IT Compact Course

Hardware and Software

Internet and Web

Cryptography

Kaspar Etter, October 2011

License: CC BY-NC-ND 3.0

3. Cryptography

Outline

3.1. Introduction

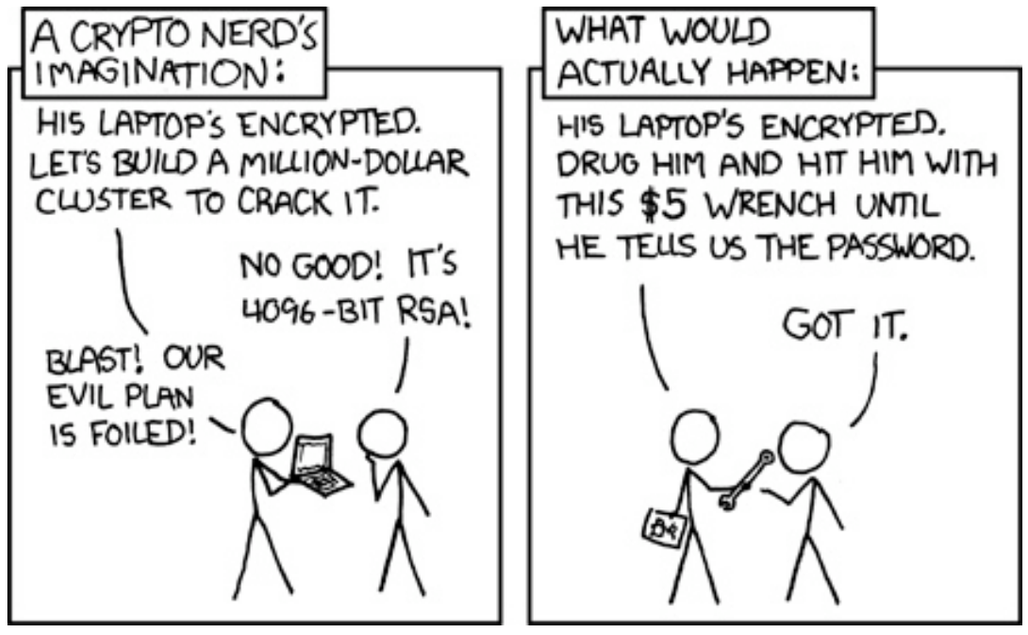
3.2. Cryptographic Primitives

3.3. Symmetric-key Crypto.

3.4. Public-key Cryptography

3.5. Group Theory

3.6. Cryptosystems



General reference and source for further reading: www.wikipedia.org

3. Cryptography

3.1. Introduction

- Cryptography: κρυπτός (secret) and γράφειν (writing)
  $\Rightarrow$ Secure communication in the presence of third parties
- Long history of encryption, dating back to ancient times
- For a long time, encryption was more art than science
- Today based on computational hardness assumptions about problems like integer factorization or discrete logarithm
- Distinction between computationally secure (hard to break) and information-theoretically secure (cannot be broken)
- No quantum computing and cryptography covered today

3.1. Introduction

Encryption and Decryption

Encryption converts plaintext into unintelligible ciphertext

Decryption requires special knowledge to recover plaintext

Setting: Alice wants to send a message to Bob, while a passive or active adversary intercepts their (secret) communication

Attacker model (what the adversary can do):

- Adversary has limited or unlimited computing power
- Passive: Eavesdrop on communication channel (called Eve)
- Active: Remove and insert messages (Mallory or Trudy)

3.1. Introduction

Authentication, Authorization

Authentication is about confirming the identity of an entity

Identity is the property that makes two objects the same
(The identity relation is transitive, symmetric and reflexive)

Authorization is about specifying the permissions of an entity

Authentication factors for human users:

- Something they are: fingerprint, retinal pattern, signature
- Something they have: wrist band, badge, security token
- Something they know: password/PIN, response to challenge

3.1. Introduction

Information Security

Three main objectives in information security (CIA):

- Confidentiality: Keeping communication and data secret
- Integrity: Checking messages and authenticating the sender
- Availability: Ensuring the access to information and services

Possible attacks:

- Brute-force: Check all possible keys (vs. increase key length)
- Man-in-the-middle: Relay messages between unaware victims
- Denial-of-service (DoS): Make a service unavailable to users

Kerckhoffs's Principle

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge (1883)

$\Longrightarrow$ Less expensive to change the keys than the cryptosystem

$\Longrightarrow$ A necessity for cryptosystems deployed in user machines

$\Longrightarrow$ Keeping it secret is still another hurdle for the enemy

Contrary principle: Security through obscurity

$\Longrightarrow$ Problems: Reverse engineering, no peer review, backdoors

$\Longrightarrow$ Steganography: Hiding even the existence of a message

3.1. Introduction

Legal Issues

- In the UK, a suspected criminal can be forced to hand over their encryption key if asked by law enforcement (violation of the right of not being forced to incriminate oneself)
- In China, a license is required to use cryptography
- Cryptography considered important for national security, therefore the USA regulated its export until the 1990s
- Cryptography central to digital rights management (DRM)
- Digital Millennium Copyright Act, signed by B. Clinton, 1998: Criminalizes the circumvention of DRM technologies

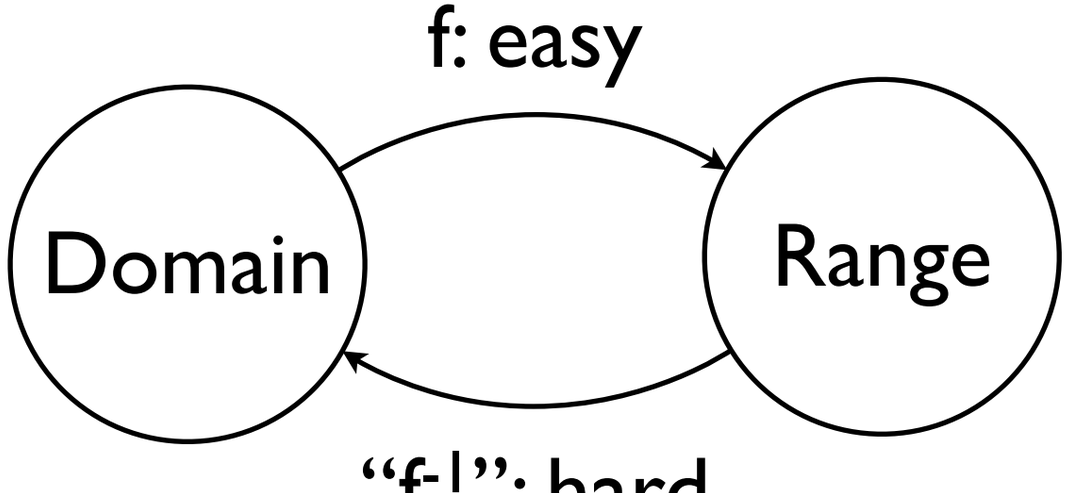
3.2. Cryptographic Primitives

- Cryptographic primitives are fundamental techniques used as building blocks in the design of cryptographic protocols
- They fulfill a specific task according to an abstract concept
- In practice, they are instantiated with concrete algorithms
- Combining crypto. primitives is an art requiring deep insight

Examples:

- Cryptographic hash functions
- Random number generators

One-way Functions



Given a one-way function f and y, hard to find x with f(x) = y

Existence of one-way functions is a conjecture ($\Longrightarrow$ P ≠ NP)

Hash function: arbitrary block of data → fixed-size bit string

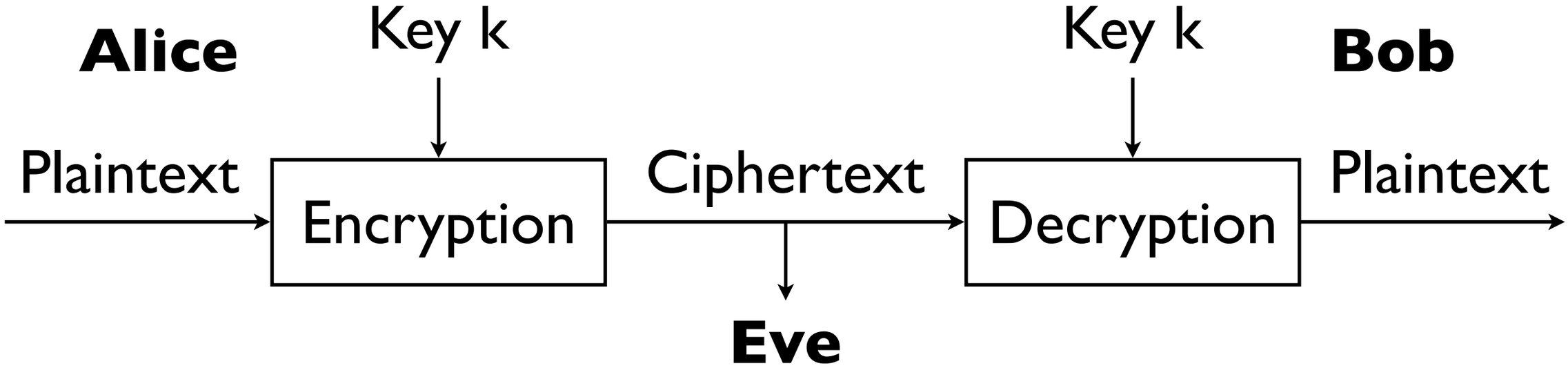
A collision-free hash function is a one-way function for which no two distinct values x and y with f(x) = f(y) can be found

3.2. Cryptographic Primitives

Pseudorandom Functions

- Random values needed for the generation of crypto. keys
- A random number generator generates random numbers
- Difficult to get true randomness in deterministic systems: Measure some physical phenomenon assumed as random or use an algorithm producing apparently random results
- Pseudorandom number generators initialized with a seed
- Seed must not be leaked in cryptographic applications
- It is hard to distinguish the output of a pseudorandom function from the output of a normal random function

3.3. Symmetric-key Crypto.



- Both sender and receiver share same key for encryption
- Only encryption method publicly known before June 1976
- Given any message m: $Decryption_k(Encryption_k(m)) = m$
- Add randomization: $E'_k(m) = E_k(r \,||\, m)$ (|| is concatenation)
  $\Rightarrow$ Prevents recognition of identical messages (+ no replay)

Perfectly Secure Encryption

- The so-called one-time pad encryption is perfectly secure
- Not practical in most contexts since key as long as message
- *Exclusive or* operator ⊕:"One or the other but not both"
  ⟹ Flips the bit if input is 1 and leaves the bit if input is 0
- Cipher given by $E_k(m) = m \oplus k = c$ and $D_k(c) = c \oplus k$
- Since k is chosen at random, ciphertext c is also random
  ⟹ Ciphertext c contains no information about message m
- Only perfectly secure if key used only once (*one-time* pad)
- However, easily malleable if no additional measures taken

3.3. Symmetric-key Cryptography

Block and Stream Ciphers

- A cipher is a pair of encryption and decryption algorithms
- Block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size
- Various possibilities for how to combine successive blocks
- Examples: Data Encryption Standard (DES) and Advanced Encryption Standard (AES) designated by US government
- Increase key size by encrypting several times, e.g. Triple DES
- Stream ciphers create an arbitrarily long stream of key material that is combined with plaintext like one-time pad

3.3. Symmetric-key Cryptography

Message Authentication Code

- Provide not only confidentiality of data but also integrity
- Message authenication codes (MACs) authenticate messages
- MAC algorithms take secret key + message and append a tag
- They work like hash functions with key: E.g. t = hash(k || m)
- Receiver does the same computation and compares the tags
- Alternatively, just encrypt hash of message with the message
- Contrary to digital signatures, no accountability with MACs (because the receiver can compute the same tag themself)

3.4. Public-key Cryptography

- Main problem with symmetric-key cryptography is the key management: Each pair of parties must share a different key or rely on a trusted third party to distribute one on demand
- Public-key (or asymmetric-key) crypto. proposed in 1976: Based on a pair of related keys (private key and public key), where it is infeasible to compute the former from the latter
- The public key, used for encryption, can be freely distributed, the private key, used for decryption, must remain secret
- Public-key crypto. assumes existence of trapdoor one-way functions: Hard to invert unless secret information is known

3.4. Public-key Cryptography

Digital Signatures

- Two algorithms for digital signatures: Signing and verification
- Signatures are easy to produce but hard for others to forge
- Digital signature: Tied to the content of the signed message
- Handwritten signature: Same for every file and easy to forge
- Use the private key for signing, the public key for verifying
- Public-key algorithms are computationally more expensive: Only sign hash of message and only encrypt symmetric key
- The autograph signature and the qualified digital signature are treated as equal in CH (Art. 14 Abs. 2bis OR) (SuisseID)

Public Key Infrastructure (PKI)

- Not possible to authenticate someone you have never met – unless you trust a third party to confirm their identity
- Introduce certificate authorities (CAs) that bind public keys with the identities of registered users by signing certificates
- Users authenticate themselves by presenting a certificate & by showing that they possess the corresponding private key
- Certificates have a scope and an expiration date, and since they can be revoked, trusted timestamping facility required
- Binding users to their public key ensures non-repudiation
- Advantage of public-key c.: CA can be offline after issuance

3.5. Group Theory

- A group is a structure consisting of a set and an operation that combines any two of its elements to form a third one
- To qualify as a group, set and operation must satisfy a few conditions called group axioms (stated on the next slide)
- The goal of algebra is to understand the properties of such structures at the highest level of generality and abstraction
- Abstraction: Eliminate unnecessary details for simplification

Quantification to make statements about elements of a set:

- Universal quantifier ∀ ("for all"): ∀ elements ∈ set: stmt.
- Existential quantifier ∃ ("it exists"): ∃ element ∈ set: stmt.

Group Axioms

A group is a set G and a binary operation • that satisfy:

- Closure: $\forall\ a, b \in G: a \bullet b \in G$
- Associativity: $\forall\ a, b, c \in G: (a \bullet b) \bullet c = a \bullet (b \bullet c)$
- Identity: $\exists\ e \in G\ \forall\ a \in G: e \bullet a = a \bullet e = a$
- Invertibility: $\forall\ a \in G\ \exists\ b \in G: a \bullet b = b \bullet a = e$

Due to associativity, parentheses are usually omitted

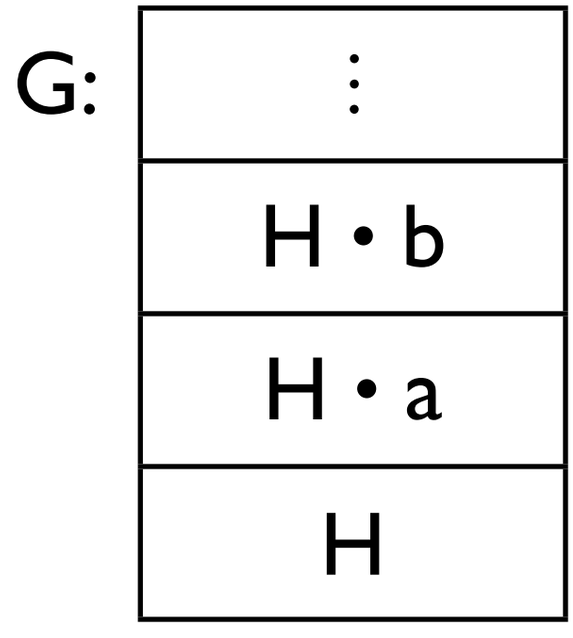
There exists only one neutral element: $e = e \bullet e' = e'$

A group is called commutative if also $\forall\ a, b \in G: a \bullet b = b \bullet a$

Element and Group Order

- For convenience, we use multiplicative notation for groups: We denote the operation as "•" and the inverse of a as a^{-1}
- Similarly, $a^m \cdot a^n = a^{m+n}$ and $(a^m)^n = a^{mn}$, therefore $a^0 = e$
- The number of elements is called the order of the group
- The order of an element a is the least $m \geq 1$ with $a^m = e$
- In a finite group, every element has a finite order "ord(a)"
- A subgroup is a group H contained within a bigger one G so that the neutral element of G is contained in H and whenever h_1 and h_2 are in H, then so are $h_1 \cdot h_2$ and h_1^{-1}

Lagrange's Theorem



«For any finite group G, the order of every subgroup H of G divides the order of G»

Proof sketch:

- Set $H \cdot a \stackrel{\text{def}}{=} \{h \cdot a \mid h \in H\}$ for $a \in G$ is called *right coset* of H
- The cosets of H form a partition of G: Union of all cosets is equal to G and two cosets are either equal or disjoint
- Two cosets $H \cdot a$ and $H \cdot b$ are equal iff $\exists\, h \in H: b = h \cdot a$
- All cosets have the same "order" $\Longrightarrow$ each has $|H|$ elements
- Define bijective mapping $f: H \cdot a \rightarrow H \cdot b$ as $f(x) = x \cdot a^{-1} \cdot b$

Modular Arithmetic

- For a positive integer n, two integers a and b are said to be congruent modulo n (written $a \equiv_n b$) if a - b is multiple of n
- If $a \equiv_n b$ and $a, b \geq 0$, then a/n and a/b have same remainder
- This relation partitions integers into congruence classes: $a_1 \equiv_n b_1$ and $a_2 \equiv_n b_2 \Longrightarrow a_1 + a_2 \equiv_n b_1 + b_2$ and $a_1 a_2 \equiv_n b_1 b_2$

Examples of groups:

- Integers modulo 12 with addition (like analog clock), e = 0
- Integers modulo 5 with multiplication but w/o multiples of 5

Euler's Totient Function

- Having 2 sandglasses of different durations, the measurable times are multiples of their greatest common divisor (GCD)
- In case of multiplicative groups modulo n, only integers that are coprime to n (their GCD is 1) have an inverse element
- The totient $\varphi(n)$ of a positive integer n is the number of positive integers less than n that are coprime to n (Euler)

Cryptography is only interested in two cases:

- Prime p: $\varphi(p) = p - 1$ (positive integers smaller p except 0)
- Product of primes p and q: $\varphi(p \cdot q) = (p - 1)(q - 1)$ (count)

Euler's Theorem

Euler's Theorem: $a^{\varphi(n)} \equiv_n 1$ (for coprime integers n and a)

Proof (application of group theory):

- Treat a as an element of the multiplicative group modulo n
- Subgroup $\langle a \rangle = \{e, a, a^2, \ldots, a^{ord(a)-1}\}$ of G has order ord(a)
- By Lagrange's theorem: ord(a) divides |G| for every $a \in G$
- Consequently, $a^{|G|} = e$ for every $a \in G$ and finite group G
- Number of elements in multiplicative group modulo n: $\varphi(n)$

3.6. Cryptosystems

In general, cryptosystems consist of three algorithms:

- Key generation: Generates a random pair of keys
- Encryption: Transforms the plaintext into ciphertext
- Decryption: Recovers the plaintext from the ciphertext

These algorithms have to be efficient for the legitimate parties (polynomial time in terms of key size) but hard for any computationally limited adversary (exponential)

Given this asymmetry, increasing the key size compensates for any performance improvements of modern computers

Diffie-Hellman Key Exchange

Alice	insecure channel	**Bob**
selects $a \in_R \{0, \ldots, p - 2\}$ (subscript R: at random)		selects $b \in_R \{0, \ldots, p - 2\}$ (p is a prime number)
$A \equiv_p g^a$	$\xrightarrow{A}$	$B \equiv_p g^b$
$k_{AB} \equiv_p B^a$	$\xleftarrow{B}$	$k_{BA} \equiv_p A^b$

$$k_{AB} \equiv_p B^a \equiv_p (g^b)^a \equiv_p (g^a)^b \equiv_p A^b \equiv_p k_{BA}$$

$k_{AB} \equiv_p k_{BA}$ used to derive a key for symmetric encryption

Discrete logarithm assumption: Given p, g and y with $y \equiv_p g^x$, computing x is computationally infeasible (one-way function)

Remark: g^x modulo p can be computed efficiently with *exponentiation by squaring*

ElGamal Encryption

Key generation:

- Bob chooses a multipl. group G of order q with generator g
- Bob chooses b $\in_R$ {0, …, q - 1} and computes B = g^b
- Bob publishes G, q, g and B as public key, retains b private

Encryption:

- Alice chooses a $\in_R$ {0, …, q - 1} and computes A = g^a
- Alice computes ciphertext as (A, M) = (g^a, m • B^a)

Decryption:

- Bob computes m = M • A^{-b} = m • B^a • $(g^a)^{-b}$ = m • $(g^b)^a$ • $(g^b)^{-a}$

Extended Euclidean Algorithm

How to find the multiplicative inverse of integer a modulo n?

- The extended Euclidean algorithm finds for input a and b two integers x and y that satisfy $ax + by = \gcd(a, b)$
- Remember: Multiplicative inverse only exists if $\gcd(a, n) = 1$
- Taking $b = n$ gives us $ax + ny = 1 \Rightarrow ax \equiv_n 1$ and $x \equiv_n a^{-1}$

Step	Quotient	Remainder	Substitute	Combine terms
1		**120 = a**		= 120 • 1 + 23 • 0
2		**23 = b**		= 120 • 0 + 23 • 1
3	5	**5** = 120 - 23 • 5	= (120 • 1 + 23 • 0) - (120 • 0 + 23 • 1) • 5	= 120 • 1 + 23 • -5
4	4	**3** = 23 - 5 • 4	= (120 • 0 + 23 • 1) - (120 • 1 + 23 • -5) • 4	= 120 • -4 + 23 • 21
5	1	**2** = 5 - 3 • 1	= (120 • 1 + 23 • -5) - (120 • -4 + 23 • 21) • 1	= 120 • 5 + 23 • -26
6	1	**1** = 3 - 2 • 1	= (120 • -4 + 23 • 21) - (120 • 5 + 23 • -26) • 1	**= 120 • -9 + 23 • 47**
7	2	0	End of algorithm	→ x = -9 and y = 47

RSA Algorithm

$\longrightarrow$ hard
$\longleftarrow$ easy

Key generation:

- Choose two random primes p and q, compute $n = p \cdot q$
- Compute $\varphi(n) = (p - 1)(q - 1)$, choose e w. $\gcd(e, \varphi(n)) = 1$
- Determine the multiplicative inverse of e: $d \equiv_{\varphi(n)} e^{-1}$ (Euclid)
- Publish modulus n and exponent e, retain d as private key

Encryption/Verification:

- Compute ciphertext $c \equiv_n m^e$ (with m being group element)

Decryption/Signing:

$d \cdot e - k \cdot \varphi(n) = 1$ (Euler's theorem)

- Compute $m \equiv_n c^d \equiv_n (m^e)^d \overset{\downarrow}{\equiv_n} m^{1+k \cdot \varphi(n)} \equiv_n m(m^{\varphi(n)})^k \overset{\downarrow}{\equiv_n} m \cdot 1^k$

3. Cryptography

Concepts Learned Today

- Abstraction
- Authentication
- Authorization
- Confidentiality
- Digital signatures
- Encryption and decryption
- Integrity
- Public-key cryptography
- Symmetric-key cryptography

3. Cryptography

Clip of Today

Medieval helpdesk with English subtitles (2:46)

My favorite comment:
The bookmarks will come in version 2.0.

http://www.youtube.com/watch?v=pQHX-SjgQvQ