CSE 5852: Lecture 1

Jonathan Huang

August 29, 2016

1 The class trajectory

In this class we will spend most of our time on three activities:

1. Formalizing a security notion.
2. Trying to build a mathematical object or protocol that can be used to satisfy this definition.
3. Providing a formal, mathematical proof that the object or protocol satisfies the definition we described. These proofs will usually come in one of two forms:
   (a) An unconditional proof of security. This is the strongest form of security and we will focus on it for the first month of the class. In this setting, the only things that are required to understand the security is to understand the definition and be sure the proof is correct.
   (b) A conditional proof of security. This proofs will be stated as *if A then B* where B is that the object or protocol satisfies our definition. In addition to the conditions for an unconditional proof one must understand whether the assumption A is reasonable. Informally, A should be a "simpler" assumption than B. (One could always just assume security of the object or protocol.) We will talk about some well studied assumptions that are used in cryptography and some newer assumptions that are currently being evaluated. Primarily we will consider assumptions on mathematical problems, however one can base cryptography on any well understood assumption, for example, hardware with certain properties, that the attacker is limited to a certain computational power, etc.

2 Creating a private channel

Suppose we have two people on the internet that would like to talk to each other securely. What does this mean to talk securely? In order to answer this question, we need to define both positive and negative functionality (security). For now we will consider a relatively simple example, we have two people that meet in person and want to be able to send a single, unknown message from the *sender* to the *receiver* at a later point in time.

What we've defined above is our functionality. We now need to consider security. We will assume for the moment that the sender and receiver are able to talk completely privately when they meet in person. We will ask for security when they attempt to send their single message over the internet. (Later on, we will revisit the assumption they can meet in person and talk with any interference.) When sending a secure message we may want the following properties:

- Secrecy: no one is able to tell what is sent
- Correctness: the sent message is what the receiver sees
- Delivery: the receiver sees the sent message
- Timeliness: the receiver sees the sent message immediately
- Authenticity: Sender identity is known to receiver
- Cannot tell when channel is in use

For multiple messages we may also want:

- In-Order Delivery: Messages are sent in order
- Subsequent messages don't reveal information about previous messages

For now we will consider a simple example, where two people meet in person and want to be able to send a single, unknown message from the *sender* to the *receiver* afterwards. Summarizing the situation:

1. *sender* and *receiver* can exchange information with no adversarial involvement (message to be determined).[1]
2. Assume that *sender* and *receiver* both share a *key* k
3. We have two algorithms:
   (a) Encrypt or $\mathsf{Enc}(k, m)$ which will produce c, the ciphertext
   (b) Decrypt or $\mathsf{Dec}(k, c)$ which will return m, the message, with the input k
4. Assume *adversary*, sees c but not k

2.1 Historial Ciphers

Consider the classic example of the Caesar cipher, aka rot13. Here we think of the twenty six English characters and space as numbers $0, 1, ..., 26$. An example encoding is in Table 1.

The shift is stored as our key. But there are some problems:

- The key space is small. There are 26 possible keys which can be used exhaustively (brute force) to learn the message.
- The shift is repeated across letters.

[1] We'll see later this is known as symmetric key cryptography.

Character	Number	New Number	New Character
_	0	13	m
a	1	14	n
b	2	15	o
c	3	16	p
d	4	17	q
e	5	18	r
f	6	19	s
g	7	20	t
h	8	21	u
i	9	22	v
j	10	23	w
k	11	24	x
l	12	25	y
m	13	26	z
n	14	$27 \equiv 0 \mod 27$	_
o	15	$28 \equiv 1 \mod 27$	a
p	16	$29 \equiv 2 \mod 27$	b
q	17	$30 \equiv 3 \mod 27$	c
r	18	$31 \equiv 4 \mod 27$	d
s	19	$32 \equiv 5 \mod 27$	e
t	20	$33 \equiv 6 \mod 27$	f
u	21	$34 \equiv 7 \mod 27$	g
v	22	$35 \equiv 8 \mod 27$	h
w	23	$36 \equiv 9 \mod 27$	i
x	24	$37 \equiv 10 \mod 27$	j
y	25	$38 \equiv 11 \mod 27$	k
z	26	$39 \equiv 12 \mod 27$	l

Table 1: Example encoding using Caesar cipher with $K = 13$

Original message	`This is a simple breakable message`
Encoded message	`fuvemvemnmevzbyrmodrnxnoyrmzreentr`

Table 2: Message encryption using Caesar cipher.

Original alphabet	`_abcdefghijklmnopqrstuvwxyz`
Encoded message	`yajvcmstieq_lbrfkozpnhgdxuw`

Table 3: Substitution or Vernam cipher.

- Statistical information about the English language allows us to rule out possible keys. This further reduces the possible key space.

Now we can use a more complicated substitution cipher where each letter is mapped to a random letter. Also known as the Vernam Cipher. An example is presented in Table 3. In this case, there are 26! possible keys. This has eliminated the problem of small key space. However, each letter is still mapped to the same letter, this means that statistical information about the language can still be used to learn the key and recover the message.

Assuming that *adversary* the will be able to see the ciphertext, let's list some desirable properties:

- No property of the message can be learned **This is the strongest notion**
- No property of the key can be learned
- Some part of the message is hidden
- There are multiple plausible messages
- The message is difficult to guess
- Important message properties can't be recovered
- The length of the message cannot be determined

Its important to note that in general we don't really care about the privacy of the key. This was not defined as part of our functionality. It may be necessary for the key to remain private, for example to send multiple messages. However, a scheme can be a good encryption and reveal information about the key.

The strongest of these requirements will be our basis for security:

Definition 1. *The adversary learns nothing about the message m from the ciphertext c.*

This definition is impossible to achieve. Since the two parties are meeting ahead of time, they may not know the length of the message to be sent. The size of the ciphertext must be at least the same size as the original message:

Original message	`Another message this one is much harder to break`
Encoded message	`arfnimzybmppatmyniepyfrmyepybhviyiazcmzynfyjzma_`

Table 4: Message encryption using Vernam cipher.

$|c| \geq |m|$. Even if the two parties use padding, the ciphertext length must reveal a bound on the message length. Thus, we slightly weaken the definition as follows:

Definition 2 (Perfect Secrecy)**.** *[SWBH49] The adversary learns nothing about the message m from the ciphertext c other than its length, $|m|$.*

While this definition is more specific, it is not formal enough to write a mathematical proof. In order to write a more formal definition, we will need to review a bit of probability theory.

References

[SWBH49] Claude E. Shannon, Warren Weaver, Richard E. Blahut, and Bruce Hajek. *The mathematical theory of communication*, volume 117. University of Illinois press Urbana, 1949.