Fernando Trujant

Massachusetts Institute of Technology
6.857: Network and Computer Security
Professor Ronald L. Rivest

Handout 1
February 2, 2016
Due: N/A

6.857 Course Information (Spring 2016)

Lecturer:	Professor Ronald L. Rivest 32-G692, 253–5880, `rivest@mit.edu` Office Hours by appointment

Teaching Assistants:	Cheng Chen
	`chengch@mit.edu`
	Conner Fromknecht
	`conner@mit.edu`
	Kevin King
	`kcking@mit.edu`
	Evangelos Taratoris
	`evtara@mit.edu`

Office Hours:	Fridays, 1 PM - 5 PM Location: 24-316
Course Secretary:	Debbie Lehto 32-G675A, 253-6098, `rivest-assistant@csail.mit.edu`
Staff Email:	`6.857-staff@mit.edu`
TA Email:	`6.857-tas@mit.edu`

1 Prerequisites

The prerequisites for the course are 6.033 (*Computer System Engineering*) and 6.042J (*Mathematics for Computer Science*). It is recommended that students have had 6.006 or 6.046J (*Introduction to Algorithms*) and experience with modular arithmetic.

You must have *completed* 6.042 in order to register for 6.857 this year. Taking 6.042 concurrently is not enough. If you have successfully completed 18.310, 6.045, 6.046, or 6.875, or if the department has given approval for 6.042-equivalency for some other course or program (perhaps taken elsewhere) our prerequisite requirement for taking 6.042 is satisfied.

You must have successfully completed 6.033 already, or be taking it concurrently with 6.857, or have departmental approval for some other course or program (perhaps taken elsewhere) for satisfying the 6.033 requirement, our prerequisite requirement for taking 6.033 is waived.

We may (rarely) make exceptions to the above. If you wish to be considered for an exception, send an email to `6.857-staff@mit.edu` with a description of your year, your reason for requesting an exception, and what equivalent background you may have had. Describe also how 6.857 fits into your educational program and career plans.

2 Units

6.857 is a 12-unit (3-0-9) H-level course intended primarily for seniors and first-year graduate students. It fits within the Computer Systems Concentration. Graduate students will receive H-credit for this class.

Homework templates will be available on the course web site. For homework involving non-trivial mathematics, students are *strongly* encouraged to use LaTeX to typeset their answers. Homework that is difficult for the graders to read will lose points.

We will use Gradescope for homework submission. Homework should be submitted in PDF format. Each problem will need to submitted as a separate file to facilitate the grading process. (The submission process will be further explained in the homework handout.)

Late homework will be penalized at the rate of two points (out of ten) per problem per day of lateness. For example, if the homework is due on Monday, but you turn it in on Wednesday, then your maximum score per problem is six. (Your net score will not be allowed to go negative, however.) You may turn in your solutions to different problems at different times to avoid penalties for problems you have completed before the deadline. You may turn in more than one solution to a problem, but only the latest one will count, except that once the homework deadline has passed, you may not turn in a solution to a problem for which you have already turned in a solution. Because we allow late solutions, please do not talk with anyone outside of your group or the TAs until the Friday of the week in which the homework is due.

Solutions will be distributed with corrected homework—hopefully within a week of being collected.

Generally, homework must be done in groups (although we reserve the right to require individual homework assignments). You are to work on group problem sets and final projects in groups of (preferably) three or four. One problem set will be turned in by each group, and one grade will be given for each problem set. You *must* work in groups; homeworks turned in by individuals, pairs, pentuples, etc. will not be accepted. Be sure that *you* understand and approve the solutions turned in to *each* problem. As noted above, the initial organization into groups for the first two problem sets will be established by the staff, but you may organize your own groups for the later homeworks and for the final project.

We may occasionally assign homework that you must answer individually; see Section 11 for the policy governing these assignments.

8 Tests

We will have one in-class quiz on Wednesday, April 13, 2016. The quiz will test your knowledge of material from lectures, problem sets, and readings.

There is *no* final exam.

9 Final project

Start early. Specially if you need permission

Students will be responsible for a final project. You must work in a group of three or four people. The nature and the topic of the project is your choice, although it needs the approval of the teaching staff. See the *Term Projects* page on the course web site for a list of topics from previous years, sample proposals, and additional project-related resources. We will generally approve interesting topics about cryptography, network security, and/or computer security.

It is advisable to get started early; we will gladly accept proposals before the deadline. Early submission gives us a chance to review and approve your project proposal, and to suggest references that you may have overlooked.

Important dates for the project (subject to change):

- By Monday, March 7 - Every student must individually post one (or more) project ideas on Piazza. Each post should have a heading with the topic area. This is a way for students to learn about what other students are interested in and find teammates. If you have more than one idea or interest, feel free to post all of your ideas, but please use different posts with different hearders. submit a one-page project idea via e-mail to `6.857-staff@mit.edu`. Your ideas can be from the project ideas we post or

Nevertheless, unless you have explicit written authorization from the owner and operators of a computer network or system, you should never attempt to penetrate that system or adversely affect that system's operation. Such actions are a violation of MIT policy and, in some cases, violations of State and Federal law. Likewise, you should refrain from writing computer viruses, worms, self-reproducing code, or other kinds of potentially damaging software for this course unless you have explicit, written approval for the specific type of software that you wish to create. These kinds of programs are notoriously difficult to control and their release (intentional or otherwise) can result in substantial civil and criminal penalties.

In particular, term projects involving an evaluation of security of existing commercial products or systems need the approval of the course staff, who may require that you obtain permission from the vendor/supplier (depending on the nature of your proposed evaluation).

We strongly recommend that you consult the *Athena Rules of Use* at

`http://ist.mit.edu/services/athena/olh/rules`

and Section 13.2 of the MIT Policies and Procedures "Policy on the Use of Information Technology" at

`http://web.mit.edu/policies/13.2.html`.

Finally, we recommend that you read and review the *ACM Code of Ethics and Professional Conduct* which can be found online at

`http://www.acm.org/constitution/code.html`.

(Or Google for "acm ethics".)

We expect all students in this class to follow the guidelines presented in this document, and in the documents just cited. If you are in doubt about the legality or ethics of any activity related to this course, please consult the staff before undertaking any such activity.

2/3/15

6.857 Network and Computer Security

Fernando Trujo

Security = communicating or computing in precense of adversaries
↑ Have to be able to think like

Security Policy (Goals): principals (parties that take actions), roles, permissable and impermissable actions
classes of objects.

CIA: confidentiality, integrity, availability

Security Mechanisms (controls)
↳ Means by which one attempts to achieve security policy
Prevention Mechanisms - encryption, fences, lock
Detection " - motion sensors, => then what ? → Recovery Mechanism
Recovery : Removing virus, etc ↳ Deterrance

Who could adversary be ? - insider → (vendor, admin, developer)
- outsider

know ? - same code / design
- password
→ assume that some security mechanisms are already broken.

Security in Depth

What reasources does a have? - computation power
- ability to control network (countries/militaries)
- time / patience / persistance
↳ APT - Advanced persistant Threat
↳ worst threat.
Every system is vulnerable

Vulnerability: weakness that allows adversary to defeat security policy

Mechanism
Identification: way to identify various parties
Authentication:
Authorization: What access authenticated user has
Physical Protection:
* Cryptography:

1/1

2/5/16

6.857 Recitation #1

Fernando Trujo

Finite Fields
- $\mathbb{Z}_p$
- $GF(4)$
- generators

A Finite Field is a system $(S, +, \cdot)$

- S is a finite set that contains elements $0, 1$

$(S, +)$ is an abelian group with identity 0

- If $a, b \in S \rightarrow a + b \in S$
- For $a, b \in S$: $a + b = b + a$
- $\forall a \in S$: $a + 0 = a$ ← so 0 is the identity
- $\forall a \in S \ \exists \ b \in S$: $a + b = 0$ ← b is the additive inverse of a $b = -a$
- $\forall a, b, c \in S$: $(a + b) + c = a + (b + c)$
- $S^* = S - \{0\}$
- ~~$(S^*, \cdot)$ is an abel~~

$(S^*, \cdot)$ is an abelian group w/ identity 1 ← 0 does not have multiplicative inverse

- $\forall a, b \in S^*$: $a \cdot b \in S^*$

$\forall a, b \in S^*$: $a \cdot b = b \cdot a$

$\forall a \in S^*$ $a \cdot 1 = a$

$\forall a \in S^* \ \exists \ b \in S$: $a \cdot b = 1$ ← b is multiplicative inverse $b = a^{-1}$

$\forall a, b, c \in S^*$: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

Distributive Law: $\forall a, b, c \in S$: $a \cdot (b + c) = a \cdot b + a \cdot c$

$(a + b) \cdot c = a \cdot c + b \cdot c$

$\mathbb{Z}_p$ = the integers mod p = $\{0, 1, 2, \ldots, p-1\}$

Can still solve linear equations in $\mathbb{Z}_p$

$ax + b = 0 \pmod p$ $\quad a, b \in \mathbb{Z}_p$

$ax = -b \pmod p$ - additive inverse of b

$x = a^{-1} \cdot -b \pmod p$ - multiplicative inverse of a

Ex: $3x + 5 = 6 \pmod 7$

$3x = 1 \pmod 7$

$x = 5 \pmod 7$

Finding multiplicative inverse

$\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$

mult inverse of 3?

$3 \times 0 = 0 \pmod 7 \neq 1$ $\quad$ $3 \cdot 3 = 9 \pmod 7 \neq 1 \pmod 7$

$3 \cdot 1 = 3 \pmod 7 \neq 1$ $\quad$ $3 \cdot 5 = 15 \pmod 7 = 1 \pmod 7$

$3 \cdot 2 = 6 \pmod 7 \neq 1$ $\quad$ ↑ so 5 is mult inverse

Notation: $GF(q)$ is the finite field

Thm: $\forall$ prime numbers p, $\forall$ positive integers n

$\exists$ a Finite Field with p^n elements $GF(p^n)$

What are the elements of $GF(4)$ ie $GF(2^2)$

$GF(4) = \{0, 1, x, x+1\}$

· Addition: $(x+1) + 1 = x + 0 = x$ $\quad$ ($1+1 \pmod 2$, $1+1 \to 0$)

Ex: $a, b \in GF(2^8)$

$a: x^6 + x^3 + x^2 + 1$

$b: x^5 + x^3 + x + 1$

$$\begin{array}{r} a \quad x^6 + x^3 + x^2 + 1 \\ b + \quad x^5 + x^3 + x + 1 \\ \hline x^6\ x^5\ 0\ 0\ x^2\ x\ 0 \end{array}$$

· Multiplication

Multiplication happens modulo $x^2 + x + 1$

- $x, 1 \to x \cdot 1 = x$
- $x, x+1 \to x^2 + x \pmod{x^2+x+1} = -1 = 1$ ($\pmod 2$)

↑ not in $GF(4)$ $\quad$ ↑ In $GF(4)$

- $x \cdot x \to x^2 \pmod{x^2+x+1} = -x - 1 = x + 1$

Properties

In a Finite field with K elements: $\underbrace{1+1+\ldots+1}_{k \text{ times}} = 0$

$\forall a$ $\quad \underbrace{a+a+\ldots+a}_{\pi} = 0$

Thm: Fermat's Little Theorem

In $GF(q)$: for all $a \in GF(q)^*$

$a^{q-1} = 1$

↑ Identity

- $a^{q-1} = 1 \rightarrow a^q = a \quad \forall\, a \in GF(q)$
- $\forall\, a \in GF(q)^* : a^{-1} = a^{q-2}$

Example:

Find inverse of $3 \pmod 7$

$= 3^5 \pmod 7 = 5 \pmod 7$

Proof of Fermats Little Theorem (in $GF(q)$ $q = p^3$)

Lemma 1: IF $a^m = a^n$ and $m > n$ then $a^{m-n} = 1$

Lemma 2: $(a^n)^{-1} = (a^{-1})^n$

$a^m = a^n$

$a^m \cdot (a^n)^{-1} = a^n \cdot (a^n)^{-1} = 1$

$\frac{a^{m-n} \cdot a^n}{}$

$a^m (a^{-1})^n = 1$

$a^{m-n} \cdot a^n (a^{-1})^n = 1$

$a^{m-n} (a^m (a^{-1})^n) = 1 \implies a^{m-1} \cdot 1 = 1 \rightarrow a^{m-n} = 1$

Lemma 3:

$\forall\, a \in GF(q)^* \;\exists\; m, n \quad (m > n)$

s.t $a^m = a^n$

$\rightarrow a^{m-n} = 1$

Lemma 4: $\forall\, a\ GF(q)^* \;\exists\; N > 0 : a^N = 1$

Lemma 5: Let N_m be the minimum positive integer s.t $a^{N_m} = 1$ then we claim that N_m divides q

Proof by contradiction: We can write $q - 1 = a \cdot N_m + b$

$0 \le b \le N_m$

⋮

2/8/16

6.857 Lecture #2

Fernando Trujar

Psets + groups will be up today.

.ioms for class

- Think adversarially
- Be sceptical / paranoid
- Attacker only has to find one, defender has to protect all
- "Assume breach"
- Don't underestimate time/effort attacker will put. ATP = Advanced persistant thre
- Separation of priviladge
- Complete mediation
- Education and training → people can mess up
- Sharing info about vulnerabilities
- computers + systems as toys → they will break
- Adversaries: Insiders, nation-states

- Don't aim for perfection (does not exist)
- Tradeoff: cost vs security
- Defense in depth → Layer in defense
- Least privilage → don't give more permission that they need
- Transparency - security through obscurity
  ↳ weakest form of protection

The growth of Cryptography

Fermat Little Theorem: $a^{p-1} = 1 \pmod{p}$ (p: prime)

Euler's Theorem: $a^{\Phi(n)} = 1 \pmod{n}$

$\Phi(n)$: # $x < n$ s/t $\gcd(x, n) = 1$

See research problems slide for project ideas!

2/10/16

6.857 Lecture #3

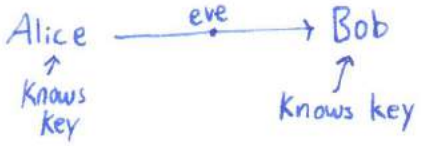
Fernando Trujano

Pset #1 due 2/22

Encryption and one-time-pad

Reading: Katz and Lindell Ch 1, 2, 3, 5

Confidentiality



Eve doesn't know key

Eve can hear cyphertext

↳ may learn length of msg.

Generating a random key:

- coin-flipping - nature - quantum - instruction on CPU

$K \leftarrow Gen(1^\lambda)$ ← length of key

1^λ = 111.... 1 (λ ones)

$C \leftarrow Enc(K, M)$ ← message

↳ could be randomized

$M = Dec(K, C)$

Eve knows: CT

CT/PT pairs

choose PT

choose CT

Eve can't distinguish b/w $Enc(K, M_1)$ and $Enc(K, M_2)$ if she can choose M_1 and M_2 (same length).

↳ Cipher Text indistinguishability / Semantic security

One Time Pad - Veram 1917

Message, cipher text, key all have same length.

Key called pad.

Enc:

$$\begin{array}{r} M = 101100\ldots \\ +\ K = 011010\ldots \\ \hline C = 110110 \end{array}$$

XOR / addition mod 2

$C = M \oplus K$

$C \oplus K = (M \oplus K) \oplus K$

$= M \oplus (K \oplus K)$

$= M \oplus 0$

$= M$

(OTP)

One time pad is unconditionally secure.

↳ no need to make assumptions about computing power

$P(M)$

↑ Prior probability.

P that eve thinks that M will be sent

$P(M|C)$ = Eve's posterior probability after having heard ciphertext

Thm: $P(M) = P(M|C)$ for OTP ≡ Eve learns nothing by hearing secret.

Assume $|M| = |C| = |K| = \lambda$

$P(K) = 2^{-\lambda}$

↳ Prob that particular k gets used.

Lemma: $P(C|M) = 2^{-\lambda}$

$= $ Prob of C, given M

$= $ Prob $K = C \oplus M$

$= 2^{-\lambda}$

$P(C) =$ prob of ciphertext C

$= \sum_m P(M) \cdot P(C|M)$

$= \sum_m P(M)\, 2^{-\lambda}$

$= 2^{-\lambda} \underbrace{\sum_m P(M)}_{=1} = 2^{-\lambda}$

Each message "smeared" uniformly over ciphertext.

Each ciphertext is equally likely.

Bayes Rule

$$P(M|C) = \frac{P(C|M)\,P(M)}{P(C)}$$

$$= \frac{2^{-\lambda} P(M)}{2^{-\lambda}} = P(M)$$

∴ Eve learns nothing from seeing the cipher text.

□

Limitations of ONE time Pad:

Sending two messages with the same key. => Eve hears both!

$C_1 = M_1 \oplus K$

$C_2 = M_2 \oplus K$

Eve can XOR both ciphertexts:

$$C_1 \oplus C_2 = (M_1 \oplus K) \oplus (M_2 \oplus K)$$
$$= M_1 \oplus M_2 \oplus (K \oplus K) = M_1 \oplus M_2$$

can "guess" M_1 and M_2 From here.

Fun read:
Project Venona

2/12/16

6.857 Recitation #2

Fernando Trujano

Hashing

Definition: A hash function H maps a universe $U \subseteq \{0,1\}^*$ to some finite set $S \subseteq \{0,1\}^\lambda$

Desirable Properties

One-way: H is one way if for $x \xleftarrow{\$} U$ (uniformly at random), given $y = H(x)$ it is infeasable to find x' s.t. $H(x') = y$

(non-invertable)

Collision Resistance:

Infeasable to find $x \neq x'$ s.t. $H(x) = H(x')$

Attacking CR → Finding collisions

- Birthday Attack
  - Hash random x_i and store $(x_i, H(x_i))$ in hash table until you find collision, or run out of resources.
  - relatively high constant prob of success in $O(\sqrt{|S|})$ time/memory. ($|S| = 2^\lambda$)
- Floyd's Cycle Detection
  - Hash $H(x)$ output until cycle/collision detected

2 pts a, b
set $a = b = x \xleftarrow{\$} U$
until $a = b$
  set $a = H(a)$
  set $b = H(H(b))$
Found cycle.

$a = x_{(i-t)} \bmod n$

$b = x_{2(i-t)} \bmod n$

Complexity: $\max(t + (-t \bmod n), n)$ iterations
3 hashes per iteration.
Memory: $O(1)$ (4 pointers)

After Floyd's.
set $a = x = x_t$
$b = x_{-t \bmod n}$ ← where Floyd terminated
Keep track of last value for a, b = a', b'
Progress a, b one step at a time until they mee
return (a', b')

- Could let Floyd running for a very long time → won't run out of memory (unlike birthday attack)

Inverting H - Rainbow Table

$x \xleftarrow{\$} U$ given $y = H(x)$ want x' s.t $H(x') = y$

- memory time tradeoff using hashchains of length K.

Algorithm: want to attack n hashes

Precomp:
- pick n/k random pre-images, x_i
- store $(x_i, H^{(k)}(x_i))$ hash k times (chaining)

Query for hash y

for i in (0, k): check for $H^{(i)}(y)$ in chain tails

if chain exists, then ~~start~~ we know $y = H^{(j)}(x_i)$ $0 \le j \le k$

Problem: only works on pre-images that are outputs of H

Fix : define a reduction function $R: S \to P$ where P is a set of likely pre images.

Rainbow table with reduction function

let $C: R \circ H$; $C(x) = R(H(x))$

Precomp: store $(x_i, C^{(k)}(x_i))$

Query hash y,

check for $C^{(i)}(R(y))$ for i in $0 \ldots k$

(note) : false positives (collision in R) possible.

2/12/16 6.857 NSA Talk Notes Fernando Trujano

To protect your network → know your network
know technologies in use ★

- Reduce the attack surface → remove what you are not using.

RED Team testing to find misconfigured network.
↳ Penetration testing

+ Listen and follow through with security report. Fix known bugs ASAP.
+ Don't assume a crack is too small
APT - Advance Persistant threats will find them

Invest in continious protection / defense.
Initial Vectors (typical) CBE

1) Email → Anti-explotation features
2) Website
3) Removable media

Network should not trust everything it's wels are trying to do.

→ Teach and train best practices to people.

Understand what is normal for a user to do in a network → don't let lost credential destroy network
- monitor credential uses and catch abnormal use.
→ Least priviladge - don't give ppl access to things they don't need
Segmenting ~~purpuse~~ network.

Pass the Hash vulnerability - learn and protect against it
- Don't hardcore things into scripts.
- Enable and look at logs.

1/2

Application whitelisting

- Reputation services ie. only allow reputable domain names.
- Force 2FA
- Limit ways to move laterally once inside network → Assume breach
- Dynamic privilages → different access based on location.

Differentiate b/w cybercriminals vs nation/state threats

www.nsa.gov

2/16/16

6.857 Lecture

Fernando Troja

Cryptographic Hash Functions

Cryptographic Hash Function : maps some $D = \{0,1\}^*$ (all possible bit strings) to range $R = \{0,1\}^d$

$h: \{0,1\}^* \to \{0,1\}^d$

in an efficient, deterministic, public, "random" manner

Ex: MD5 d = 128
SHA-256 d = 256
SHA3-256 d = 256

VIL = variable input length
FIL : Fixed input length

VOL : Variable output length
FOL : Fixed "

Random Oracle Model ROM

Oracle (in sky)

receives input x, returns h(x)
for any $x \in \{0,1\}^*$ $|h(x)| = d$ bits

Oracle has a book of previous answer
if x not in book:
Flip coin d times to determine h(x)
record (x, h(x)) in book
return y where (x, y) in book

Desired Properties

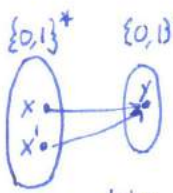
① One-way (OW) - pre-image resistance

"Infeasable" given y ($x \in_R \{0,1\}^*$, $y = h(x)$)
to find x' s.t. $h(x') = y$
(x': pre image of y)

Security level determined by d.

In ROM, expect to need 2^d to find pre-image.

② Collision-resistance



Like birthday paradox

Infeasable to find x, x' s.t. $x \neq x'$ and $h(x) = h(x')$

Claim: In ROM, need to check $2^{d/2}$ values of x before we find 2 that colide.

If we hash $x_1, x_2 \ldots x_n$ (distinct)

$$E(\#\text{collisions}) = \sum_{\substack{i,j \\ i \neq j \\ 1 \leq i,j \leq n}} P(h(x_i) = h(x_j)) = \binom{n}{2} \cdot 2^{-d} \approx \frac{n^2}{2} 2^{-d}$$

(# pairs)

≥ 1 when $n \geq 2^{(d+1)/2} \approx 2^{d/2} = \sqrt{2^d}$ □

③ Target Collision Resistance (TCR) "weak CR"

Infeasable given $x = \{0,1\}^*$ to find $x' \neq x$ s.t $h(x) = h(x')$

- Like CR, but x is already chosen.

Time in ROM is expected $O(2^d)$

CR → TCR
Implies

④ Pseudo-Randomness (PR)

Indistinguishable from random oracle.

⑤ Non-malleability:

Controlled change in input should not lead to desired change in output.
Infeasable given $h(x)$ to produce x' where $h(x')$ and $h(x)$ are related in some predefined way. (close)

Applications

Password storage: store hash of password instead of plaintext. OW
disclosure of h(pw) doesn't reveal pw.

File Modification - store h(F) (digest) and transmit $F \Rightarrow F'$ check that $h(F') = h(F)$ TCR
(checksum)

Digital Signature: PK_A = public key
SK_A = secret key

$Sign(SK_A, M) \rightarrow \sigma$
$Verify(M, \sigma, PK_A) \rightarrow \{True, False\}$ TCR CR

Hash + Sign: $Sign(SK_A, h(M)) \rightarrow \sigma$
verify-recompute h(m) on

Commitments: Alice has x. computes C(x) subtits C(x) as her bid. binding → CR, secrecy → OW
Auction asks alice to reveal (x) => can be verified with C(x)
Binding: Alice can only open C(x) to reveal x. Secret: no one but A can learn from C(x

2/17/16

6.857 Lecture

Fernando Trujano

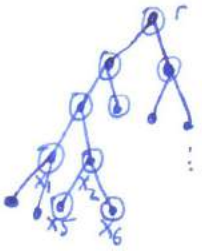
Cryptographic Hash Functions II - Merkle

Merkle Trees

- Validate membership in a set.

set $\{x_1, x_2 \ldots x_n\}$

value at $x = h(y \,||\, z)$ (|| = concatenated)



- Given the root r of the tree → how to verify x_5 is in tree
  - need to know all ancestors and "uncles" of x_5.

↳ Size of proof that $x_5 \in$ tree rooted at r is $\lg(n)$

Protected by CR of Hash

Puzzles

$h: \{0,1\}^* \to \{0,1\}^d$

given $y = \{0,1\}^*$ to find x s.t $h(x) = y$ takes 2^d time on average. (2^d: controllable difficulty)

Choose small d (d=40) to make a puzzle (eg. truncate SHA-256)

variant hash function key k : $h_k(x) = h(k \,||\, x)$

Adam Back's "Hash Cash"

Anti-spam. Proof of work = "stamp" POW

PR

POW: $h(k \,||\, r)$ ends in 20 zeroes

k = sender || receiver || date + time

find r by brute force search.

include r in email header

Receiver can check that $h(k \,||\, r)$ ends in 20 zeroes.

Interesting iclea .... but never caught on.

- This puts the burden on the spammers but they might not be using their own computers to solve puzzles. ie botnets.

Public Key Cryptography

based on puzzles

- Secure communication without prior arrangements

Alice ——— (Eve) ——— Bob

Bob creates n puzzles, each of difficulty $D = 2^d$

$P_1 \ldots P_n$ => sends them to Alice. $P_i = (y_i, E_{x_i}(K_i))$

Alice solves a random one P_i => solution x_i $h(x_i) = y_i$ (K_i: session key)

Alice can send $h(K_i)$ to Bob.

Further communication encrypted with K_i

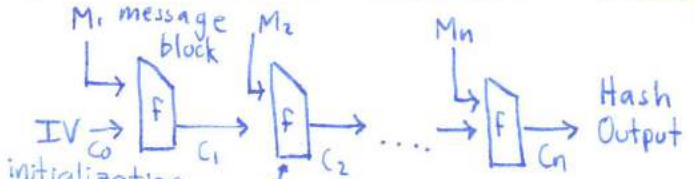
+ Bobs work = $O(n)$
  Alice work = $O(D)$
  Good guy work = $O(D+n)$

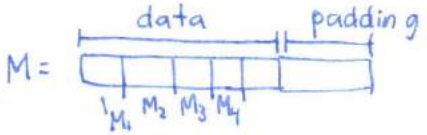
Eve's work = $O(\frac{n}{2} \cdot D) \cdot O(n^2)$

How hash functions are built

MD4, MD5

- design for Fixed Length Inputs (FLI)
- expand to handle Variable " (VLI)

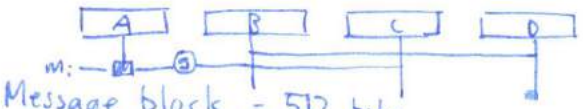




Thm: If f is CR so is h

MD5

128 bit state

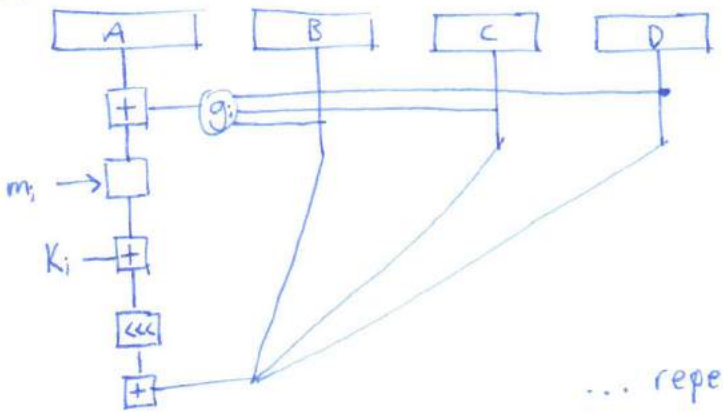


4 32 bit/registers

Primative Operations

- ⊞ Addition mod
- ⊕ XOR
- <<< rotate

1 round



... repeat 64 times

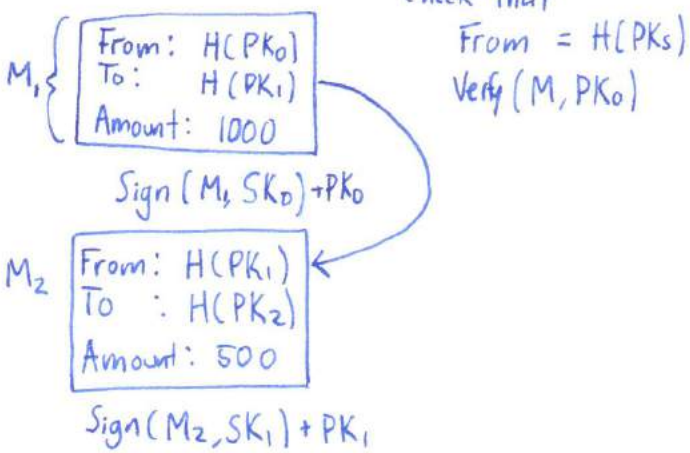
2/19/16

6.857 Recitation

Fernando Trujan

Bitcoin

Transaction -
- Sender, receiver, amount
- Addresses → Hash of a public key
- Chained - Provides History



Double spending

Problem: prevent spending of money twice

Timestamp Server:
- publishes hash of transactions at regular intervals
- Prove data existed @ point in time
  ⇒ partial ordering on transactions
- Hashes are chained, each one commits all prior hashes

New Problem: Centralized. Need to make this distributed

Proof of work

- Prove resource expenditure
- Large asymmetry between solving and verifying

Bitcoin PoW — Nonce, Difficulty

$H(data, n) < 2^{256-D}$

$H = SHA256(SHA256(data))$

$Pr[H(data, :) < 2^{256-D}] = 1/2^D$

Requires $\Theta(2^D)$ to solve, only $O(1)$ to verify.

- Self-adjusting D, moving average every 2 weeks

Blockchains

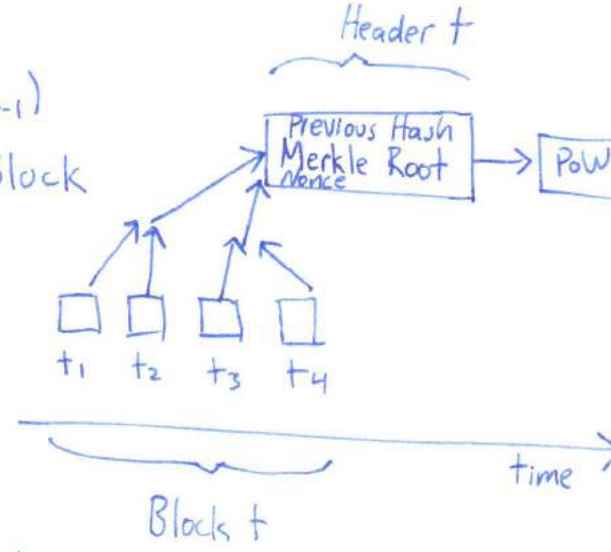
Block – list of transactions

Block Header -

Hash of previous header $H(\text{header}_{t-1})$

Merkle Root over transactions in Block

Nonce, h



- Blockchains provide append-only database
  - All chains extend genesis block ← starting point
  - Miners append to chain by computing PoW on block headers.
  - A block is valid iff PoW satisfies D and the transactions are valid

Mining Algorithm

Given Header_{t-1} ↙ includes PoW

1) ↳ Gather unprocessed transactions
2) Compute Merkle Root r of ↑
3) Nonce $n \xleftarrow{\$} \mathbb{Z}$ (random)
4) Assemble $\text{Header}_t = \{H(\text{header}_{t-1}), r, n\}$ ↓ found PoW
5) If $H(\text{Header}_t) < 2^{256-D}$, broadcast to peers and restart Header_t.
6) If fail: Increment n, go to 4).

Security

p- Probability that honest node finds block

q = " attacker "

q_z = " attacker will overtake z mainchain from z blocks behind.

$$q_z = \begin{cases} 1 & \text{if } p \le q \\ (q/p)^z & \text{if } p > q \end{cases}$$

← if attacker has more hashing power than honest

See notes for more formal proof

2/22/16

6.857 Lecture

Fernando Trujano

e-cash

money, e-check, signed coin ID, ledges, bitcoin

Money

- Exclusive ownership - Lost when given.

Digital currency - model the transfer. NO double spending

How value is stored:

Possesion-based or Account based

+ owning bits = owning value + maintain track of transactio

E-checks

check = order to the bank to transfer $$$ from 1 account to another

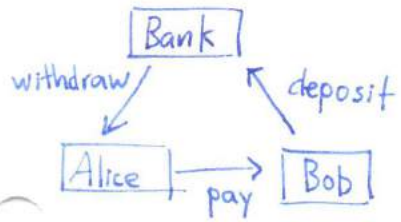
PK_B, SK_B ← Bank PK_U, SK_U ← User, certificate on (U, PK_U) by bank

check = cert + sign(SK_U, "Pay Bob $100, date, serial #")

Desired Properties

- Transfer value
- Non-forgeable
- Transitivity - can spend recieved coin.
- Anonimity
- Divisibility and Combineability
- Online vs offline transactions
- Scalability

Life of coin



Micro mint coin = k-way collision of hash fn (need to look $n^{(k-1)/k}$ by birthday parad

idea: Bank can afford computational power to mint. Others cant

hash into range of size n.

(x_1, x_2, x_3) distinct collission.

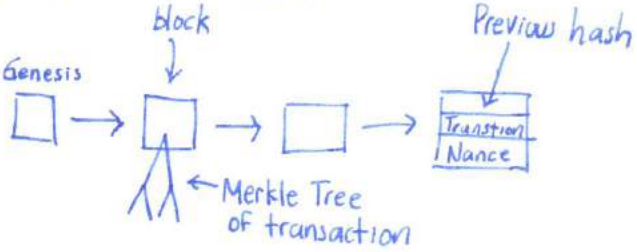
2-way colision: look at $n^{1/2}$ x's

~~k-way co~~

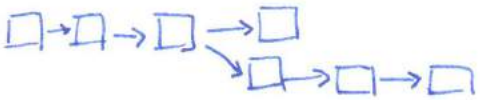
Bitcoin 2008 Satoshi Nakamoto

ID's are PK ← hashes of them ← addresses ← repositories for coins.

Public ledger records all transactions



* Does not prevent double spending



← only longer cham will matter to miners.

2/24/16

6.857 Lecture

Fernando Troja

Shamir's Secret Sharing
Block ciphers: DES, AES, modes

• Project ideas 3/7 on Piazza

Secret Sharing

Master secret s

n friends $A_1, A_2 \ldots A_n$

t threshold $1 \le t \le$ # people needed to recreate s

s_i share of secret s given to A_i

any t or more friends can get together and reconstruct s.
fewer than t cannot.

Ex: $t=1$ each party gets $s_i = s$

$t=n$ $s_1, \ldots s_{n-1}$ random s_n chosen s/t $s = s_1 \oplus s_2 \oplus s_3 \oplus \ldots \oplus s_n$

$1 < t < n$??

Coefficient Rep

$f(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \ldots + a_1 x + 0$ over $\mathbb{Z}_p$, $s \in \mathbb{Z}_p$, t coeff

Points rep

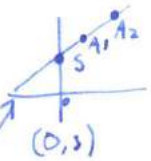
can also have t (x_i, y_i) for $1 \le i \le t$ on curve. x_i's distinct

$\{(x_i, y_i)\}$ ← Evaluation — $(a_{t-1}, a_{t-2}, \ldots a_1, a_0)$

t pt/valu pairs

Points Rep ↑ Interpolation → Polynomial Rep

$a_0 = \text{secret} = s = f(0)$



Degree t polynomial.
(not necesarily a line)

Choose $\not\exists$ $a_1, a_2 \ldots a_{t-1}$ from $\mathbb{Z}_p$

Let $x_i = i$ for $1 \le i \le n$

$x_i = F(i)$

(x_i, y_i) = share given to A_i

Interpolation:

Given any t points on curve $\Rightarrow a_0, a_1, \ldots a_{t-1}$

(x_i, y_i) $1 \le i \le t$

$$f(x) = \sum_{i=1}^{t} y_i \cdot F_i(x)$$

$f_i(x) = 1$ at $x = x_i$
0 at $x = x_j$ $j \neq i$

$$F_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

$$s = f(0) = \sum_{i \neq t}^{t} y_i \frac{\prod_{j \neq i} (-x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

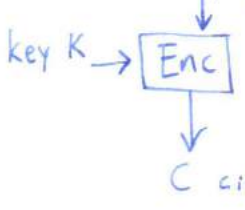
suppose adversary gets t-1 pts (x_i, y_i)
imagine adding $(0, s)$ to collection for any s
any secret s is consistent with what he knows.

↗ Information Theory

Only t-1 points => adversary know nothing about secret
(even with infinite computational power)

Block Ciphers

2 inputs → one output
Fixed length P, C, K



Standards:
DES : $|P| = |C| = 64$ bits $|K| = 56$ bits
AES : $|P| = |C| = 128$ bits $|K| = 128, 192, 256$

modes of operation : extend to variable length inputs

Ideal Block cipher

1:1
Given key k : mapping $Enc(K, \cdot)$ b/w plaintext space and ciphertext space.
and chosen independently at random from a set of all such mapping.

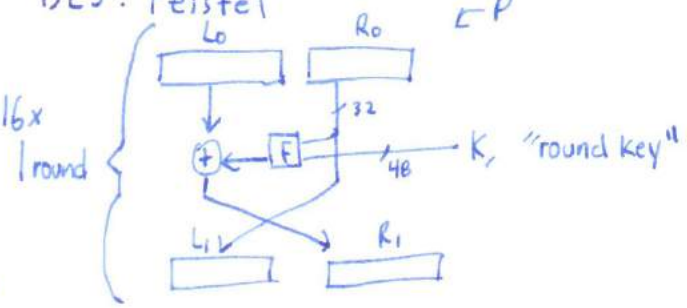
Ex: K=0

P	C
00	10
01	11
10	01
11	00

n-bit plaintext (cipher text)
k-bit keys

$(2^n!)^{2^k}$

DES: Feistel

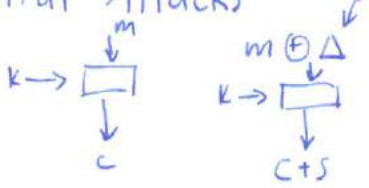


- Invertable ← can decrypt

Attacks: Brute-force search
2^{56} keys (matching PT/CT)

Differential Attacks

Flip a few bits of m.



- chosen PT attack
- statistics on (Δ, S) pairs
- 2^{47} pairs to reconstruct

Linear Attack (Matsoi)

$M_3 \oplus M_{15} \oplus C_2 \oplus K_{14} = 0$ true $\frac{1}{2}-\epsilon$ of the time

↳ Find biased equations

- known PT/CT attack
- amount of data needed $\approx 1/\epsilon^2$
- 2^{43} samples enough

6.857 Quiz 1 Notes

Security

Unconditional Security does not imply it can't be malleable (One Time pads)

Hashing

- CR must not allow second-preimage attacks
- In commitment schemes, randomization is required for hiding. OW -> Hiding. CR->Binding
- Birthday paradox: $2^{d/2}$ to find collisions for 2^d possibilities.
  *OW: Hard to find any preimage of random output. So f(x) = 1 is not one way.

Block Ciphers

- CBC decryption is much faster than encryption because CBC decryption can happen in parallel.
- CBC-MAC IV is set to 0 so it is **not** randomized.

Public Key Encryption

- Cramer-SHoup and RSA-OAEP are IND-CCA2 secure
- Can re-randomize El Gammal (generate a different encryption of the same message without knowing message or sk) by picking random r' and turning (g^r, my^r) to $(g^{r+r'}, my^{r+r'})$

Commitments

- Pedersen commitment are computationally binding but any value could yield the same commitment (so perfectly hiding).

Number Theory

- If CDH is easy $\rightarrow$ DDH is easy.
- If z is a quadratic redisdue (mod p) $\rightarrow x^{(p-1)/2} \equiv 1(modp)$
- BLS would be insecure if DLP was easy on G_2 where $G_1 \times G_1 \Rightarrow G_2$
- No need to know size of Group to computer inverse of element in group. Only need to know $\phi(n)$
- Eliptic curve defined by $y^2 = x^3 + ax + b(modp)$ will have $p + 1 + t$ solutions.

$5^{32} \bmod 31$?

By Fermat $\rightarrow a^{P-1} = 1 \pmod{p}$

$5^{30} = 1 \bmod 31$

so $5^{32} = 25 \bmod 31 = 25$

$6^{46} \bmod 23$

By Fermat

$6^{22} = 1 \bmod (23)$

so

$6^{46} = \underbrace{(6^{22})^2}_{1} 6^2 = 36 \pmod{23} = 13$

$3^{35} \bmod 17$ ← p

By Fermat $a^{P-1} = 1 \bmod P$

$3^{16} \bmod = 1 \pmod{17}$

$\underbrace{(3^{16})^2}_{1} 3^3 = 27 \pmod{17} = 10$

1

6. 857 Recitation #4

Thursday, February 25, 2016 11:20 PM

Improved Generic Algorithm for 3-way collisions

Goal:

Given random map $H : [0, N-1] \to [0, N-1]$

Find distinct x_1, x_3 s.t

$$H(x_1) = H(x_2) = H(x_3)$$

Remarks:

(1) random map

(2) $2^d \ll$ space of nounces

(3) Expected fraction of points with exactly k distinct pre-images is $e^{-1}/k!$.

So $\approx 8\%$ points have at least 3 distinct preimages

(4) Functional Graph G_H

① nodes are all values in $[0, N-1]$

② Directed edges from node x to node y iff $H(x) = y$

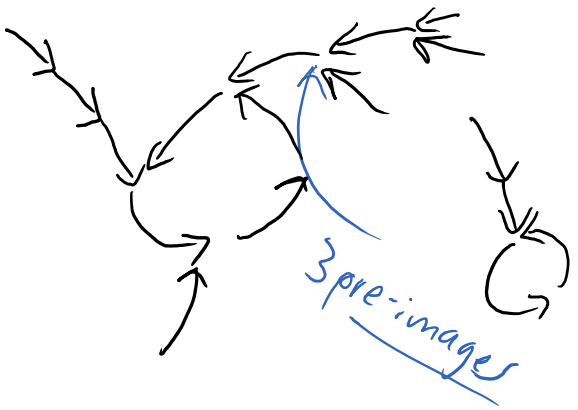
#components : $\frac{1}{2} \log N$

Among them, there is one **giant component.**

In Addition it contains a **giant tree.**

giant component: $2N/3$

giant tree : $N/3$



Pset 2 problem 1:
Expected length of tail and cycle are both $O(\sqrt{N})$

(5) Find r-collision for small r.

one needs to evaluate $\Omega(N^{(r-1)/r})$ the map.

3-collision : $\Omega(N^{2/3})$

Space and Parallelizable

6.857 Recitation 2/

Friday, February 26, 2016 11:35 AM

(6) Motivation:

cryptoanalysis of NSA, SHA3 candidates.

6.857 Guest Lecture

Monday, February 29, 2016 11:06 AM

with Nikola Zeldovich

Private messaging

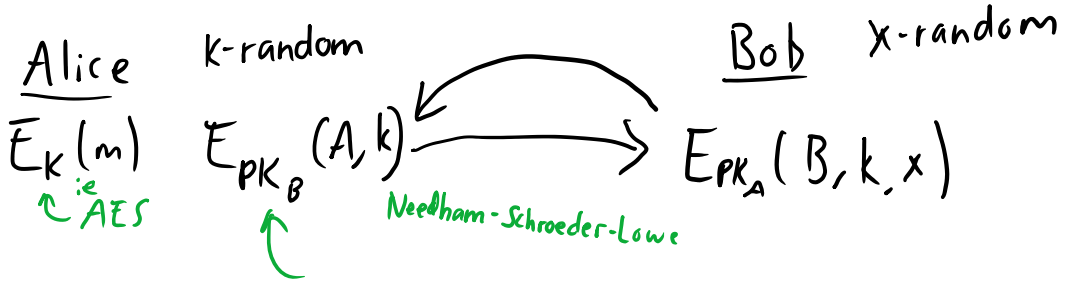
- confidentiallity
- performance
- integrity/auth

- Anonimity
- Deniability
- Forward-secrecy

confidential even if receipient compromised later

Confidentiality - encrypt

Initial Key exchange



send symmetric key using public key. Since PK is expensive vs symmetric

Both parties use $S = (k \oplus x)$

Guarantee only Alice and Bob know S

forward secrecy - Ephemeral Keys

can forget SK and r after exchange

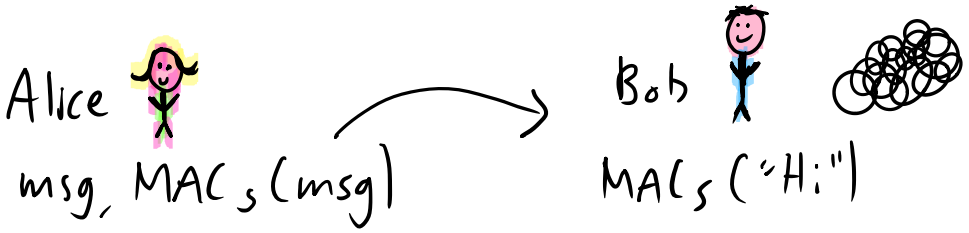
Authentication / Integrity

Deniability

'Bob can verify message but cannot prove to someone else.

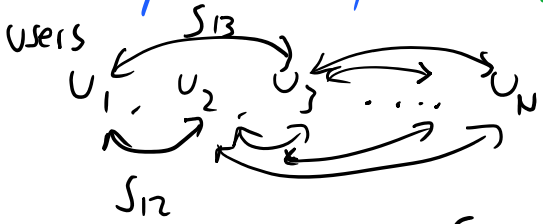
↳ message authentication code

$MAC_k(m) \rightarrow token \approx H(k \| m)$



(Textsecure / signal app implements this)

Anonimity w/ crypto (DCnets)



Pairwise secret for all users

$s_{ij} = s_{ji}$

$b_i = s_{i1} \oplus s_{i2} \oplus s_{i3} \oplus \ldots \oplus s_{iN} \oplus msg$ (or $\emptyset$)

$msg_{out} = b_1 \oplus b_2 \oplus \ldots \oplus b_N$

Each shared secret cancels out in pairs

No way to know who sent message!

Information Theoretic secure.

But => Fragile and slow and bad availability

What if attacker got a user's secret bits?

Anonimity set → user who could have sent message

compromised users not in Anonimity set

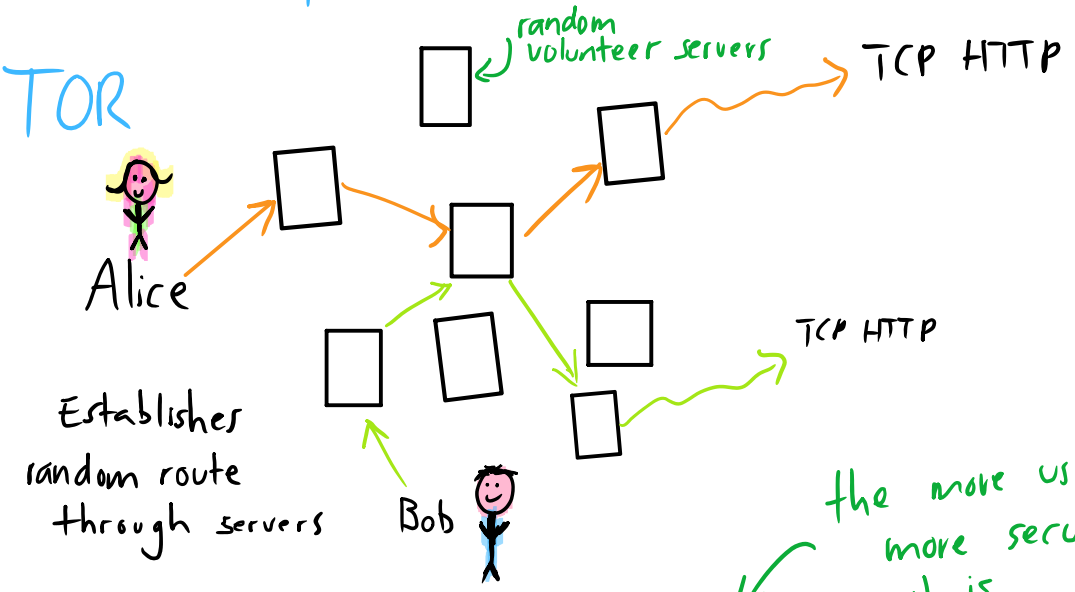
↳ not as secure.

Anonimity w/ systems

Onion encryption

Servers S_1 S_2 S_3 $E_{S_1}(E_{S_2}(E_{S_3}(msg)))$

TOR



the more users the more secure / anonimous it is.

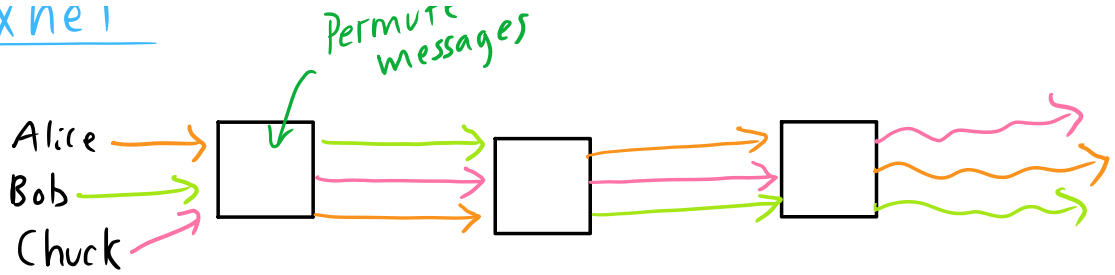
Secure and used in practice.

Seceptable to traffic analysis to "guess" routing

Mixnet

Randomly Permute messages

Mixnet



Not susceptable to traffic analysis.

but have to wait for x people to send message

6.857 Guest Lecture

Wednesday, March 2, 2016 11:07 AM

"Going Dark" with Timothy H. Edgar

1928 - Olmstead v. United States
Wiretapping not a "search" under 4th amendment
"The right to be let alone"

protects against unreasonable search and seizure
need warrant

Technologists and government define privacy differently

Technologists - Privacy = encryption = cannot break into

Lawyers - Privacy = secure, but should have access if needed

F.B.I. v. Apple
↳ Jim Comey
↳ Tim Cook

Cloud - Anything stored by someone else not subject to the 4th amendment.

"Encryption is the 2nd amendment of the digital world" → self defence/protection

Policy 1 - Must be backdoors in all data and communications

Policy 2: Lawful hacking -> okay for government to hack but,

If device or service can be broken can government make you break it?

United States v. Public Telephone 1970's

Government wanted to install pen register device on phone for metadata analysis

Using All Writs Act - government made telephone company comply with surveillance operation.

Where does this stop? Can government force individuals to help w/ operations?

Limiting Principles

1. Nexus - Must have a connection ie your phone line
2. Unreasonable Burden
3. Necessity - Government needs assistance to conduct lawful search.

Policy 3: Government can only get insecure data and communicate

Should government be required to report security vulnerabilities?

vulnerability equity process ← recently declassified → perhaps ethical

- Individuals don't have a legal obligation

6.857 Recitation

Friday, March 4, 2016 11:12 AM

- Alternate investigative means
- Evading Encryption access
- consequences of limiting encryption

Grounded truth about encryption

Ground Truths

How damaging is encryption to law enforcement?

- ~10% of cases hindered by encryption

Recent Incidents

Paris - Attackers used WhatsApp (encrypted) and unencrypted SMS

Garland - Attackers exchanged 109 encrypted messages with known oversees terrorists on day of attack.

San Bernardino / Apple -

- Good investigative practice relies on the concatenation on small, seemingly disperate pieces of information
- Since 2010 a non trivial # of cases solved by FBI have utilized tips by social media.
- Some methods are deemed "out of bounds"

Evading Encryption access

- Variety of "low-tech" methods and/or use alternative tech.
- TOR (Anonimity on the web) Anonimity vs. safety
- Even if US agrees does not mean it will be universally pervasive

Consequences of Limiting Encryption

- Any new cryptosystem that adds exceptional access into existing architecture will be more vulnerable to unauthorized access and covert use of the decrypting mechanism.

Athens Affair 2004-2005 → backdoor

Perpetrator exploited intercept system in phones to listen in on 100s of government officials for 10 months. Including prime ministers

RSA Seed key breach 2011

Attacked Lockheed Martin. Shows that it is difficult to store and maintain keys at scale.

International Consequences

- Foreign adoption of Access Requirements

UK - IPB

France - After Paris attacks - proposals made to limit encryption. Pressure to ban TOR and open wifi. Declined (for now)

Australia - Criminalizes export of encryption tech.

Creating new risks and vulnerabilities

- Companies will not be interested in innovating their security if they are forced to break it.
- Confidentiality, Integrity, Authentication

- Important when used to protect infrastructure and not just information.

- No distinguishing between different types of keys
  - keys that allow access to system may also allow modification of the system

Examples: IaaS (infrastructure as a service)
- keys giving access also give admin access.

Eroding American Advantage

- Snowden revelations - impact to US IT → $35-$180b
- American companies building infrastructure abroad due to fear of US government spying.
- Foreign countries breaking contracts with US companies

6.857 Lecture

Monday, March 7, 2016 11:06 AM

Modes: common: ECB / CTR / CBC
CFB. Ideal: IND$ → CCA
Desai's UFE mode

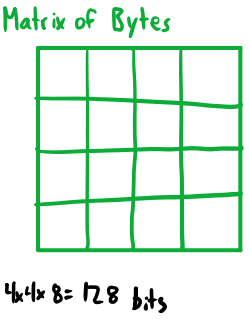
Block ciphers: AES

AES Contest 1997-1999: (Rijndael won)

128-bit block size

128, 192, 256-bit keys

10, 12, 14 rounds



Simplest Version:

128-bit key
10 rounds
11 round keys
in each round:

non linear part
- XOR in round key
- substitute bytes
- rotate rows (by different amounts)
- mix each column (linear operation)

↳ In last round, use XOR of last round key

Modes of Operation - Domain extention

- Design so it works with different length inputs.

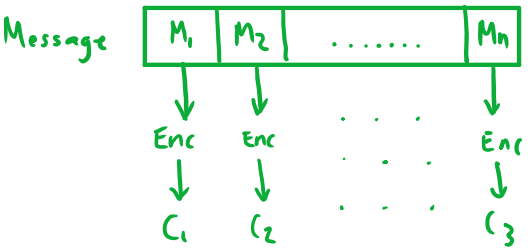
Padding: append 1 bit and enough 0 bits to give a multiple of 128 bits.

message → pad → encrypt → send → decrypt → remove pad → msg

* Pad has to be removable.

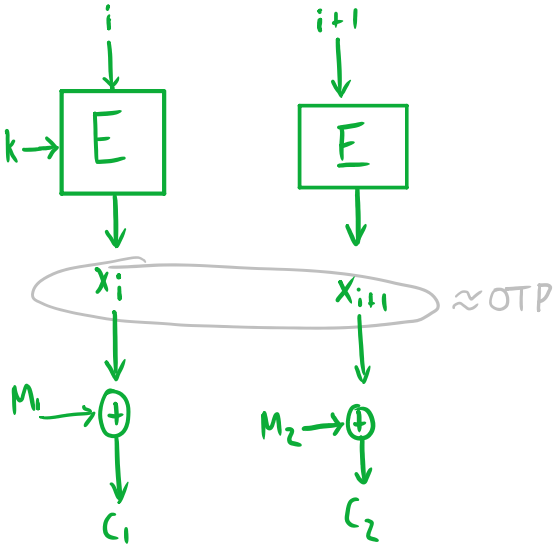
ECB: (Electronic Code Book)

- Patterns in message reveal themselves in ciphertext
- OK if you are encrypting random data



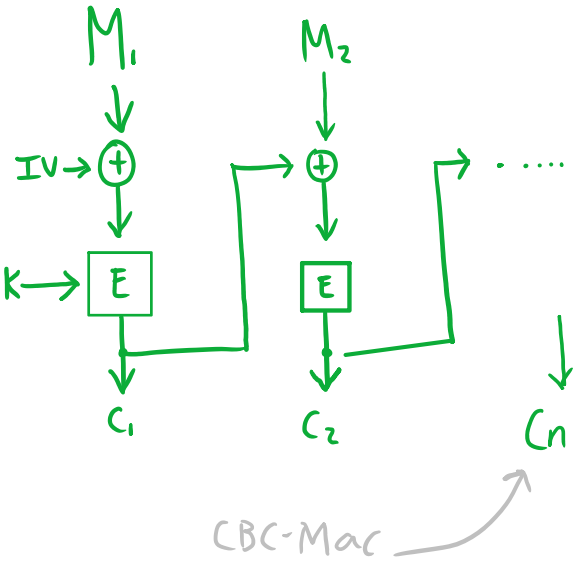
CTR (Counter Mode)

- Need to ensure i's are distinct to avoid reusing pads



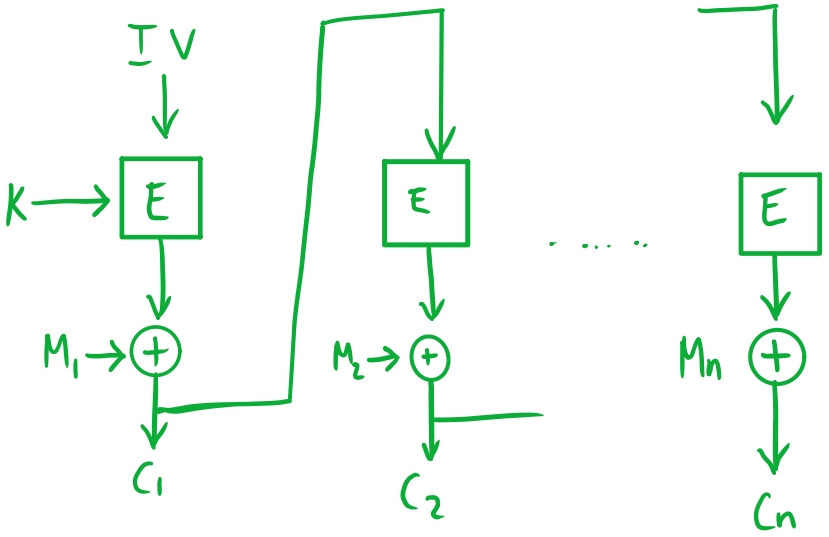
CBC

- Hides patterns in ciphertext that may exist in message
- Last key behaves like "checksum".
- Encrypt last message with different keys



CBC-Mac

CFB (Cipher Feedback Mode)



How to pick mode of operation

(Indistinguishable Chosen Cipher Text Attack)

IND-CCA

K key
E_K encrypt w/ K
D_K decrypt w/ K

Define game that gives adversary an advantage.

Phase I ("Find")

Adversary given E_K, D_K
"give us m_0, m_1 s.t $|m_0| = |m_1|$"
output state s

Phase II

can still use E_K, D_K but not with y. no $D_K(y)$

$d \xleftarrow{R} \{0,1\}$, $y = E_K(m_d)$

Adversary gets y, s outputs his guess $\hat{d}$ for d.

Adversary advantage $P(\hat{d} = d) - 1/2$

Scheme is IND-CCA secure if advantage is "negligible"

Deterministic Encryption cannot pass this!

IND-CTA Secure

IND-CTA NOT Secure

ECB

Idea:

Messaging app that hides the fact that you are sending messages.

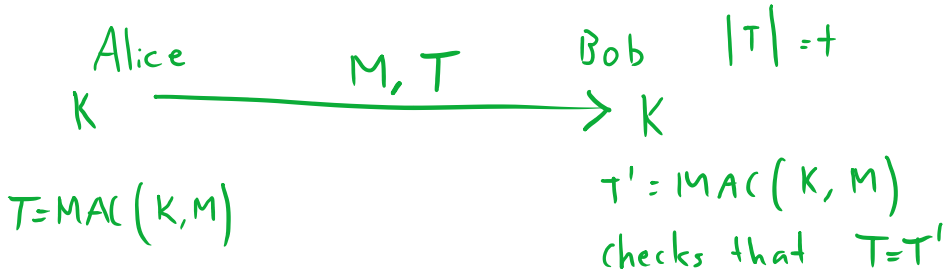
It does this by sending encrypted gibberish every X minutes when message queue is empty and encrypted message when it is not

6.857 Lecture

Wednesday, March 9, 2016 11:03 AM

Message Authentication Gates (MAC)

HMAC
PRF-MAC
CBC-MAC
One-time MAC
AEAD
EAX MAC
Encrypt-then MAC



Adversary wants to forge a message that Bob will accept
Random guessing: 2^{-t} chance of success

Replay Attack - Have way to keep "freshness" of the message. Otherwise attacker can rapture and send the same message later.

Game: Adversary gets to hear many valid (M,T) pairs. Possibly with messages of her choice.
↳ give adversary advantage

wants to forge a new (M', MAC(K,M')) that Bob will accept.

MAC is secure (under chosen message attack) if Prob (Adv wins) is negligible.

... is negligible.

HMAC

K: shared key

$K_1 = K \oplus opad$

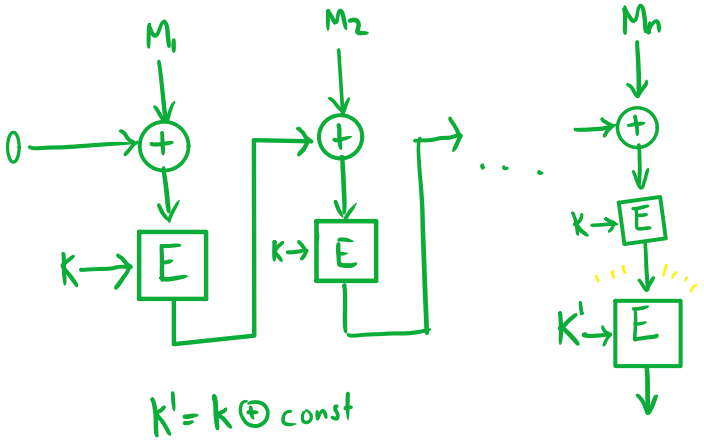
$K_2 = K_2 \oplus ipad$

$h(k_2 \| M)$ → PRF MAC (Pseudo Random Function)

$HMAC(K, M) = h(k_1 \| h(k_2 \| M))$

CBC-MAC

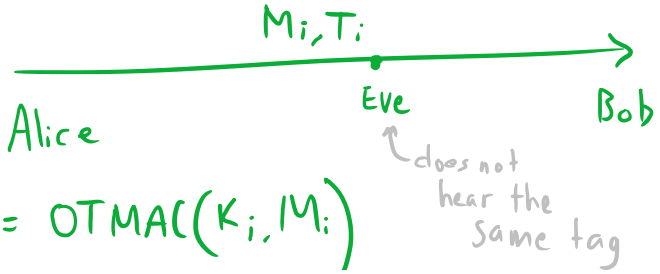
↳ Cipher Block chaining



CBC-MAC(K, m) (truncate to t bits)

One-time MAC

- Unconditional Security
- Confidentiality



use once key needed

$T_i = OTMAC(K_i, M_i)$

use once key needed $T_i = OTMAC(K_i, M_i)$ hear the same tag

Game: Eve sees (M, T) on wire, wants to replace it with (M', T') $M' \neq M$ that Bob accepts.

Recap

	Confidentiality	Integrity
Unconditional	OTP	OT Mac?
constitutional	Block Ciphers	MACs
Public Key options	PK encryption	PK signatures ↑ non repudiation

AEAD (Authenticated Encryption w/ Associated Data)

Encrypt-then-MAC

Alice —— C, T ——→ Bob

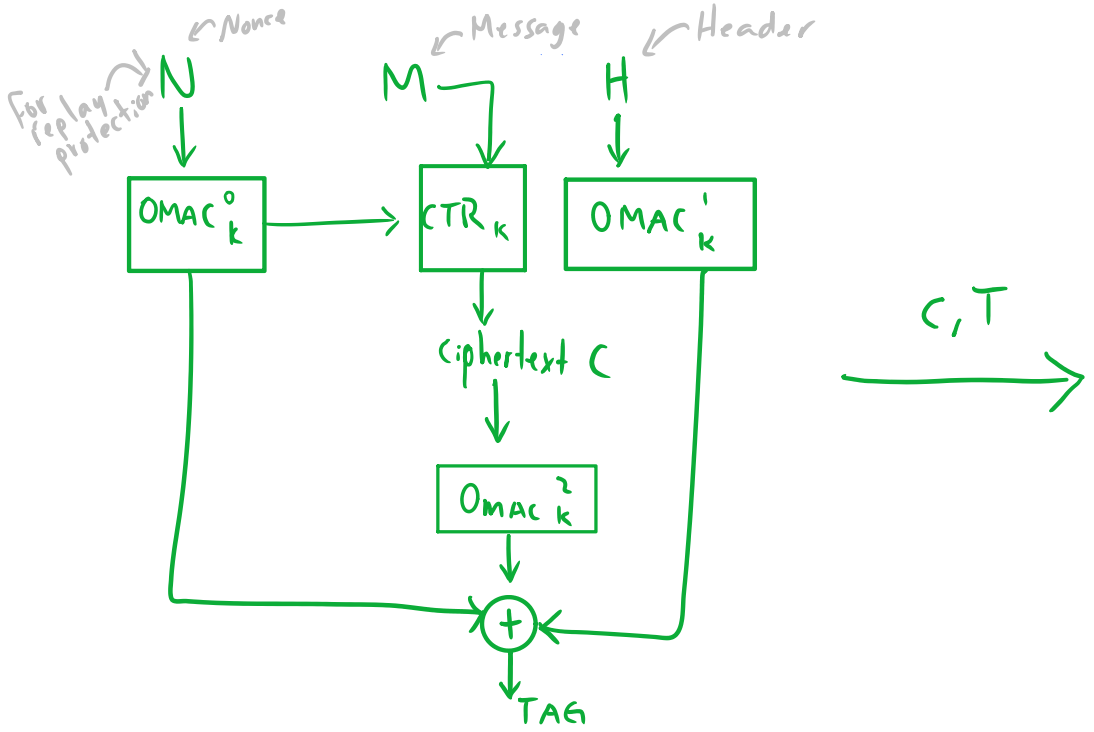
$C = Enc(K, M)$ ← header no need to enc

$T = MAC(K', H \| C)$

Encrypt first then Tag because tag T does not promise not to leak info about message

check that $T' = MAC(K', C)$ then decrypt $M = Dec(K', C)$

EAX mode of authenticated encryption



Groups and Fields

Finite Field: $(S, +, \cdot)$

$0 \in S$ additive identity

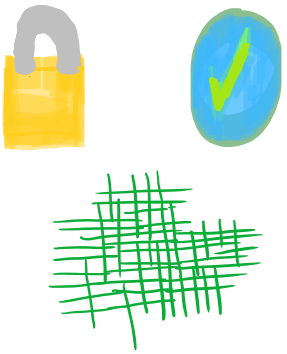
$1 \in S$ multiplicative

$+, \cdot$ commutative, associative
distributive

6.857 Lecture

Monday, March 14, 2016 11:07 AM

π day



Number Theory for Crypto

Repeated Squaring

$$a^b = \begin{cases} 1 & \text{if } b = 0 \\ (a^{b/2})^2 & \text{if } b > 0 \text{ b even} \\ a \cdot a^{b-1} & \text{if b odd} \end{cases}$$

- efficient
- recursive

Fermat's Little Theorem

Finding inverses

In a multiplicative group of $GF(q)$

$$\forall a \in GF(q)^* \quad a^{q-1} = 1$$

$$\Rightarrow a \cdot \underbrace{a^{q-2}}_{a^{-1}} = 1$$

Finding k-bit prime

Generate a random k-bit integer until we find a prime.

primes are common

$$\approx 2^k / \ln(2^k)$$ k bit primes

By Fermat's

$\approx 2^k / \ln(2^k)$ k bit primes

Testing Primality

By Fermats

For random p

is $2^{p-1} = 1 \pmod p$ ↗ prime always passes "good enough"

↳ non-prime almost never pass

(or use Miller-Rabin test)

Euclids Algorithm

see notes!

Find gcd efficiently.

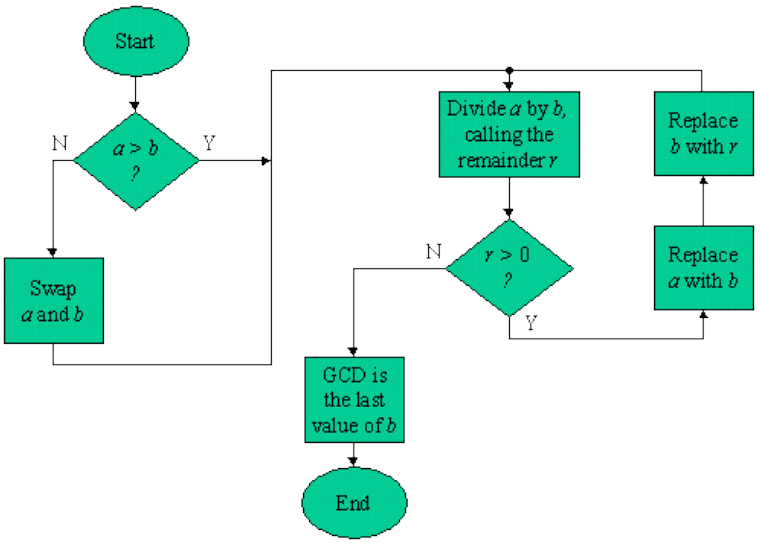
$\gcd(5,7) = 1$

Efficiently computable by Extended Euclid Alg.

$(\forall a,b)(\exists x,y)\ ax+by = \gcd(a,b)$

What is $a^{-1} \pmod n$ $\quad ax + bn = \gcd(a,n) = 1$

$ax = 1 \pmod n \quad x = a^{-1} \pmod n$



Generators

p prime ↙ p-1 elements

$$\mathbb{Z}_p^* = GF(p)^*$$

$\text{order}_n(a) \triangleq$ least $t > 0$ s.t. $a^t = 1 \pmod n$

Euler's Theorem

$$(\forall n)\,(\forall a \in \mathbb{Z}_n^*)\; a^{\varphi(n)} = 1 \pmod n$$

where $\varphi(n) \triangleq |\mathbb{Z}_n^*| = |\{1 \le a < n : \gcd(a,n) = 1\}|$

$n = 10$

$\mathbb{Z}_{10}^* = \{1,3,7,9\}$ $\quad \varphi(10) = 4$ $\quad 3^4 = 1 \pmod{10}$

$\mathbb{Z}_7^*$

	1	2	3	4	5	6	7	...	order
1:	1	1	1	1	1	1	1	...	1
2:	2	4	1	2	4	1	2	4 ...	3
3:	3	2	6	4	5	1	3	...	6
4:	4	2	1	4	2	1	4	...	3
5:	5	4	6	2	3	1	5	...	6
6:	6	1	6	1	6	1	6	...	2

if p prime

$\text{order}_p(a) \mid p-1$

$\langle a \rangle = \{a^i,\ i \ge 0\}$

$|\langle a \rangle| \mid |\mathbb{Z}_n^*|$

g is a generator mod p if $\text{order}_p(g) = p-1$

$(0 \le x \le p-2)$

$\Rightarrow$ Equation $g^x = y \pmod p$ has a unique solution x for each y in $\mathbb{Z}_p^*$

x is the discrete logarithm of y, base g, modulo p

How to find generators?

$\mathbb{Z}_n^*$. Thm: $\mathbb{Z}_n^*$ has a generator (ie $\mathbb{Z}_n^*$ is cyclic) IFF

n is one of $2, 4, p^m, 2p^m$ for some prime p and integer $m > 0$

$\mathbb{Z}_p^*$: # generators (mod p) is $\varphi(p-1)$

$p = 11$: $\varphi(p-1) = \varphi(10) = 4$

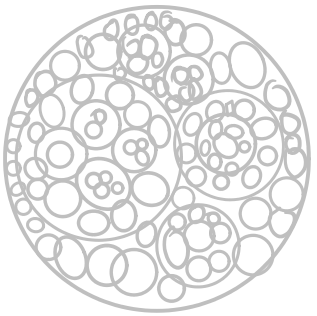
Generators are: 2, 6, 7, 8

Both primes

$p = 2q + 1$

p is a "safe prime"

q is a Sophie Germain "prime"



$$\varphi(p-1) = |\mathbb{Z}_{2q}^*| = (2q) - q - 1$$

Find generator for safe prime

$p = 2q + 1$

$|\mathbb{Z}_p^*| = 2q$

order(a) $\in \{ 2, q, 1, 2q \}$

$\text{order}_p(a) \in \{2, q, 1, 2q\}$

is $a^t = 1 \pmod p$ for t ↖

$\Rightarrow \text{order}_p(a)$ easily computable

To find a generator $g \bmod p$ $(p = 2q+1)$

pick $g \in \mathbb{Z}_p^*$ at random

compute $\text{order}_p(g)$ (one of $1, 2, q, 2q$)

if $\text{order}_p(g) = 2q \Rightarrow g$ is generator.

else

⋮

(see notes)

6.857 Lecture

Wednesday, March 16, 2016 11:06 AM

Group Theory

Group Theory Review

DH Key exchange

5 Crypto groups: $\mathbb{Z}_p^*$, Q_p, $\mathbb{Z}_n^*$, Q_n, elliptic curves

Group Theory Review

$(G, *)$ is a finite group of size t ($\exists$ identity 1)

$(\forall a \in G)\ a \cdot 1 = 1 \cdot a = a$

$(\forall a \in G)(\exists b \in G)\ a \cdot b = 1$ ← inverse of a

$(\forall a, b, c)\ a \cdot (b \cdot c) = (a \cdot b) \cdot c$

$(\forall a, b)\ a \cdot b = b \cdot a$

order(a) = least $u > 0$ s.t. $a^u = 1$

order(a) | t ← divisible by — Lagrange's Thm.

$\forall (a \in G)\ a^t = 1$ $(a^{p-1} = 1 \pmod p)$

$\langle a \rangle$ = subgroup generated by a

$|\langle a \rangle| = \text{order}(a)$

if $\langle a \rangle = G$ then G is cyclic and a is a generator of G

Excersise In a finite abelian group G of size t

where t is prime $(\forall a \in G)\ a \neq 1$

$\Rightarrow a$ is a generator

$\mathbb{Z}_p^*$ is always cyclic

g generates G

$x \longmapsto g^x$ is one-to-one between $[0, 1, \ldots, t-1]$ and G

$x \to g^x$ is easy. (repeated squaring)

$g^x \to x$ is usually in "crypto groups"

Discrete Log Problem (DLP)

DLP seems to be hard in $\mathbb{Z}_p^*$

where p is large randomly chosen prime.

Assume mapping b/w set of messages and elements in G.

API ← "sage" in python

create group G

G.identity()

.product(x,y)

.power(x,k) x^k

.random_element()

inverse(x)

G.order()

.elements()

.generator()

.discrete log(g,y)

Public Key setup (for some system)

p large prime

g generator mod p.

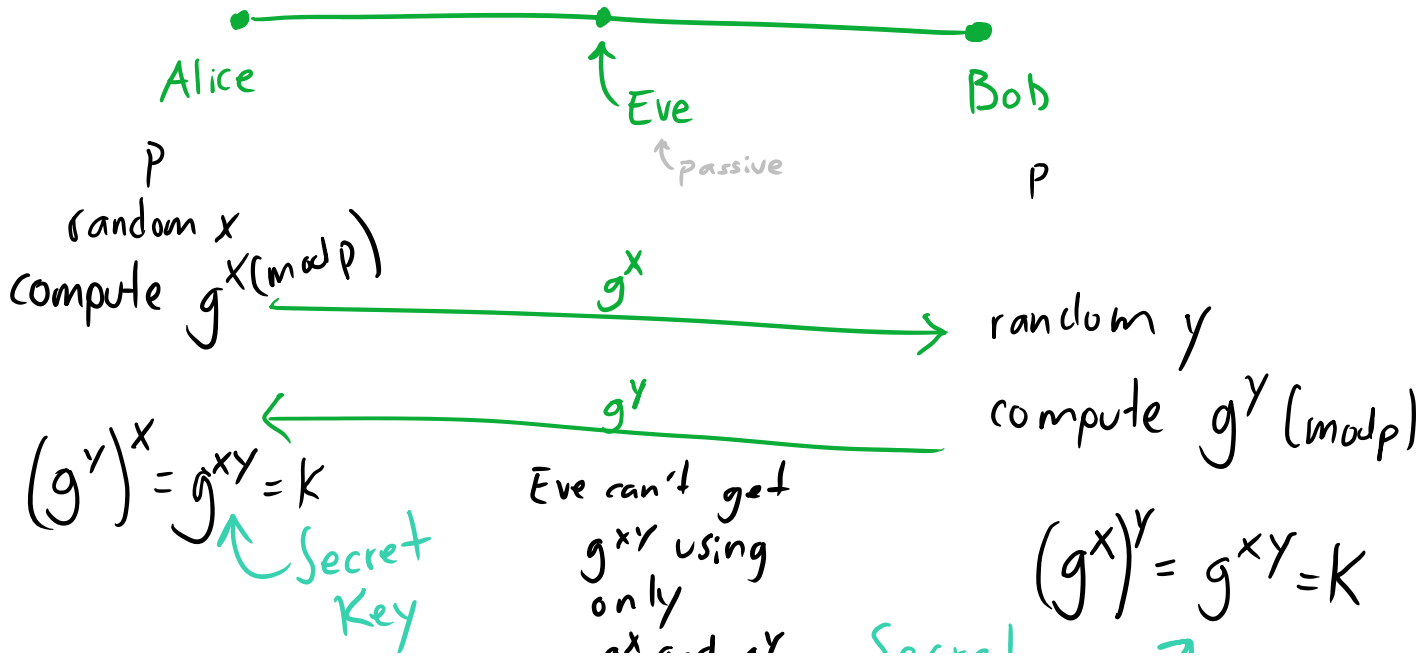
Alice chooses secret key x $0 \le x \le p-1$

computes public key $y = g^x$

Assume: DLP is hard mod p.

Diffie-Hellman Key Exchange

Problem: Establish a secret key between A and B without Eve knowing key



Key

only g^x and g^y

Secret Key → $(g^y)^x = g^{xy} = K$

CDH Assumption: Given g^x and g^y is hard to compute g^{xy}

- Computational Diffie Hellman
- Forward Secrecy ✓

Cryptogroups

① $\mathbb{Z}_p^* = \{1 \le a < p\}$ Integers mod p

- always cyclic
- safe prime $p = 2q+1$ is a "safe" prime.
- aside from 1, half of $\mathbb{Z}_p^*$ are generators other are squares.

② Q_p square mod p $= \{a^2 \bmod p\}$

$|Q_p| = (p-1)/2$

g^2 generates Q_p $g^{2i} = (g^2)^i$

$p = 2q+1 \Rightarrow |Q_p| = q$

Every element (except 1) of Q_p is a generator of Q_p

③ $\mathbb{Z}_n^* = \{ a : \gcd(a,n) = 1 \text{ & } 1 \le a \le n \}$ RSA Group

$|\mathbb{Z}_n^*| = \varphi(n)$ Not Cyclic

if $n = p \cdot q$: $\mathbb{Z}_n^* = \mathbb{Z}_p^* \cdot \mathbb{Z}_q^*$

④ $Q_n = \{ a^2 : 1 \le a \le n \text{ & } \gcd(a,n) = 1 \}$

Squares mod n. Quadratic Residues mod n

if $n = p \cdot q$ and $p = 2r+1$ (safe), $q = 2s+1$ (safe)

$\Rightarrow |Q_n| = r \cdot s$ & Q_n is cyclic

p = prime

a, b be elements of $\mathbb{Z}_p$ such that

$$4a^3 + 27b^2 \neq 0 \pmod{p}$$

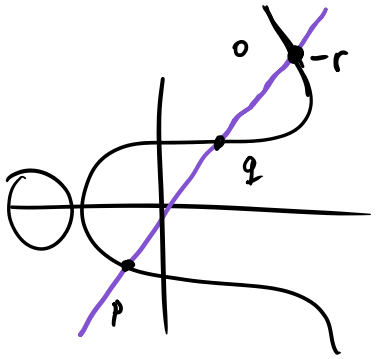
Curve:

$$y^2 = x^3 + ax + b \pmod{p}$$

Bernstein's Curve ← way of defining a finite group

$$y^2 = x^3 + 486662x^2 + x \pmod{p}$$

$$p = 2^{555} - 19$$



$$E_{a,b} = \{(x, y)$$

$$y^2 = x^3 + ax + b \pmod{p}$$

$$+ \mathcal{O}$$

$$R = P + Q$$

6.857 Recitation

Friday, March 18, 2016 11:06 AM

Elliptic Curves

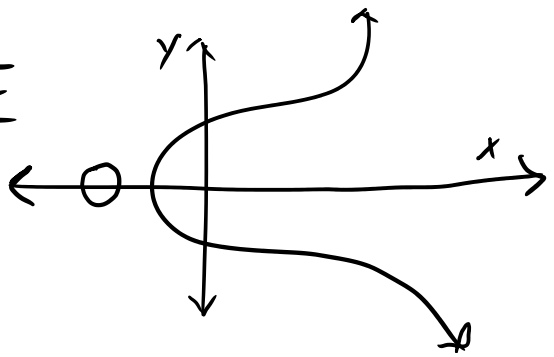
prime p, $a, b \in \mathbb{Z}_p$ s.t $4a^3 + 27b^2 \neq 0$ (no dp)

$$E = \{(x,y) : y^2 = x^3 + ax + b \pmod p\} \cup \{\infty\}$$

Fact 1: $|E| = p + 1 + t$ where $t \leq 2\sqrt{p}$

Fact 2: $|E|$ can be computed efficiently

if $(x,y) \in E \rightarrow (x,-y) \in E$



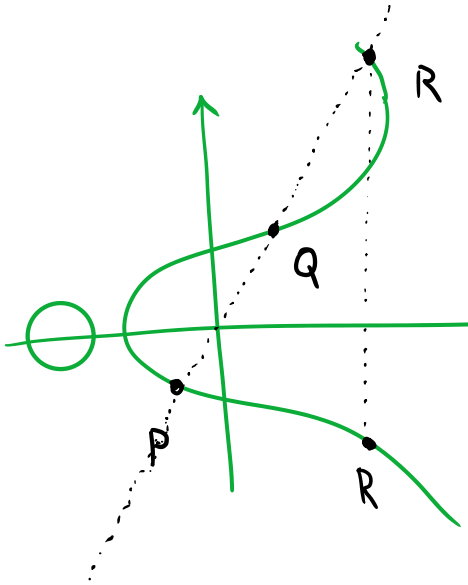
We can define a binary operation "+" on E s.t $(E,+)$ is a finite abelian group.

- ∞ is the identity element:

  $\forall P \in E \rightarrow \infty + P = P + \infty = P$

- $-(x,y) = (x,-y) \rightarrow (x,y) + (x,-y) = \infty$

- $-\infty = \infty \rightarrow \infty + \infty = \infty$



$P = (x_1, y_1)$

$Q = (x_2, y_2)$

$R = P + Q = (x_3, y_3)$

Case 1

- $x_1 \neq x_2$; slope $m = \dfrac{y_2 - y_1}{x_2 - x_1}$ $\qquad x_3 = m^2 - x_1 - x_2$

$$y_3 = m(x_1 - x_3) - y_2$$

- Case 2

  $x_1 = x_2$ and $y_1 \neq y_2$: $P + Q = \infty$ (vertical line)

- Case 3

  $P = Q$ and $y_1 = 0$: $P + Q = \infty$

- Case 4

  $P = Q$ and $y_1 \neq 0$: $m = \dfrac{3x_1^2 + a}{2y_1}$

$$x_3 = m^2 - 2x_1$$

$$y_3 = m(x_1 - x_3) - y_1$$

Thm: "+" is associative $P+Q=Q+P \quad \forall P,Q \in E$

Cor: $(E,+)$ is a finite abelian group

Fact: We can use $GF(q^k)$ instead of $\mathbb{Z}_p$

Why does this matter?

4096 bits RSA Key $\approx$ 313 bits eliptic curve key

Discrete Log Problem for eliptic curves

if $P \in E$, k is a positive integer then

$$kP \triangleq \underbrace{P+P+P\ldots.+P}_{k \text{ times}}$$

ElGamel Public Key Encryption

Bob — Alice

- choose E (eliptic curve) s.t
  DLP is hard
- choose $P \in E$ (usually)

order(P) to be large prime.

- Choose integer s and compute $B = sP$

- (E, F_q, P, B) — Public Key → Downloads Public Key

- Expresses message as point $M \in E$
- Chooses a random secret k integer and computes

$$M_1 = kP = \underbrace{P+P+P \ldots}_{k \text{ times}}$$

$$M_2 = M + kB$$

- Sends M_1, M_2 to Bob ←

Calculate

$M = M_2 - sM_1$

$M_2 - sM_1 = M + kB - s(kP)$

$= M + k(sP) - s(kP) = M$

EVE
- Knows M_1, M_2, Public Key
→ hard to decrypt M if DLP is hard

Important note:

Alice should use different random K every time she sends message to Bob.

if she uses same k for $M_1, M' \rightarrow M_1 = M_1'$

Then Eve computes $M_2' - M_2$

$= (M' + kB) - (M + kB) = M' - M$

if M is public → Eve calculates

$M + (M' - M) = M'$ BAD

GCD Algorithm

If $a, b \in \mathbb{Z}_n$ then gcd(a,b) is the largest possible integer that divides both a and b.

gcd(24, 30) = 6

Thm: If $k \mid a$ (k divides a) and $k \mid b$

$\Rightarrow k \mid \gcd(a, b)$

Thm: ($\forall a, b$ $\exists$ integers x, y s.t

$ax + by = \gcd(a, b)$)

↳ can find a, b with extended Euclidean Alg

6.857 Lecture

Monday, March 28, 2016 11:10 AM

Commitments and PK encryption

Pederson Commitment Scheme

commit(x) → C

Reveal(c)

Properties: Hiding: C reveals nothing about x

Binding: Can only open C to one value

Non-malleability: Can't generate commit(x+1) from commit(x)

p, q large primes where $q \mid p+1$ (eg $p = 2q+1$)

g generator of order q subgroup of $\mathbb{Z}_p^*$

$h = g^a$ (a - secret) ← chosen by auctioneer

p, g, q, h are public $\langle g \rangle = \langle h \rangle$

$x \in \mathbb{Z}_q$

r random in $\mathbb{Z}_q$ (randomization)

Commit = $c = g^x h^r \pmod p$ (Commitment)

Reveal: Give x, r

Hiding: Given $c = g^x h^r = g^{x'} h^{r'}$ for any $x \in \mathbb{Z}_q$

$$g^x h^r = g^{x'} h^{r'}$$

$$g^x g^{ar} = g^{x'} h^{ar'}$$

$$g^{x+ar} = g^{x'+ar'} \pmod p$$

$$x + ar = x' + ar' \pmod q$$

$$r' = \frac{(x - x')}{a} + r \pmod q$$

r' exsists

Information Theoretic Secure ✓

$\Rightarrow$ C could be for any commitment in the group

Binding: $c = g^x h^r = g^{x'} h^{r'}$

$$x + ar = x' + ar'$$

$$a = \frac{(x - x')}{(r - r')}$$

Computentially Binding ✓

Lying $\Rightarrow$ Can solve DLP $\Rightarrow$ HARD

Non-malleability

Can find a comitment of $x+1$ given a commitment for x.

Malleable ✗

Public Key Cryptography

λ = security parameter $\quad 1^\lambda = \underbrace{111\ldots1}_{\lambda}$

1. keygen$(1^\lambda) \rightarrow (PK, SK)$
2. $E(PK, m) \rightarrow c$ (may be randomized)
3. $D(SK, c) \rightarrow m$

Decouple encryption and decryption with different keys!

El Gamal Encryption

$G = \langle g \rangle$ some group

Keygen:

$SK = x \quad \in_R [0, \ldots |G|-1]$ (random)

$PK = g^x$

Encryption:

Let $y = g^x$ = SK of reccipient

pick k at randomm from $[0, 1, \ldots |G|-1]$

ciphertext $c = (g^k, m \cdot y^k)$

↑ element of the group

Decryption:

$c = (a, b)$

$m = b / a^x \quad a^x = g^{kx} = g^{xk} = y^k \Rightarrow m = b/y^k$ ✓

Like Diffie-Hellman key exchange

Need to assume DLP hard (infeasable to compute x from g^x)

Semantic Security

Phase I: Examiner generates PK, SK pair

gives PK to adversary

Adv can compute two messages m_0, m_1 $(|m_1| = |m_0|)$

$m_0 \neq m_1$

Phase II: Examiner picks $b \xleftarrow{R} \{0,1\}$

$c = E(PK, m_b)$

Gives c to Adv

Adv computes on c, outputs $\hat{b}$
Adv wins if $b = \hat{b}$

Scheme is **semantically secure** if $p(\text{Adv wins}) \leq \frac{1}{2} + n(\lambda)$
negligible function of λ

To be semantically secure, PK should be randomized
(or stateful encryption)

DDH (Decision Diffie Hellman)

← Assumption

Given group G with generator g.
Infeasable for an Adv to decide if a triple of inputs was generated as

(g^a, g^b, g^c) [a,b,c random]

or

(g^a, g^b, g^{ab}) [a,b random]

Like computational Diffie Hellman
↑ Infeasable to compute g^{ab} from g^a and g^b

Thm: DDH $\Rightarrow$ CDH

Security of El Gammal

Thm: El Gammal is secure in G

$\iff$ DDH Holds in G.

6.857 Lecture

Wednesday, March 30, 2016 11:07 AM

Public Key II + RSA

El Gamal malleable
IND-CCA2 Security
RSA, & OAEP, RSA Security

E Gamal

$$E(m) = (g^k, m \cdot y^k)$$

$$E(2m) = (g^k, 2m \cdot y^k)$$

Ciphertexts for different messages are related ∴

El Gamal is homomorphic

$$E(m_1) = (g^r, m_1 \cdot y^r)$$

$$E(m_2) = (g^s, m_2 \cdot y^s)$$

$$E(m_1)E(m_2) = (g^{r+s}, m_1 m_2 \cdot y^{r+s})$$

valid ciphertexts

Can multiply ciphertexts to create another

$$E(m) = (g^r, m \cdot y^r)$$

$$E(1) = (g^s, y^s)$$

$$E(m)E(1) = E(m) = (g^{r+s}, m \cdot y^{r+s})$$

another ciphertext for E(m)

Can rescramble messages without using PK

IND-CCA2 Indistinguishable - chosen cipher Attack

Phase I "Find": Examiner generates PK, SK

PK → Adv

Adv computes (time poly(λ))

↳ access to Decryption oracle

Outputs m_0, m_1, $|m_0| = |m_1|$, $m_0 \neq m_1$, and state info s

Phase II "Guess":

Examiner picks $b \in_R \{0,1\}$

$C^* = E(PK, m_b)$

$C^*, s \rightarrow$ Adv

Adv can compute (poly(λ) time)

can use Decrypt except C^*

Adv guesses $\hat{b}$

Adv wins if $\hat{b} = b$

Scheme is IND-CCA2 secure if

$P(\hat{b} = b) \leq \frac{1}{2} + \text{negligible}(\lambda)$

Strongest definition of security for PK crypto

El Gamal not IND-CCA2 secure

Ideas: Add redundancy to ciphertext checksum
Decryption to fail if checks fail

$c \to [D] \to \begin{matrix} m \\ \text{or} \\ \perp \end{matrix}$ ← Void aka ciphertext rejected

Cramer-Sharp. IND-CCA2 Secure Scheme

Keygen: $PK = (g_1, g_2, c, d, h)$

$SK = (x_1, x_2, y_1, y_2, z)$

Encrypt: $e = h^r \cdot m$ ← Hash Function

checks / redundancy depend on message

$c = (e, u_1, u_2, v)$

Decrypt: $\alpha = H(u_1, u_2, e)$

check $u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha} \stackrel{?}{=} v$

if not output $\perp$

else output $m = e / u_1^z$

Thm: Cramer-Sharp is IND-CCA2 Secure
if: DDH Hard and H is target collision resistant

Key Takeaways: Can have decryption fail to make scheme more secure.

RSA
↳ Rivest!

Diffie-Hellman notion of PK encryption (1976)

Diffie-Hellman notion of PK encryption (1976)

Keygen: $(1^\lambda) \to (PK, SK, \mathcal{M}, \mathcal{C})$ $|\mathcal{M}| = |\mathcal{C}|$

Encrypt: efficiently computable one-to-one map $\mathcal{M} \to \mathcal{C}$ deterministic

Decrypt: $D(SK, E(PK, m)) = m$

Hard to compute SK given PK

Inspiration for RSA

Keygen: Find 2 large primes p, q $\lambda = 1024$ bits

$n = p \cdot q$

$\varphi(n) = |\mathbb{Z}_n^*| = (p-1)\cdot(q-1)$

$e \xleftarrow{R} \mathbb{Z}_{\varphi(n)}$ so $\gcd(e, \varphi(n)) = 1$

$PK = (n, e)$ $\quad d = e^{-1} \pmod{\varphi(n)}$

$SK = (d, p, q)$

Encrypt: $C = E(m) = m^e \pmod{n}$

$m = D(c) = c^d \pmod{n}$

Correctness:

Chinese Remainder Theorem (CRT)

$n = p \cdot q$

$x = y \pmod{n} \iff x = y \pmod{p}$ AND $x = y \pmod{q}$

$e \cdot d = 1 \pmod{\varphi(n)}$

$e \cdot d = 1 + t\underbrace{(p-1)\cdot(q-1)}_{\varphi(n)}$ for some t

$e \cdot d = 1 \pmod{p-1} \Rightarrow d = e^{-1} \pmod{p-1}$

To show $(m^e)^d = m \bmod n \ (\forall m)$

Suffices to $(m^e)^d = m \pmod p$ by CRT

Case I: $m = 0 \pmod p$ $\quad 0^{ed} = 0 \pmod p$ ✓

Case II: $m \neq 0 \pmod p$

$m^{p-1} = 1 \pmod p$ → by Fermat's Little Thm

$m^{ed} = m^{1+u(p-1)}$

$= m \cdot (m^{p-1})^u = m \cdot 1^u = m$ ✓

$m^{ed} = m \pmod p \ (\forall m \in \mathbb{Z}_p)$

$m^{ed} = m \pmod q \ (\forall m \in \mathbb{Z}_q)$

Combining by CRT

$m^{ed} = m \pmod n \ (\forall m \in \mathbb{Z}_n)$

$\Rightarrow D(SK, E(PK, m)) = m$ 

d is secret

$d = e^{-1} \bmod \varphi(n)$

Knowledge of $d \equiv$ knowledge of $p \cdot q$

Assumptions: Factoring is secure

Difficulty of Factoring n

$$\exp\left\{c \cdot (\ln n^{1/3} (\ln \ln n)^{2/3}\right\}$$

↳ exponential in cube root of length of n

768 bits Factored already

1024 bits soon

Is RSA semantically secure?

RSA is not randomized ⇒ NOT semantically secure.

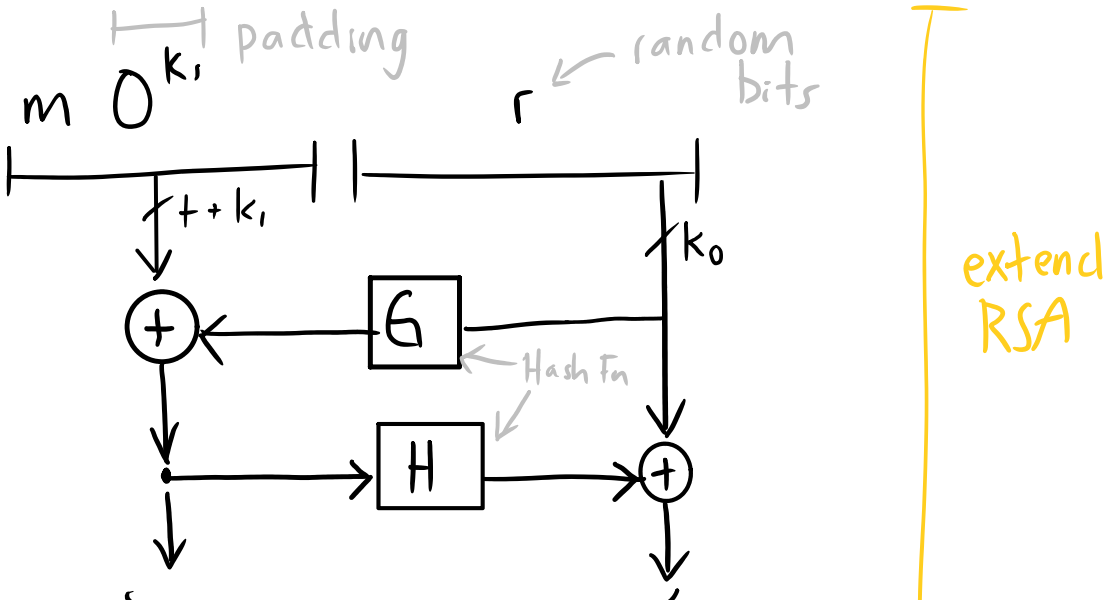
$m_0 \Rightarrow c_0 = E(PK, m_0)$

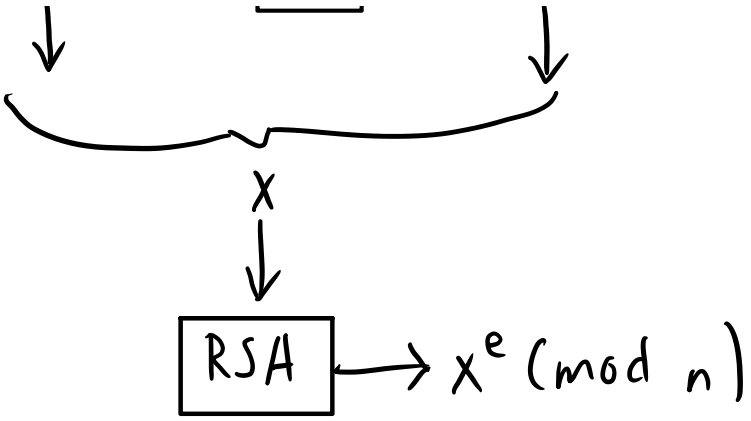
$m_1 \Rightarrow c_1 = E(PK, m_1)$

Optimal Asymetric Encryption Padding (OAEP)

modified RSA to achieve stronger notion of security. → preprocessing

Encrypt





Decrypt

invert RSA
invert OAEP
reject (output $\perp$) if 0^k
else output m

Thm: RSA w/ OAEP is IND-CCA2 secure
Assumes ROM for G and H
↖ Random Oracle Model
and RSA hard to invert on random inputs.

6.857 Recitation

Friday, April 1, 2016 11:16 AM

PK Crypto Review

RSA
El Gamal
IND-CCA 2
Cramer-Sharp
Elliptic Curve
Pedersen Commitments

RSA [Katz pg 355]

$Gen(1^\lambda)$: $N, e, d = GenRSA(1^\lambda)$
$PK = (N, e)$
$SK = (N, d)$

$Enc(PK, m \in \mathbb{Z}_N^*)$: ciphertext $c = m^e \bmod N$
$Dec(SK, c \in \mathbb{Z}_N^*)$: message $m = c^d \bmod N$

$GenRSA(1^\lambda)$ $(N, p, q) \leftarrow GenModulos(1^\lambda)$
$\phi(N) = (p-1)(q-1)$
choose e s.t $gcd(e, \phi(N)) = 1$
Compute $d \equiv e^{-1} \bmod \phi(N)$
return N, e, d

El Gamal [Kats pg 365]

Defined over cyclic group G with order q and gen g.

$Gen(1^\lambda)$: Construct group $(G, q, g) = GenGroup(1^\lambda)$
Choose $r \xleftarrow{R} \mathbb{Z}_q$. $h = g^r$

$PK = (G, q, g, h)$

$SK = (G, q, g, r)$

Group definitions

$Enc(pk, m \in \mathbb{Z}_q)$:

choose $r \xleftarrow{R} \mathbb{Z}_q$

output ciphertext $c = (g^r, m \cdot h^r)$

$Dec(sk, c)$

Let $(c_1, c_2) = c$

Compute $c_2 / c_1^r = m$

IND-CCA2 (ACCA)

- Strongest Definition of Security for PK crypto

Defined as 2 phase game between an examiner E and an adversary A

Phase I: Find

E generates $(PK, sk) = Gen(1^\lambda)$

E sends PK to A — essentially gives access to enc oracle

A compute for poly-time in λ with access to $Dec(sk, \cdot)$

A outputs m_0 and m_1 and any state info S.

Phase II Guess

E picks $b \xleftarrow{R} \{0, 1\}$ and computes

$c' = Enc(PK, m_b)$

E sends c' and s to A

A computes (poly-time in λ) w/ access to $Dec(sk \neq c')$

A outputs guess $\hat{b}$

Encryption scheme IND-CCA2 Secure if

$$Pr[\hat{b} = b] \leq \frac{1}{2} + negligible(\lambda)$$

El Gamal not IND-CCA2 Secure

$\Downarrow$

Cramer-sharp cryptosystem

Solves malleability problems in El Gamal

$Gen(1^\lambda)$

Choose $g_1, g \xleftarrow{R} G$

$SK = (x_1, x_2, y_1, y_2, z) \xleftarrow{R} \mathbb{Z}_n$

Hash Function $H: G^3 \to \mathbb{Z}_q$

$c = g_1^{x_1} g_2^{x_2} \quad d = g_1^{y_1} g_2^{y_2} \quad h = g_1^{z}$

$PK = (g_1, g_2, c, d, h, H)$

$Enc(pk, m \in G)$

Choose $r \xleftarrow{r} \mathbb{Z}_q$

$u_1 = g_1^r \quad u_2 = g_2^r \quad e = mh^r$

$\alpha = (u_1, u_2, e)$

$v = c^r d^{r\alpha}$

$C = (u_1, u_2, e, v)$

Dec(sk, c)

$a = H(u_1, u_2, e)$

IF $u_1^{x_1 + x_2 \alpha} u_2^{y_2 + y_2 \alpha} \neq v \Rightarrow$ REJECT

$m = e / u_1^z$

Eliptic Curve Pederson Commitments

Setup(1^λ) Construct $(E_p, q, G) = ECGen(1^\lambda)$ with prime order q and generator G

Choose $a \xleftarrow{R} \mathbb{Z}_q$

compute $H = aG$ → another Generator

output (E_p, q, G, H)

Commit($x \in \mathbb{Z}_q$)

Choose $r \xleftarrow{R} \mathbb{Z}_q$

compute $c = xG + rH$

$c = g^x h^r$ for El Gamal

Reveal($x, r \in \mathbb{Z}_q, c$)

checks $c == xG + rH$

Perfect Hiding

$xG + rH = x'G + rH$

There is a way to reveal your value such that

it reveals any value => Information Theoretic Secure

6.857 Lecture

Wednesday, April 6, 2016 11:09 AM

Bilinear Maps

Gap groups + bilinear maps
BLS signatures
3-way key agreement
IBE (Identity based encryption)

"Decision Diffie Hellman"

Gap group: DDH is easy (to decide if (g, g^a, g^b, g^c) easy to decicle if $a \cdot b = c \pmod{|G|}$

Gap ↻ but CDH is hard (to compute g^{ab} given g, g^a, g^b

note: CDH easy ⇒ DDH easy

Use two groups: New properties

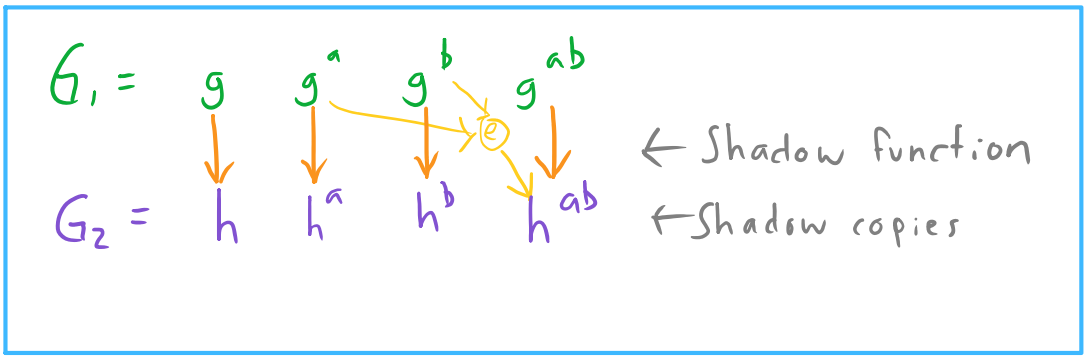
G_1 group $|G_1| = q$ generator g

G_2 ← Shadow Group $|G_2| = q$ generator h

$e : G_1 \times G_1 \to G_2$ ← "Shadow Function"

$(\forall a, b)\ e(g^a, g^b) = h^{ab}$ moved to the other group

$e(g, g) = h \quad e(g, g^a) = h^a$



With this setup DDH is easy:

$$e(g^a, g^b) \stackrel{?}{=} e(g, g^c)$$

Thm If you can compute DL in G_2 easily then you can compute them easily in G_1.

$$DL_{G,g}(g^a) = DL_{G_2,h}(h^a) = a$$

How to create a gap group? (complicated)

G_1 = supersingular eliptic curve $p = 2 \pmod 3$ $a = 0$

$G_2 = F_{p^k}$ small k $y^2 = x^3 + ax + b$ $b \in \mathbb{Z}_p^*$

↳ may use subgroups of order q for some prime $q \approx 2^{160}$

e = "Weil pairing", "Tate pairing"

BLS Signatures (2001) ← Short signatures

Keygen G_1, G_2 prime order $q \approx 2^{160}$

$e: G_1 \times G_1 \to G_2$

H: string $\to G_1$ CR

Secret: $x \in \mathbb{Z}_p^*$

Public: $g^x = y$

Sign(M) $m = H(M)$ ← representative in the group

$\sigma = \sigma_x(M) = m^x$ ← can be represented compactly

Verify (M, y, σ) $m = H(M)$

$e(m, g^x) = e(g, m^x)$

$e(g, m)^x$

Thm: BLS secure against existential forgery under Adaptive Chosen Message attack in ROM assuming CDH is hard in G_1.

3-way key agreement

Recall Dlt Key agreement: $g^a, g^b \to g^{ab}$ (only 2-way)

A publishes g^a
B " g^b
C " g^c

g^{a+b+c}

~~g^{abc}~~ $h^{abc} : e(g^{ab}, g^c) = h^{abc}$

Thm: Secure assuming BDH (Bilinear Diffie Hellman)

Given g, g^a, g^b, g^c, e hard to compute $e(g,g)^{abc}$

Identity Based Encryption (IBE)

MIT = Trusted third party
only need name to encrypt

TTP PK $y = g^x$

H_1 : string $\to G_1$

H_2 : G_2 to $\{0,1\}^*$ (PRG)
↳ Pseudo Random Generator.

y, name, M

$r \xleftarrow{R} \mathbb{Z}_q^*$

$Q_A = H(\text{name})$

$(g^r, M \oplus H_2(e(Q_A, y)^r))$

$d_A = Q_A^s$ ← Secret key given by MIT

Receive (U, V)

$$V \oplus H_2(e(d_B, U))$$
$$= V \oplus H_2(e(Q_A, g^r))$$
$$= V \oplus H_2(e(Q_A, y)^r)$$
$$= M$$

Key management.

6.857 Review

Monday, April 11, 2016 7:06 PM

Security Principles
Encryption
Hashing
Block Ciphers
Block Cipher Modes

Quiz I Review

Security Principles

Adversary has advantage. Only needs to find one vulnerability

Encryption

Information Theoreic

One Time Pad
malleable!

Can't break even w/ infinite computational power

IND-CPA (semantic security)

A can't distinguish b/w ciphertexts
access to encryption oracle.

must be randomized!

IND-CCA

A can access decryption oracle

IND-CCA 2

throughout challenge

A can access decrypt + encrypt
- Needs non-malleability

TODO CHECK

Hashing

non invertability / pre-image resistance

OW: Given $y = h(x)$, hard to find (x)

Good: Passwords in database

CR Hard to find any $x \neq x'$ st. $h(x) = h(x')$

CR ⇒ TCR TCR ⇏ CR

TCR Given $y = h(x)$ hard to find $x' \neq x$ s.t. $h(x') = y$

Good: Digital Signatures

Pseudo Randomness

Sample Question:

$H(x) = CBC\_MAC_k(x) \,||\, k$ secure? (OW)

No, the function is invertable. (Key is known)

2010 T/F #13

If H is CR ⇒ must be hard to compute the last bit of x given H(x)

False: set $h'(x_1 \ldots x_n) = h(x) \,||\, x_n$

Finding collisions in h' ⇒ Find collision in h

Block Ciphers

Ideal block cipher creates random permutations over fixed length message-space.

- invertible

DES - 56 bit keys

AES - 128, 196, 256 bit keys

Sample questions

2011 T/F #9

if AES decrypts a block c then someone must have encrypted a plaintext to obtain c.

False

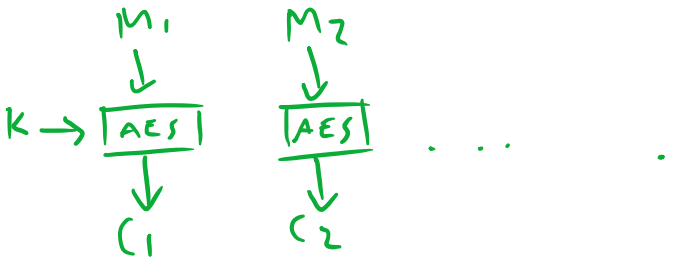
Block Cipher Modes

ECP (Electronic Code Book)

Every block encrypted in parallel w/ same key and same cipher.

Problem: Repeated Message block revealed

=> Not IND-CPA

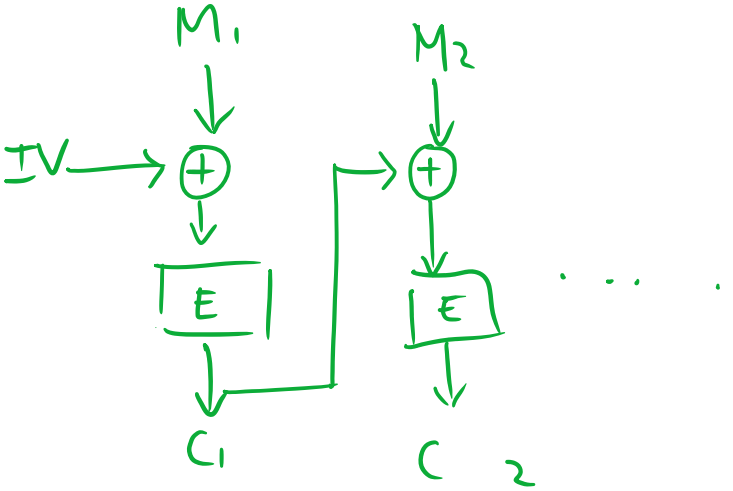


CBC (Cipher Block Chain)

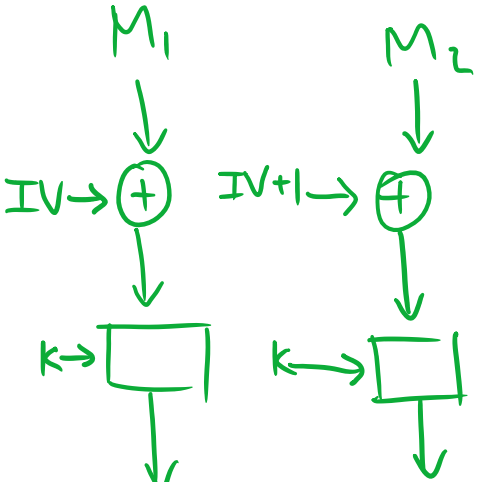
Random IV => IND-CPA

otherwise

static IV => deterministic



CTR



Sample Question

ECB but first half of every message block is random

IND-CCA2. ← Access to decryption throughout channel

False: can change 1 block of challenge ciphertext and decrypt.

Number Theory

Quadratic Residue

A quadratic residue r mod p is a number s.t. the equation $x^2 = r \pmod p$ has a solution.

+ 0 always quadratic residue

+ Have exactly $\frac{(p+1)}{2} + 1$ QR mod p (From 0)

+ **(i) (3 pts)** Explain how to tell if y is a quadratic residue modulo a prime p, given only a generator g for Z_p^* and the discrete logarithm x of y (so that $y = g^x \pmod p$).

Solution: y is a quadratic residue if x is even.

Sample Question

$ax^2 + bx + c = 0 \pmod p$ $\quad a, b, c \in \{0, 1, 2, \ldots, p-1\}$

Question: How many of these equations have solutions?

Answer:

- $a=0, b=0, c=0 \Rightarrow \exists$ infinetely many solns
- $a=0, b \neq 0, c=?$ $(p-1)(?)$ ← pick c

- $a=0, b=0, c=0 \Rightarrow \exists$ infinetely many solns
- $a=0, b \neq 0, c=p$: $(p-1)(p)$ ← pick c, pick b, all have solution
  - $(bx+c) \equiv 0 \pmod p$
  - $x \equiv b^{-1}(-c) \pmod p$

Diffie-Hellman Key Exchange

Alice ⟷ Bob

Eve

Broken if E performs a man in the middle attack

① Publicly agree on $G = \langle g \rangle$ $|G| = n$

Alice chooses secret integer a and sends g^a → Bob chooses secret b and sends g^b

Alice calculates g^{ab} ← Bob calculates g^{ab}

CDH: Given g^a and g^b it is computationally infeasable to calculate g^{ab}

DDH Given (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) it is infeasable to prove $g^c = g^{ab}$

El Gamal Encryption

Choose G s.t $|G| = q$ and generator g.

Alice

$x \xleftarrow{rand} \{1, 2, \ldots q-1\}$

$h = g^x$

$PK = (h, G, q, g)$

$SK = x$

Bob

$m \in G$

$y \xleftarrow{rand} \{1, 2 \ldots q-1\}$

$s = h^y$

$c_1 = g^y$

$c_2 = m \cdot s = m h^y$

$(c_1, c_2) = (g^y, m h^y)$

$= (g^y, m \cdot g^{xy})$

IND-CPA secure

Not IND-CCA2 secure

The El-Gamal signature scheme is not secure if multiple messages are signed with the same secret key x and the same random seed k. Show how to recover x given two messages m_1, m_2 and their signatures $(r_1 = g^k, s_1 = (\text{HASH}(m_1) - r_1 x)k^{-1}), (r_2 = g^k, s_2 = (\text{HASH}(m_2) - r_2 x)k^{-1})$.

Solution: s_1 and s_2 are a system of two equations with two unknowns which you can solve to recover x and k. $k = \frac{s_2 - s_1}{\text{HASH}(m_2) - \text{HASH}(m_1)}$, and $x = \frac{\text{HASH}(m_1) - k s_1}{r_1}$.

Eliptic Curves

$E: y^2 = x^3 + ax + b \pmod p$

$G = \{(x,y) : y^2 = x^3 + ax + b \pmod p\} \cup \{\infty\}$

- If $(x,y) \in G_E \Rightarrow (x,-y) \in G_E$
- Order of an elliptic curve group can be calculated efficiently.

Fermat's Little Theorem: $a^q \equiv a \pmod q$

MACs: Need to be hard to forge
Don't need to be pseudo random

Probability of finding 3-way collision: 2^{2n/3}

a^{p−1} = 1 mod p for all primes p.

6.857 Lecture

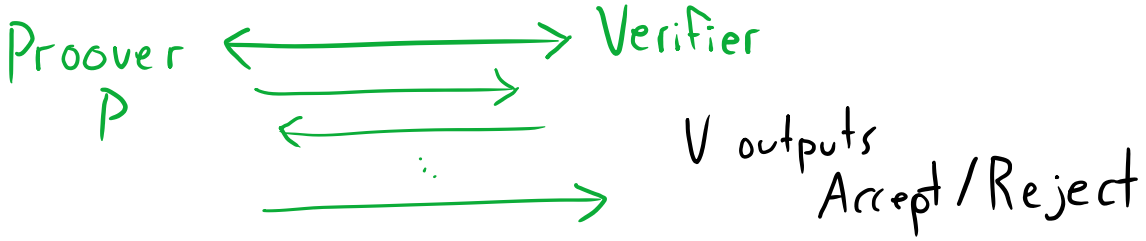
Monday, April 11, 2016 11:06 AM

Zero Knowledge Proofs

ZK Proofs
Interactive Proofs
Sudoku, 3 coloring, ham cycle, DL
Any problem in NP has ZK Proofs

Prove that you know the solution to a problem without revealing any info about the solution

Interactive Protocol



$$x = (\exists w)\, R(y, w)$$

- w: solution P knows
- y: Problem
- solution

Desired Properties

Completeness: if x true, V accepts

Soundness: if x not true, V rejects w/ $p \geq \epsilon$ (depends on problem)

ZK: if x true verifier learns nothing about w

If no solution, V accepts w/ prob $\leq (1-\epsilon)$

iterate t times: $\leq (1-\epsilon)^t \to 0$ as $t \to \infty$

commitment: $c = commit(v, r)$ Pedersen: $c = g^v h^r$

$open(c) \to (v, r)$

perfectly hiding ← Information Theoretic

computationally binding

6.857 Recitation

Friday, April 15, 2016 11:08 AM

Zero Knowledge Proof

Prover, Verifier

Prover wants to convince V that they know a solution to a poly-time algorithm without revealing anything about the solution.

Properties

Complete: If P knows solution, can always convince V.

Sound: If P doesn't know a solution, cannot convince V.

Zero-Knowledge: During the protocol, V gains no info about solution.

3 color in ZK Proof

P V

Coloring c on G

commit to permuted coloring →

← request opening of 2 adj colors

Opens 2 colored nodes

completeness: valid coloring ⇒ adj nodes always diff.

soundness: Invalid coloring ⇒ at least one 'bad' edge
$Pr = 1/|E|$
opens same colors

Zero Knowledge: Simulate by picking two random adjacent nodes and two random adjacent colors

Hamiltonian Cycle

↳ Path that visits every node exactly once.

P V

Knows cycle C on G

Create isomorphic graph H and commit.
Commit to cycle on H. ⟶ Randomly ask P to reveal one.

Completeness: IF P knows cycle in G
can compute mapping in H.

Soundness: IF P doesn't really know...
Either lied about H or not a valid H cycle.

Zero Knowledge:

DLP

$y = g^x \pmod p$ $\quad DLP(y) = x$

Schnorr Group
- $\mathbb{Z}_p^*$ p prime
$p = rq + 1$ q is prime.

Generator g with order q.

P V

Draws random k from $\mathbb{Z}_q$

Commits $(a = g^k)$ ⟶ Draws random value from $\mathbb{Z}_q = c$

⟵

sends $s = k + cx$ ⟶

Verify that $g^s = ay^c$

Completeness: $g^s = ay^c$

$g^{k+cx} = g^k g^{xc}$

$g^{k+cx} = g^{k+cx}$ ✓

Soundness: Show that if P can succeed for any two different challenges c and c' then P can compute x anyways.

$$s = k + cx$$
$$s' = k + c'x$$
$$s - s' = x(c - c')$$
$$x = (s - s')/(c - c')$$

Zero Knowledge:

$c \leftarrow \mathbb{Z}_q$
$s \leftarrow \mathbb{Z}_q$

$a = g^s y^{-c}$

$g^{k+cx} g^{-cx} = g^k$

NP $\in$ ZK

Can reduce any problem in NP to 3-coloring and proove using ZK.

NPC and $\in$ ZK

6.857 Lecture

Wednesday, April 20, 2016 11:17 AM

Voting

Evidence Based Voting

need convincing evidence that winner really won

Security Requirements

- only eligible voters may vote
- each cast vote is secret
  - no vote-selling
  - no receipt showing how you voted
- Final outcome is verifiably correct
- No "trusted parties". all are suspects = vendors... voters... officials ...

Software Independance

- Software is not to be trusted
- software independance. undetected error in software can not cause undetectable change in results.

Black-box auditing

Audit any voting systems

↳ any social choice function

6.857 Lectures

Monday, April 25, 2016 11:06 AM

Quines

Quines
Decidability / Anti-Virus Protection

Certificates
SPKI / SDSI

① Can you write a program that prints itself?

```
char *s = "char *s = %c%s%c; main(){
    printf(s,34,s,34);}";
main(){
    printf(s,34,s,34);
}
```

↑ double quote

YES

② Write a program that puts it's own source code into a string constant

✓ Program can operate on its source code as data.

$$P \equiv \begin{cases} s = \text{source code for } P \\ \text{define } A(x) \equiv \\ A(s) \end{cases} \qquad P = A(P)$$

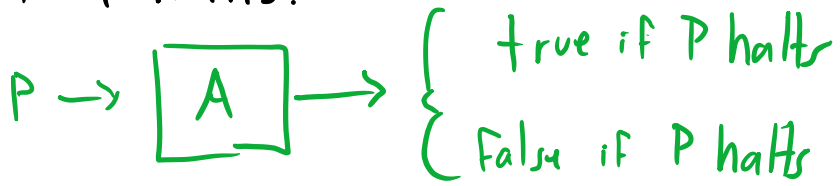
Halting Problem

Tell if a given program will ever stop.

Thm: Halting Problem is undecidable

⇒ There does not exist a program A that takes as input another program P s.t. A always halts and correctly

that takes as input another program P. s,t. A always halts and correctly outputs if P halts.



Proof (by contradiction)

Assume A exists and have code/access

$$P = \begin{cases} \text{if } A(P) \text{ then loop} \\ \text{else halt:} \end{cases}$$

□

Malware Detection

Rice's Thm: Determining any non-trivial property of output behavior is undecidable.

Proof

Suppose A can decide if program P has the property. (ie P is ransomware)

$$P = \begin{cases} \text{if } A(P) \text{ then } X'() & \leftarrow \text{does not have property} \\ \text{else} \quad X() & \leftarrow \text{has property} \end{cases}$$

Corollary: Virus detection is undecidable.

$\Rightarrow (\forall A)\ (\exists P)\ A(P)$ is wrong

$\Rightarrow (\exists P)\ (\forall A)\ A(P)$ is wrong