CSE 207B: Applied Cryptography

Nadia Heninger

UCSD

Spring 2022 Lecture 1

Course Mechanics

- Course Web Page: `https://cseweb.ucsd.edu/classes/sp22/cse207B-a/`
  - Schedule, assignments, etc.
- Lectures: 3:30-4:50pm Pepper Canyon Hall 120
- Canvas: `https://canvas.ucsd.edu/courses/35984`
  - Gradebook
- Piazza: `https://piazza.com/class/spring2022/cse207b`
  - Asynchronous Q&A
- My office hours: Tuesday 2-3pm EBU3B 3138/outside CSE
- TA: Adam Suhl: Wednesday 2-3pm outside CSE

Assignments and grading

70% Homework assignments

- Approximately 8 assignments, one a week
- Both programming and written exercises
- Submit to Gradescope by deadline

10% Peer grading

- We will use peer grading for homeworks
- You will grade one assignment (in a group)
- Work with Adam on rubric and grading
- Sign up after first assignment is due and enrollment becomes clearer

20% Final exam

- Take-home; June 6

Course policies

- Collaboration policy:
  - Talk to your classmates about approaches, write your own code and solutions.
  - This is a grad class: Cite your sources.
- Late policy:
  - We're all in this together, and we're all adults.
  - Please turn in your work on time and try not to push boundaries.
  - That said, if you have further extenuating circumstances let me know and we'll work something out.
- Pandemic policy:
  - This is a fully in person class
  - Follow campus covid rules/symptom checker/etc.

Textbook and other resources

We will mostly be following *Applied Cryptography* by Boneh and Shoup.

- Draft available here: `https://toc.cryptobook.us/`
- *Introduction to Cryptography* by Katz and Lindell also covers a lot of the same material.

Links to other textbooks and lots of papers for additional reading on web site.

Topics Covered

- Cryptographic security notions
  - Security definitions and reductions between concepts
  - Attacks exemplifying security notions
  - Focus on cryptography in broad use
- Symmetric cryptography
  - Stream ciphers
  - Block ciphers, modes of operation, chosen plaintext attacks
  - Message integrity, chosen ciphertext attacks
  - Hash functions, constructions and collision attacks
- Public-key cryptography
  - Basic computational number theory
  - Diffie-Hellman key exchange, discrete log security and attacks
  - RSA encryption, factoring hardness and attacks
  - Digital signatures: RSA, (EC)DSA, practical attacks

Topics *not* covered

- Cryptocurrencies
- Blockchains
- NFTs
- ...

Prerequisites

- Mathematical maturity, CS-style reductions and proofs
- You should have done well in your algorithms and complexity classes
- You will be happier in the second half of the course if you know what a group and field are.
- Programming challenges will be in Python3/Sage. If you don't know it already you can pick it up.

My work: Cryptographic systems security

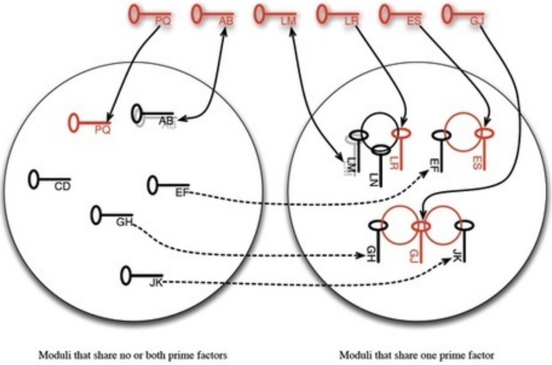

4LD

Crypto shocker: four of every 1,000 public keys provide no security (updated)

Almost 27,000 certificates used to protect webmail, e-commerce, and other …

by **Dan Goodin** - Feb 15 2012, 7:00am EST

Share Tweet 68



Keys that share one prime factor are vulnerable to cracking by anyone. Keys that share both prime factors can be broken by the other holder.

Researchers reveal a method the NSA may use to spy on Web traffic

By Sean Sposito | October 21, 2015 | Updated: October 21, 2015 5:05pm

1

RISK ASSESSMENT —

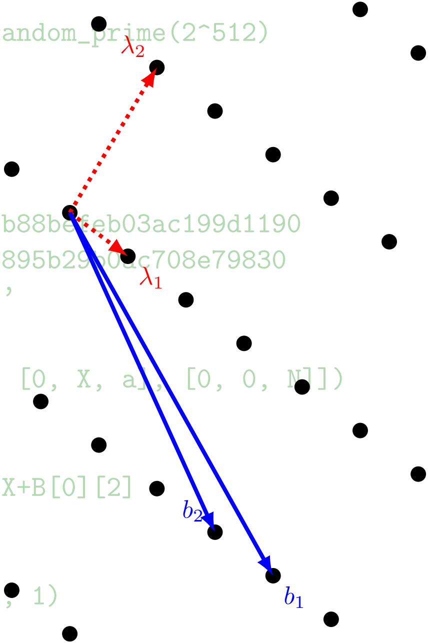
NSA could put undetectable “trapdoors” in millions of crypto keys

Technique allows attackers to passively decrypt Diffie-Hellman protected data.

DAN GOODIN - 10/11/2016, 7:30 AM

My work: Mathematical cryptography

```
p = random_prime(2^512); q = random_prime(2^512)
N = p*q

a = p - (p % 2^86)

sage: hex(a)
'a9759e8c9fba8c0ec3e637d1e26e7b88befeb03ac199d1190
76e3294d16ffcaef629e2937a03592895b29b0ac708e79830
4330240bc000000000000000000000'

X = 2^86
M = matrix([[X^2, 2*X*a, a^2], [0, X, a], [0, 0, N]])
B = M.LLL()

f = B[0][0]*x^2/X^2+B[0][1]*x/X+B[0][2]

sage: f.factor()[0]
(x - 2775338500016599864377709, 1)
```

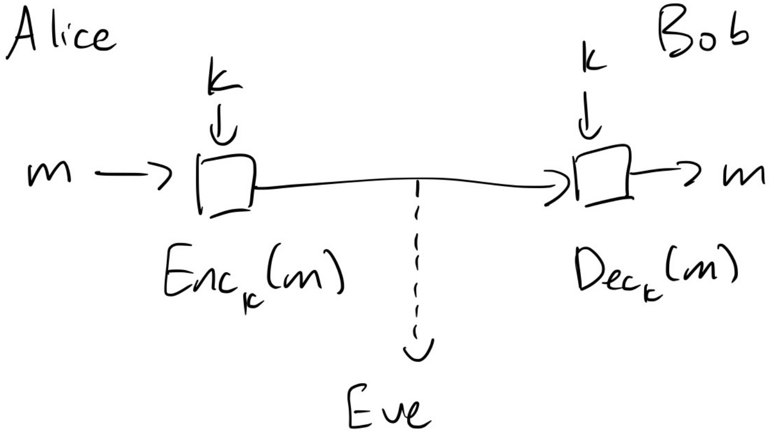


We'll see a lot of this stuff in this class.

Today's topics

- A little bit of classical cryptanalysis
- One-time pad

Cipher syntax



- Key generation: Generate key $k \in K$
- Encryption: $c = \mathrm{Enc}_k(m)$
- Decryption: $m = \mathrm{Dec}_k(c)$
- Correctness: $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$

Example: Substitution cipher

Alphabet $\Sigma = \{A, \ldots, Z, \sqcup\}$

Key generation: Choose random permutation $k : \Sigma \to \Sigma$

Encryption: $\mathrm{Enc}_k(m) = k[m[0]], k[m[1]], \ldots, k[m[\ell - 1]] \quad m \in \Sigma^\ell$

Decryption: $\mathrm{Dec}_k(c) = k^{-1}[c[0]], k^{-1}[c[1]], \ldots, k^{-1}[c[\ell - 1]]$
$c \in \Sigma^\ell$

k^{-1} is inverse permutation of k

Example: Substitution cipher

Alphabet $\Sigma = \{A, \ldots, Z, \textvisiblespace\}$

Key generation: Choose random permutation $k : \Sigma \to \Sigma$

Encryption: $\mathrm{Enc}_k(m) = k[m[0]], k[m[1]], \ldots, k[m[\ell - 1]] \quad m \in \Sigma^\ell$

Decryption: $\mathrm{Dec}_k(c) = k^{-1}[c[0]], k^{-1}[c[1]], \ldots, k^{-1}[c[\ell - 1]]$
$c \in \Sigma^\ell$

k^{-1} is inverse permutation of k

- Correctness?

Example: Substitution cipher

Alphabet $\Sigma = \{A, \ldots, Z, \sqcup\}$

Key generation: Choose random permutation $k : \Sigma \to \Sigma$

Encryption: $\mathrm{Enc}_k(m) = k[m[0]], k[m[1]], \ldots, k[m[\ell - 1]] \quad m \in \Sigma^\ell$

Decryption: $\mathrm{Dec}_k(c) = k^{-1}[c[0]], k^{-1}[c[1]], \ldots, k^{-1}[c[\ell - 1]]$
$c \in \Sigma^\ell$

k^{-1} is inverse permutation of k

- Correctness?
- Cryptanalysis?

Example: Substitution cipher

Alphabet $\Sigma = \{A, \ldots, Z, \textvisiblespace\}$

Key generation: Choose random permutation $k : \Sigma \to \Sigma$

Encryption: $\mathrm{Enc}_k(m) = k[m[0]], k[m[1]], \ldots, k[m[\ell - 1]] \quad m \in \Sigma^\ell$

Decryption: $\mathrm{Dec}_k(c) = k^{-1}[c[0]], k^{-1}[c[1]], \ldots, k^{-1}[c[\ell - 1]]$
$c \in \Sigma^\ell$

k^{-1} is inverse permutation of k

- Correctness?
- Cryptanalysis?
  - Frequency analysis
  - Brute force?

Example: Substitution cipher

Alphabet $\Sigma = \{A, \ldots, Z, \sqcup\}$

Key generation: Choose random permutation $k : \Sigma \to \Sigma$

Encryption: $\mathrm{Enc}_k(m) = k[m[0]], k[m[1]], \ldots, k[m[\ell - 1]] \quad m \in \Sigma^\ell$

Decryption: $\mathrm{Dec}_k(c) = k^{-1}[c[0]], k^{-1}[c[1]], \ldots, k^{-1}[c[\ell - 1]]$
$c \in \Sigma^\ell$

k^{-1} is inverse permutation of k

- Correctness?
- Cryptanalysis?
  - Frequency analysis
  - Brute force? No: $27! \approx 2^{93}$

Caesar Cipher in the wild

- 2006: Mafia boss Bernardo Provenzano used Caesar cipher encoded with numbers
  - http://itre.cis.upenn.edu/~myl/languagelog/archives/003049.html
- 2011: Rajib Karim plotted to blow up airplanes using Excel Caesar cipher
  - https://www.theregister.co.uk/2011/03/22/ba_jihadist_trial_sentencing/

Kerckhoff's Principle

Auguste Kerckhoff 1883

- Encryption scheme should not be a secret
- Only key needs to be a secret
- For modern cryptography: Algorithms should be public, standardized, and scrutinized in public

One-time pad

Alphabet: $\{0,1\}$

Key generation: $k \in_R \{0,1\}^\ell$

Encryption: $m \in \{0,1\}^\ell \quad c = k \oplus m$

Decryption: $m = c \oplus k$

Correctness?

One-time pad

Alphabet: $\{0, 1\}$

Key generation: $k \in_R \{0, 1\}^\ell$

Encryption: $m \in \{0, 1\}^\ell \quad c = k \oplus m$

Decryption: $m = c \oplus k$

Correctness? $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = k \oplus m \oplus k = m$

What does “secure” mean?

What does "secure" mean?

- No adversary can compute secret key from ciphertext?
- No adversary can compute plaintext from ciphertext?
- No adversary can determine a character of plaintext?
- No adversary can derive meaningful information from ciphertext?
- No adversary can compute any function of plaintext from ciphertext?

Perfect secrecy

Definition (Perfect secrecy)

Let (Enc, Dec) be a cipher over (K, M, C).

Let k be a random variable uniformly distributed over K.

$\forall m_0, m_1 \in M, \forall c \in C$:

$$\Pr[\mathrm{Enc}_k(m_0) = c] = \Pr[\mathrm{Enc}_k(m_1) = c]$$

Theorem

The following are equivalent:

(1) *(Enc, Dec) is perfectly secure.*

(2) $\forall c \in C, \exists N_c$ *s.t.* $\forall m \in M, |\{k \in K | \, \mathrm{Enc}_k(m) = c\}| = N_c$

(3) *Let k be a random variable uniformly distributed over K. Then $\mathrm{Enc}_k(m)$ has the same distribution for all m.*

Theorem

The following are equivalent:

(1) *(Enc, Dec) is perfectly secure.*

(2) $\forall c \in C, \exists N_c$ *s.t.* $\forall m \in M, |\{k \in K | \operatorname{Enc}_k(m) = c\}| = N_c$

(3) *Let k be a random variable uniformly distributed over K. Then $\operatorname{Enc}_k(m)$ has the same distribution for all m.*

Proof.

(2) $\implies$ (3) $\Pr(\operatorname{Enc}_k(m) = c) = N_c/|K|$

Theorem

The following are equivalent:

(1) *(Enc, Dec) is perfectly secure.*

(2) $\forall c \in C, \exists N_c$ *s.t.* $\forall m \in M, |\{k \in K | \, \mathrm{Enc}_k(m) = c\}| = N_c$

(3) *Let* k *be a random variable uniformly distributed over* K*. Then* $\mathrm{Enc}_k(m)$ *has the same distribution for all* m*.*

Proof.

(2) $\implies$ (3) $\Pr(\mathrm{Enc}_k(m) = c) = N_c/|K|$

(1) $\implies$ (2) Choose m_0. By (1),

$$\Pr[\mathrm{Enc}_k(m_0) = c] = \Pr[\mathrm{Enc}_k(m) = c] = P_c = N_c/|K|$$

for any $m \in M$.

Theorem

The following are equivalent:

(1) *(Enc, Dec) is perfectly secure.*

(2) $\forall c \in C, \exists N_c$ *s.t.* $\forall m \in M, |\{k \in K | \operatorname{Enc}_k(m) = c\}| = N_c$

(3) *Let k be a random variable uniformly distributed over K. Then $\operatorname{Enc}_k(m)$ has the same distribution for all m.*

Proof.

(2) $\implies$ (3) $\Pr(\operatorname{Enc}_k(m) = c) = N_c/|K|$

(1) $\implies$ (2) Choose m_0. By (1),

$$\Pr[\operatorname{Enc}_k(m_0) = c] = \Pr[\operatorname{Enc}_k(m) = c] = P_c = N_c/|K|$$

for any $m \in M$.

(3) $\implies$ (1) Fix $m_0, m_1 \in M$, $c \in C$. By (2),

$$\Pr[\operatorname{Enc}_k(m_0) = c] = P_c = \Pr[\operatorname{Enc}_k(m_1) = c]$$

$\square$

Theorem

The one-time pad is perfectly secret.

Theorem

The one-time pad is perfectly secret.

Proof.

Fix $m \in \{0,1\}^\ell$, $c \in \{0,1\}^\ell$.

$\exists$ unique $k \in \{0,1\}^\ell$ satisfying $k \oplus m = c$, $k = m \oplus c$.

Thus $N_c = 1$ in above theorem, part (2).

$\square$

Example: Variable-length one-time pad

$k \in \{0,1\}^L, m, c \in \{0,1\}^{\ell \leq L}$

$\mathsf{Enc}_k(m) = k[0,\ldots,\ell-1] \oplus m$

$\mathsf{Dec}_k(c) = k[0,\ldots,\ell-1] \oplus c$

Example: Variable-length one-time pad

$k \in \{0,1\}^L, m, c \in \{0,1\}^{\ell \leq L}$

$\mathsf{Enc}_k(m) = k[0, \ldots, \ell - 1] \oplus m$

$\mathsf{Dec}_k(c) = k[0, \ldots, \ell - 1] \oplus c$

Let m_0 have length 1, m_1 have length 2.

c has length 1.

Example: Variable-length one-time pad

$k \in \{0,1\}^L, m, c \in \{0,1\}^{\ell \leq L}$

$\mathsf{Enc}_k(m) = k[0,\ldots,\ell-1] \oplus m$

$\mathsf{Dec}_k(c) = k[0,\ldots,\ell-1] \oplus c$

Let m_0 have length 1, m_1 have length 2.

c has length 1.

$\Pr[\mathsf{Enc}_k(m_0) = c] = 1/2 \qquad \Pr[Enc_k(m_1) = c] = 0$
$\implies$ Not perfectly secret.

Theorem (Shannon)

(Enc, Dec) perfectly secret $\implies |K| \geq |M|$.

Theorem (Shannon)

(Enc, Dec) perfectly secret $\implies |K| \geq |M|$.

Proof.

Show $|K| < |M| \implies$ not perfectly secret.

Choose arbitrary m_0, k_0. Let $c = \mathrm{Enc}_{k_0}(m_0)$.

Let $S = \{\mathrm{Dec}_k(c) | k \in K\}$.

$|S| \leq |K| < |M|$.

$\implies \exists m_1 \in M \setminus S$.

$$\left.\begin{array}{ll} \Pr[\mathrm{Enc}_k(m_1) = c] & = 0 \\ \Pr[\mathrm{Enc}_k(m_0) = c] & > 0 \end{array}\right\} \text{not perfectly secret}$$

$\square$

One-time pad too difficult to use in practice: Basically only used by spies.

Numerous usability problems, including reusing one-time pad material.

Recent example: Russian spies using Cuban “numbers” station, but there was a bug and dummy “fill” traffic was missing the number 9!

According to Matt Blaze, this let the FBI tell when messages were transmitted to the spies.

`https://www.mattblaze.org/blog/neinnines/`

Your first assignment is already online! Break “one-time pad” where pad is reused multiple times.

We recommend you start early.

Next time: Relax security definitions to be more usable in the real world.