware).

Example algorithms: Cæsar, Vigenère, DES, IDEA, Blowfish, AES...

Asymmetric cryptography

Encryption and decryption use *different* keys.

- keys come in pairs: “inverses” of each other
- one is used for encryption, the other for decryption
- one can not be computed from the other

Public-key cryptography:

- each user has a key pair: one is public, one is private
- no *shared* secrets
- less problems with key distribution, but still exist

Much slower than symmetric cryptography. Often used in combination, e.g. to distribute a shared key.

Examples: RSA (Rivest-Shamir-Adleman), Diffie-Hellman.

Cryptographic protocols

Example: using public-key crypto to distribute shared key for symmetric crypto.

- 1 A creates random symmetric key k ; encrypts it with the public key of B
- 2 A sends it to B (over a public channel)
- 3 Only B can decrypt it (using the corresponding private key)
- 4 The key k can then be used for efficient confidential communication between A and B (“session key”)

Digital signatures

A signature for a message is a hash value (“checksum”) of the message, encrypted with the sender’s private key.

- *everyone* can decrypt the checksum and verify that it matches the message – using the public key
- *only* the sender could have encrypted the checksum – using the private key

Using security mechanisms

Using security mechanisms in computing should as natural(?) as e.g.

- locking your door when you're not home (but an unlocked door is not an invitation for anyone)
- using sealed envelopes for sensitive messages (rather than open postcards)
- not using open radio communication for private communication (e.g. GSM encryption)

Not always easy to use. Compare your experience with your parents'?

Exercise: get a certificate for secure email (signed, encrypted). See course web for instructions.