Algebraic Foundations of Computer Science.

Applications to Cryptography

Ferucio Laurenţiu Ţiplea

Department of Computer Science
"AL.I.Cuza" University of Iaşi
Iaşi, Romania
E-mail: fltiplea@mail.dntis.ro

Spring 2017

Outline

Outline

1. Introduction to cryptography
   - Introduction to cryptography
   - Cryptosystem and cryptanalysis
   - The RSA cryptosystem
   - Digital signatures
   - Secret sharing schemes
2. Course readings

Introduction to Cryptography

- Cryptography is the field concerned with techniques for securing information, particularly in communications;
- Cryptography focuses on the following paradigms:
  - Authentication – the process of proving one's identity (the primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak);
  - Privacy/confidentiality – ensuring that no one can read the message except the intended receiver;
  - Integrity – assuring the receiver that the received message has not been altered in any way from the original;
  - Non-repudiation – a mechanism to prove that the sender really sent this message.

Applications of cryptography

Applications of cryptography include:

- computer and information security: cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet.
- e-commerce, e-payment, e-voting, e-auction, e-lottery, and e-gambling schemes, are all based on cryptographic (security) protocols.

Examples of software tools that havily rely on cryptographic techniques: IPsec, SSL & TLS, DNSsec, S/MIME, SET etc.

History of cryptography

A brief history of cryptography is in order:

- The oldest forms of cryptography date back to at least Ancient Egypt, when derivations of the standard hieroglyphs of the day were used to communicate;
- Julius Caesar (100-44 BC) used a simple substitution cipher with the normal alphabet (just shifting the letters a fixed amount) in government communications (Caesar cipher);
- Thomas Jefferson, the father of American cryptography, invented a wheel cipher in the 1790's, which would be redeveloped as the Strip Cipher, M-138-A, used by the US Navy during World War II;

History of cryptography

- During World War II, two notable machines were employed: the German's Enigma machine, developed by Arthur Scherbius, and the Japanese Purple Machine, developed using techniques first discovered by Herbert O. Yardley;
- William Frederick Friedman, the father of American cryptanalysis, led a team which broke in 1940 the Japanese Purple Code;
- In the 1970s, Horst Feistel developed a "family" of ciphers, the Feistel ciphers, while working at IBM's Watson Research Laboratory. In 1976, The National Security Agency (NSA) worked with the Feistel ciphers to establish FIPS PUB-46, known today as DES;

History of cryptography

- In 1976, Martin Hellman, Whitfield Diffie, and Ralph Merkle, have introduced the concept of public-key cryptography;
- In 1977, Ronald L. Rivest, Adi Shamir and Leonard M. Adleman proposed the first public-key cipher which is still secure and used (it is known as RSA);
- The Electronic Frontier Foundation (EFF) built the first unclassified hardware for cracking messages encoded with DES. On July 17, 1998, the EFF DES Cracker was used to recover a DES key in 22 hours. The consensus of the cryptographic community was that DES was not secure;
- In October 2001, after a long searching process, NIST selected the Rijndael cipher, invented by Joan Daemen and Vincent Rijmen, as the Advanced Encryption Standard. The standard was published in November 2002.

Outline

1. Introduction to cryptography
   - Introduction to cryptography
   - Cryptosystem and cryptanalysis
   - The RSA cryptosystem
   - Digital signatures
   - Secret sharing schemes
2. Course readings

Cryptosystem

Definition 1

A cryptosystem or cipher is a 5-tuple $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where:

1. $\mathcal{P}$ is a non-empty finite set of plaintext symbols;
2. $\mathcal{C}$ is a non-empty finite set of cryptotext symbols;
3. $\mathcal{K}$ is a non-empty finite set of keys;
4. $\mathcal{E}$ and $\mathcal{D}$ are two sets of functions (algorithms)

$$\mathcal{E} = \{e_K : \mathcal{P} \to \mathcal{C} | K \in \mathcal{K}\} \quad \text{and} \quad \mathcal{D} = \{d_K : \mathcal{C} \to \mathcal{P} | K \in \mathcal{K}\},$$

such that $d_K(e_K(x)) = x$, for any $K \in \mathcal{K}$ and $x \in \mathcal{P}$.

e_K is the encryption rule (algorithm), and d_K is the decryption rule (algorithm), induced by K.

Encryption variants

Let $\mathcal{S} = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be a cipher. A plaintext (cryptotext) is a finite sequence of plaintext (cryptotext) symbols.

Encryption modes of a plaintext $x = x_1 \cdots x_n$:

- (Fixed-key encryption). Generate a key K and encrypt each plaintext symbol by e_K:

$$y = e_K(x_1) \cdots e_K(x_n);$$

- (Variable-key encryption). Generate a sequence of keys $K_1, \ldots, K_n$ and encrypt each plaintext symbol x_i by e_{K_i}:

$$y = e_{K_1}(x_1) \cdots e_{K_n}(x_n).$$

Remark 1

We will mainly use the fixed-key encryption mode.

Classification of cryptosystems

Cryptosystems can be classified into:

- symmetric (private-key, single-key) cryptosystems – characterized by the fact that it is easy to compute the decryption rule d_K from e_K, and vice-versa;
- asymmetric (public-key) cryptosystems – characterized by the fact that it is hard to compute d_K from e_K. With such cryptosystems, the key K is split into two subkeys, K_e, for encryption, and K_d, for decryption. Moreover, K_e can be made public without endangering security.

Symmetric cryptosystems

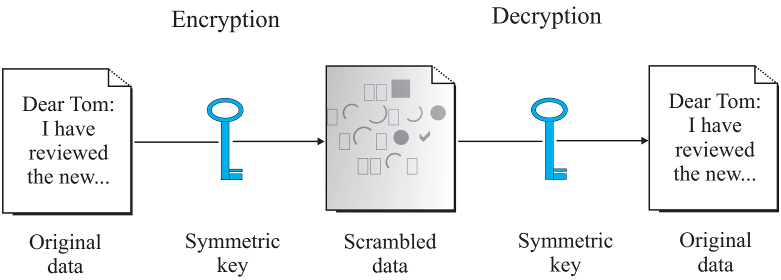

Figure: With symmetric cryptosystems, the same key is used for both encryption and decryption

Asymmetric cryptosystems

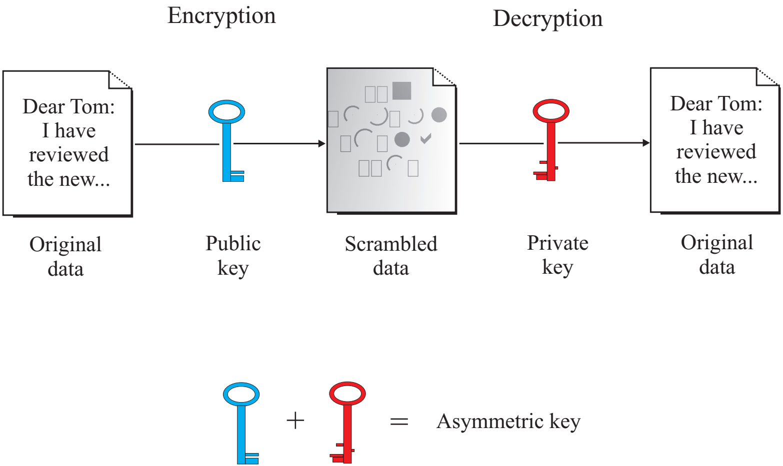

Figure: With asymmetric cryptosystems, a key is used for encryption and another key is used for decryption

Symbol – integer correspondence

Most cryptosystems are based on number theory and, therefore, it is customary to view each plaintext symbol as an integer, for instance, based on a one-to-one correspondence like the one below:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

For instance, the plaintext “home” becomes the sequence of integers “7,14,12,4”.

Affine cryptosystems

Cryptosystem 1 (Affine Cryptosystems)

An affine cryptosystem is defined as follows:

- $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$;
- $\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} | gcd(a, 26) = 1\}$;
- for any key $K = (a, b)$ and $x, y \in \mathbb{Z}_{26}$,

$$e_K(x) = (ax + b) \ mod \ 26 \quad \text{and} \quad d_k(y) = (a^{-1}(y - b)) \ mod \ 26.$$

Let $K = (7, 3)$ and the plaintext $pt = hot$ ($pt = 7, 14, 19$). Then,

$$e_K(pt) = e_K(7), e_K(14), e_K(19) = 0, 23, 6,$$

that is, the cryptotext is $ct = axg$.

Cryptanalysis of affine cryptosystems

Affine cryptosystems can be easily broken by exhaustive key search (EKS), also known as brute-force search, which consists of trying every possible key until you find the right one.

Question: If you have a chunk of cryptotext and decrypt it with one key after the other, how do you know when you have found the correct plaintext?

> *Answer:* *You know that you have found the plaintext because it looks like plaintext. Plaintext tends to look like plaintext. It's an English-language message, or a data file from a computer application (e.g., programs like Microsoft Word have large known headers), or a database in a reasonable format. When you look at a decrypted file, it looks like something understandable. When you look at a cryptotext file, or a file decrypted with the wrong key, it looks like gibberish.*

Cryptanalysis of affine cryptosystems

Question: How many keys are?

Answer: If an affine cryptosystem is developed over $\mathbb{Z}_{26}$, *then there are only* $\phi(26) \times 26 = 12 \times 26 = 312$ *possible keys.*

As a conclusion, given an affine cryptosystem, it is very easy to enumerate all its keys and break it using a laptop (assuming that you have a chunk of cryptotext).

Outline

The RSA cryptosystem

In 1977, Ronald Rivest, Adi Shamir, and Leonard Adleman, proposed the first public-key cryptosystem which is still secure and used.

Cryptosystem 2 (RSA)

- let p and q be two distinct primes, and $n = pq$;
- $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$;
- $\mathcal{K} = \{(n,p,q,e,d) | e \in \mathbb{Z}^*_{\phi(n)} \ \wedge \ ed \equiv 1 \ mod \ \phi(n)\}$;
- for any $K = (n,p,q,e,d) \in \mathcal{K}$ and $x, y \in \mathbb{Z}_n$,

$$e_K(x) = x^e \ mod \ n \ \text{ and } \ d_K(y) = y^d \ mod \ n;$$

- (n, e) is the public key, and (p, q, d) is the secret key.

The RSA cryptosystem

Example 2 (RSA with artificially small parameters)

Let $p = 61$ *and* $q = 53$. *Then:*

- $n = pq = 3233$ *and* $\phi(n) = 3120$;
- *if we chose* $e = 17$, *then* d *can be computed with the extended Euclidean algorithm. We obtain* $d = e^{-1} \bmod 3120 = 2753$;
- $n = 3233$ *and* $e = 17$ *are public parameters;* p, q, *and* d *secrete;*

Let $x = 123$ *be a plaintext. The cryptotext is*

$$y = 123^{17} \bmod 3233 = 855.$$

In order to decrypt y *we have to compute*

$$855^{2753} \bmod 3233 = 123.$$

Security of RSA

Security issues:

- if p or q is recovered (e.g., by factoring n in reasonable time), then the system is completely broken;
- if $\phi(n)$ can be computed in reasonable time, then the system is completely broken;
- if d can be easily computed from n and e, then the system is completely broken.

In practice:

- p and q are 512-bit primes (or even larger);
- e is small (fast encryption) but chosen such that $d > \sqrt[4]{n}$ (otherwise, an efficient attack can be mounted).

For more details: http://www.rsasecurity.com/.

Digital signatures

Public key cryptography solves another problem crucial to e-commerce and Internet cyber relationship: it lets you emulate written signatures. This use of public key technology is called a digital signature.

A digital signature must provide:

- authenticity and integrity. That is, it must be "impossible" for anyone who does not have access to the secret key to forge (x, σ) (x is the original data and σ is its associated signature);
- non-repudiation. That is, it must be impossible for the legitimate signer to repudiate his own signature.

Signing (encrypting with a private key) is extremely slow, so you usually add a time-saving (and space-saving) step before you encrypt messages. It is called message digesting or hashing.

Digital signatures and message digests

A hash algorithm (function) is an algorithm (function) which, applied to an arbitrary-length input data, produces a fixed-length output data (called a hash value or message digest or fingerprint). Digital signatures are usually applied to message digests.

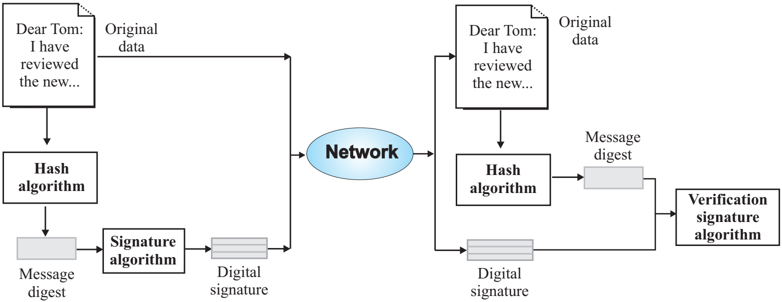

Figure: Hashing and digital signatures

Digital signatures from public key cryptosystems

Any public key cipher can be used to produce digital signatures:

- Assume that K_e is A's public key and K_d is A's private key and, moreover, $e_{K_e}(d_{K_d}(x)) = x$;
- Then, the decryption of a message x by K_d is the digital signature associated to x. It can be verified by K_e:

$$x \stackrel{?}{=} e_{K_e}(d_{K_d}(x)).$$

  Therefore, in such a case, K_d is used to sign messages (it will be secret) and K_e is used to verify signatures (it will be public).

The RSA signature is obtained from the RSA public key cipher.

Outline

1. Introduction to cryptography
   - Introduction to cryptography
   - Cryptosystem and cryptanalysis
   - The RSA cryptosystem
   - Digital signatures
   - Secret sharing schemes

2. Course readings

Threshold sharing schemes

An important application of the Chinese remainder theorem concerns the construction of (k, n)-threshold sharing schemes.

Definition 3

A (k, n)-threshold sharing scheme consists of n people $P_1, \ldots, P_n$ sharing a secret S in such a way that the following properties hold:

- *$k \leq n$;*
- *each P_i has an information I_i;*
- *knowledge of any k of $I_1, \ldots, I_k$ enables one to find S easily;*
- *knowledge of less than k of $I_1, \ldots, I_k$ does not enable one to find S easily.*

Mignotte's threshold sharing schemes

We will show how a (k, n)-threshold sharing scheme can be constructed:

- let

$$\underbrace{m_1 < \cdots < m_k}_{\text{first } k \text{ numbers}} < \cdots < \underbrace{m_{n-k+2} < \cdots < m_n}_{\text{last } k-1 \text{ numbers}}$$

  be a sequence of pairwise co-prime numbers such that

$$\alpha = m_1 \cdots m_k > m_{n-k+2} \cdots m_n = \beta;$$

- let S be a secret, $\beta < S < \alpha$;
- each P_i gets the information $I_i = S \bmod m_i$.

This is called Mignotte's threshold sharing scheme.

Soundness of secret recovery

Any group of k people, $P_{i_1}, \ldots, P_{i_k}$, can recover uniquely the secret S by solving the system:

$$(*) \begin{cases} x & \equiv & I_{i_1} \ mod \ m_{i_1} \\ & \cdots & \\ x & \equiv & I_{i_k} \ mod \ m_{i_k} \end{cases}$$

According to the Chinese remainder theorem, this system has a unique solution modulo $m_{i_1} \cdots m_{i_k}$, and this solution is S because

$$S < \alpha \leq m_{i_1} \cdots m_{i_k}.$$

Security to coalition attack

No group of $k-1$ people, $P_{j_1}, \ldots, P_{j_k}$, can recover uniquely the secret S by solving the system:

$$(**) \begin{cases} x & \equiv & I_{j_1} \ mod \ m_{j_1} \\ & \cdots & \\ x & \equiv & I_{j_{k-1}} \ mod \ m_{j_{k-1}} \end{cases}$$

According to the Chinese remainder theorem, this system has a unique solution modulo $m_{j_1} \cdots m_{j_{k-1}}$, and this solution, denoted x_0, satisfies

$$x_0 < m_{j_1} \cdots m_{j_{k-1}} \leq \beta,$$

while $\beta < S$.

Course readings

1. F.L. Ţiplea: *Fundamentele Algebrice ale Informaticii*, Ed. Polirom, Iaşi, 2006, pag. 268–283.
2. S. Iftene: *Secret Sharing Schemes with Applications in Security Protocols*, Ph.D. Thesis, "Al.I.Cuza" University of Iaşi, 2007, http://thor.info.uaic.ro/~tr/tr.pl.cgi.
3. C.C. Drăgan: *Security of CRT-based Secret Sharing Schemes*, Ph.D. Thesis, "Al.I.Cuza" University of Iaşi, 2013.