Cryptography Presentation

George Zhang

August 2018

Outline

Introduction

Symmetric Cryptosystems

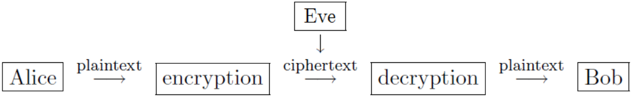
Asymmetric Cryptosystems

References

(BONUS) Diffie-Hellman

Introduction

- Cryptography is the study of secure communications in the presence of adversaries who wish to compromise messages.
- Prior to transmission, a plaintext message is encrypted into a ciphertext. On the receiver end, the ciphertext is then decrypted into the original plaintext message.



Eve

Alice plaintext encryption ciphertext decryption plaintext Bob

Figure 1: A high-level overview of an encryption system.

Introduction

Examples of usage:

- SSH/SSL protocol
  - HTTP(S)
  - Online banking
- Cryptocurrency
- Disk Encryption
- Protecting datacenters!

Introduction

Assume there exists a third party, Eve, who knows the transmission protocol Alice and Bob are using. Eve may have various goals:

- She wants to read a particular transmitted message.
- She wants to find the decryption key for the protocol, so that she can read all previously transmitted messages.[1]
- She wants to change Alice's message without Bob realizing it.
- She wants to send Bob a forged message and make it look like Alice sent it.

[1] Assuming no perfect forward secrecy

Introduction

A good cryptosystem must combat these attacks, given the possibility that:

- Eve only has the ciphertext.
- Eve also has a piece of the original plaintext sent by Alice.
- Eve is temporarily able to encrypt a message of her choice.
- Eve is temporarily able to use the decryption machine.

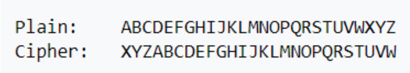
Cryptography is a very complex subject. For simplicity, this presentation will focus on the first case.

Symmetric Cryptosystems

- A symmetric cryptosystem uses the same cryptographic keys for both encryption and decryption. The keys may be the same, or there may exist a "simple" transformation to go between them.
- A simple example is the Caesar Cipher.
- It is not obvious, but there are actually cryptosystems that are (suspected to not be) symmetric. For example, it may be difficult to find the decryption key from the encryption key.

The Caesar Cipher

- Used by Julius Caesar to protect military messages.
- A symmetric cryptosystem.
- Plaintext messages are encrypted by shifting every nonspace character some fixed number of letters up or down the alphabet.
- Letters near the ends are circularly shifted to the other end.



```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVW
```

Figure 2: A Caesar Cipher using a left rotation of three places. A message of plaintext "HELLO ABC" would become "EBIIL XYZ".

The Caesar Cipher

An equivalent way of viewing Caesar Ciphers is to use modular arithmetic. Modular arithmetic will be discussed on this slide, and I will return to Caesar Ciphers on the next slide.

- A number system modulo n is restricted to integers in $\{0, 1, \cdots, n-1\}$
- Below zero and past n, the numbers "wrap around" on each other.
- Two numbers are congruent mod n if one number can be obtained from the other by repeatedly adding or subtracting n.
- For example, consider the 24-hour clock system: how do we convert 18:00 to the 12-hour system?

The Caesar Cipher

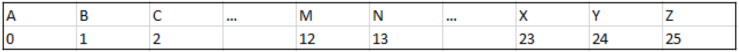
An equivalent way of viewing Caesar Ciphers is to use modular arithmetic. Modular arithmetic will be discussed on this slide, and I will return to Caesar Ciphers on the next slide.

- A number system modulo n is restricted to integers in $\{0, 1, \cdots, n-1\}$
- Below zero and past n, the numbers "wrap around" on each other.
- Two numbers are congruent mod n if one number can be obtained from the other by repeatedly adding or subtracting n.
- For example, consider the 24-hour clock system: how do we convert 18:00 to the 12-hour system?
- Use the integers modulo 12. It is 6:00 (PM), since, $18 - 1 * 12 = 6$. Succinctly, $18 \equiv 6 \pmod{12}$

The Caesar Cipher

Now, back to the Caesar Cipher:

- To begin formulating it in terms of modular arithmetic, each letter of the alphabet is assigned to a number, *starting at zero*. For example, $A = 0$, $B = 1$, ..., $Z = 25$.
- To shift a single plaintext character to the right by k, we can use the function $f : \mathbb{Z}/26\mathbb{Z} \to \mathbb{Z}/26\mathbb{Z}$
  $f(m) = m + k \pmod{26}$.



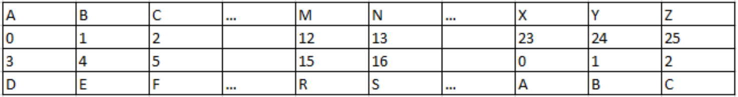
A	B	C	...	M	N	...	X	Y	Z
0	1	2		12	13		23	24	25

Figure 3: Conversion from text to integers.

The Caesar Cipher

Now, back to the Caesar Cipher:

- To shift a single plaintext character to the right by k, we can use the function $f : \mathbb{Z}/26\mathbb{Z} \to \mathbb{Z}/26\mathbb{Z}$
  $f(m) = m + k \pmod{26}$.
- For example, to shift $Z = 25$ right by 3,
  $f(25) = 25 + 3 \pmod{26} = 28 \pmod{26} \equiv 2 \pmod{26}$,
  which corresponds to the letter C.
- How about a left shift?



A	B	C	...	M	N	...	X	Y	Z
0	1	2		12	13		23	24	25
3	4	5		15	16		0	1	2
D	E	F	...	R	S	...	A	B	C

Figure 4: Row 1 - Alphabet. Row 2 - Integer conversion. Row 3 - Apply f to row 2. Row 4 - Cipher alphabet (cipher letter each plaintext letter gets converted to).

The Caesar Cipher

Problems with the Caesar Cipher.

- The transmitter and receiver must somehow agree beforehand the encryption function f to use.
  - This is a weakness of many traditional symmetric encryption algorithms.
  - In order to make this agreement, at some point they must decide on f across an insecure channel.
- It is easy to determine the decryption function from the encryption function.
  - If one is compromised, so is the other.
  - Given that the encryption function for a Caesar Cipher is $f(m) = m + k \pmod{26}$, the decryption function can easily be found as $f^{-1}(c) = c - k \pmod{26}$.
- Subject to brute-force attacks, or cryptanalysis based on occurrence of letters in the English alphabet.

Asymmetric Cryptosystems

- Also known as public-key cryptography.
- Asymmetric cryptosystems attempt to address some of the issues symmtric cryptosystems have.
  - There is still an "out-in-the-open" agreement, but we don't care if there are eavesdroppers.
- In an asymmetric cryptosystem, there are two keys:
  - The public key, which may be safely distributed publicly.
  - The private key, which must be kept a secret.
  - Everyone who wishes to receive messages must have both a public key and private key.

Asymmetric Cryptosystems

- Suppose Bob wishes to transmit a message to Alice.
- Bob uses Alice's public key to encrypt her message.
- Alice then uses her own private key to decrypt the message.

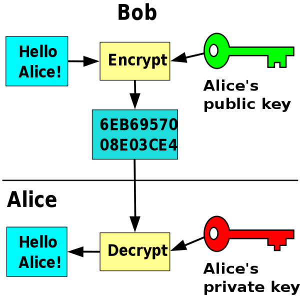


Figure 5: Visual example of public key encryption.

Asymmetric Cryptosystems

- Recall from the previous slide, the public key acts as the encryption function f and the private key as the decryption function f^{-1}.
- Problem: Given this function f, shouldn't it be possible to "efficiently" determine the inverse f^{-1}?

Asymmetric Cryptosystems

- Recall from the previous slide, the public key acts as the encryption function f and the private key as the decryption function f^{-1}.
- Problem: Given this function f, shouldn't it be possible to "efficiently" determine the inverse f^{-1}?
- ... for many of these functions used in asymmetric encryption, it is an open problem in computer science.
- Asymmetric encryption relies on the difficulty of this problems for certain functions; it is asymmetric in the fact that f is "easy" to compute but f^{-1} is believed to be "hard" to compute.

One-Way Functions

- In asymmetric encryption, the encrypting function f using the public key is said to be a *one-way* function.
- Example of one-way function: Integer Factorization
- Multiplication is easy
  - Given $f(p, q) = pq = n$, where p, q are prime, what is n?
- Factorization is (believed to be) hard
  - Given $f^{-1}(n) = (p, q)$, where n is composite, what are p, q?
- Cryptographic protocols are often expressed as using keys n with a certain bitlength[2]. What happens to the possible values of n when the bitlength is increased by one?

[2] In RSA, n is only one part of the key.

One-Way Functions

Table 1: $v = 2^b$

Bit length	**Value range**
16	6×10^4
64	1×10^{19}
463	2×10^{139}
512	1×10^{154}
768	1×10^{231}
1024	1×10^{308}
2048	3×10^{616}
3072	5×10^{924}

One-Way Functions

- The value (easily computed from p, q) n is part of the public key.
- The values (hard to figure out from n) p, q are part of the private key.

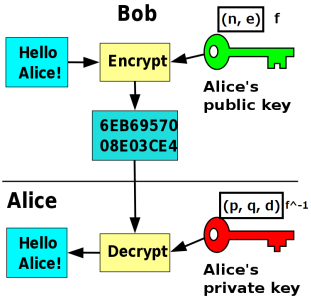


Figure 6: Visual example of public key encryption. For a detailed example of how this is actually done, see RSA.

Hash Functions

One way functions are *not necessarily* hash functions.

One-way function:

- A function that is ”easy” to compute, but believed to be ”hard” to reverse

(Ideal) Hash function:

- A one-way function, where the output is known as a ”hash”
- Extremely unlikely that two different plaintexts have the same hash
- A small change in the plaintext unpredictably changes the hash

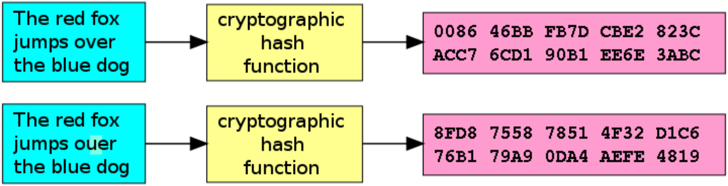
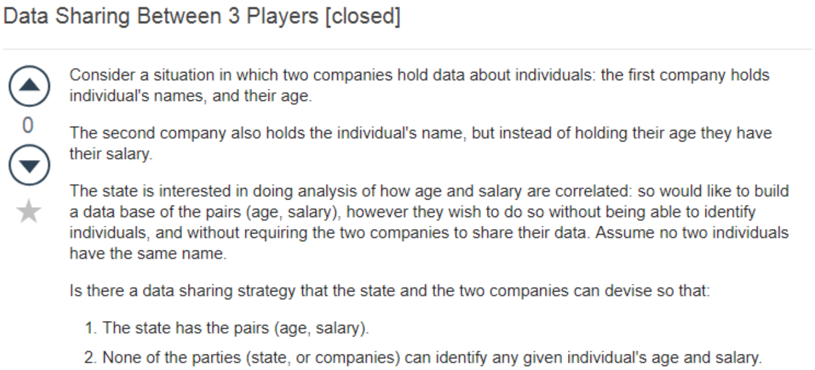

The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

Figure 7: SHA-1 Hash Function

One-Way Function Puzzle

A puzzle involving the use of (any) one-way functions



Data Sharing Between 3 Players [closed]

Consider a situation in which two companies hold data about individuals: the first company holds individual's names, and their age.

The second company also holds the individual's name, but instead of holding their age they have their salary.

The state is interested in doing analysis of how age and salary are correlated: so would like to build a data base of the pairs (age, salary), however they wish to do so without being able to identify individuals, and without requiring the two companies to share their data. Assume no two individuals have the same name.

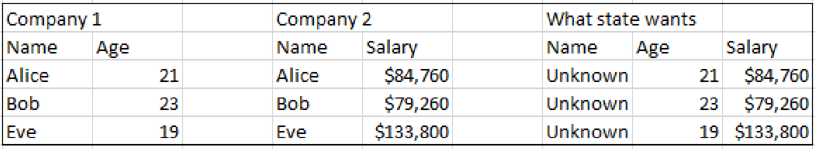
Is there a data sharing strategy that the state and the two companies can devise so that:

1. The state has the pairs (age, salary).
2. None of the parties (state, or companies) can identify any given individual's age and salary.

Figure 8: 3-way data sharing puzzle. The original problem formulation can be found on Puzzling Stack Exchange.

Data Sharing Between 3 Players

How would the state obtain this data?



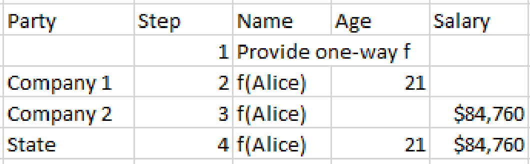
Company 1			Company 2			What state wants		
Name	Age		Name	Salary		Name	Age	Salary
Alice	21		Alice	$84,760		Unknown	21	$84,760
Bob	23		Bob	$79,260		Unknown	23	$79,260
Eve	19		Eve	$133,800		Unknown	19	$133,800

Figure 9: An visual schema of the puzzle. The companies are allowed to communicate but nobody must be able to obtain all 3 of the (name, age, salary) of an individual. Names are assumed to be unique.

Data Sharing Between 3 Players

(Incomplete) Solution

1. Have the state provide the same reliable one-way function f to both companies.
2. Company 1 encrypts their names with f, then provides (f(name), age) to the state
3. Company 2 encrypts their names with f, then provides (f(name), salary) to the state
4. The state joins these pairs on the encrypted names to get (age, salary).



Party	Step	Name	Age	Salary
	1	Provide one-way f		
Company 1	2	f(Alice)	21	
Company 2	3	f(Alice)		$84,760
State	4	f(Alice)	21	$84,760

Figure 10: Solution.

Data Sharing Between 3 Players

Summary: the state provide the same one-way function to both parties, have them encrypt their names using this function, then share this data for the state to join on the encrypted names. What's wrong?

Data Sharing Between 3 Players

Summary: the state provide the same one-way function to both parties, have them encrypt their names using this function, then share this data for the state to join on the encrypted names. What's wrong?

- It is subject to brute-force attacks since the state knows the plaintext names of all its citizens.

How can this be fixed?

Data Sharing Between 3 Players

Summary: the state provide the same one-way function to both parties, have them encrypt their names using this function, then share this data for the state to join on the encrypted names. What's wrong?

- It is subject to brute-force attacks since the state knows the plaintext names of all its citizens.

How can this be fixed?

- The companies request that the one-way function provided by the state be a cryptographically secure hash function.
- Before hashing the names, they agree on a scheme that modifies same names consistently, and do not inform the state of this scheme.

References

- Wikipedia's article on cryptography
- Dr. Manfred Kolster's notes on Cryptography
- Puzzling Stack Exchange
- Security Stack Exchange

Diffie-Hellman

Recall that classical symmetric encryption algorithms had a weakness:

- Had to exchange the (symmetric) key over an insecure channel

This problem was fixed with asymmetric encryptions using public keys and one-way functions. Actually, symmetric encryption can be made just as secure using similar number-theoretic techniques.

Diffie-Hellman

Diffie-Hellman Key Exchange:

- Rather than having one party generate and transmit a key, have both parties create the key together to obtain a shared secret.
- The attacker would not be able to recreate this shared secret through what is transmitted over the channel.

Diffie-Hellman

- In Diffie-Hellman, Alice and Bob first agree on a common paint, then come up with their own secret colors.
- They then each mix their secret color with common paint and send them over an insecure channel.
- They mix their own paint with the paint they were sent and arrive at the same color!

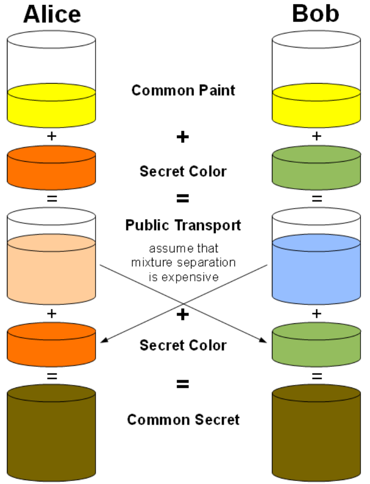


Figure 11: Illustration of the idea behind Diffie–Hellman key exchange

Diffie-Hellman

Diffie-Hellman Key Exchange:

- The common secret becomes the encryption/decryption key.
- An attacker Eve can try all she wants to separate the transported paints, but it takes too much effort.

Diffie-Hellman

Key mathematical ideas behind Diffie-Hellman Key Exchange:

- Pick g and p carefully. $(g^a \pmod{p})^b \pmod{p} = (g^b \pmod{p})^a \pmod{p} = g^{ab} \pmod{p}$
  - For example:
  - $5^4 \pmod{23} = 4$
  - $5^3 \pmod{23} = 10$
  - $10^4 \pmod{23} = 18$
  - $4^3 \pmod{23} = 18$
- g and p are the common paint. In the exponents, a is Alice's secret, b is Bob's secret.
- They both arrive at the shared secret $g^{ab} \pmod{p} = 18$.

Diffie-Hellman

Key mathematical ideas behind Diffie-Hellman Key Exchange:

- Pick g and p carefully. $(g^a \pmod{p})^b \pmod{p} = (g^b \pmod{p})^a \pmod{p} = g^{ab} \pmod{p}$
  - For example:
  - $5^4 \pmod{23} = 4$
  - $5^3 \pmod{23} = 10$
  - $10^4 \pmod{23} = 18$
  - $4^3 \pmod{23} = 18$
- 5 and 23 are the common paint. In the exponents, 4 is Alice's secret, 3 is Bob's secret.
- They both arrive at the shared secret $g^{ab} \pmod{p} = 18$.

Diffie-Hellman

Eve wants to obtain the shared secret, $g^{ab} \pmod{p}$. What information does Eve already have?

- She can intercept the common paint (5, 23), and the transported values 4 from Alice and 10 from Bob.
- i.e. she has g, p, $g^a \pmod{p}$, $g^b \pmod{p}$
- Recall that in the analogy, it was difficult to separate paints. In Diffie-Hellman, it is believed that a and b cannot be efficiently obtained from this data[3]. It is a "one-way function", but it was generated together by two participants.

[3] The most promising approach is known as the discrete logarithm problem

Diffie-Hellman

- Rather than finding a and b, perhaps try $g^a \pmod{p} \times g^b \pmod{p}$ to obtain $g^{ab} \pmod{p}$ directly?
- You would end up at $g^{a+b} \pmod{p}$, which is useless