CO 487 Course Notes

Applied Cryptography

Michael Socha

University of Waterloo

Winter 2019

Contents

1 Course Overview

This course is an applied introduction to modern cryptography. Topics covered include:

- Symmetric-key encryption
- Hash functions
- Authenticated encryption
- Public-key encryption
- Signature schemes
- Key establishment
- Key management
- Examples of deployed cryptography (e.g. SSL, cryptocurrencies, WPA)

2 Introduction - What is Cryptography?

Information security (also known as cybersecurity) deals with protecting information assets from unauthorized acquisition, damage, disclosure, manipulation, loss, or use. Cryptography deals with the mathematical, algorithmic and implementation aspects of information security.

Cybersecurity more broadly includes the study of computer security, network security and software security.

2.1 Goals of Cryptography

In short, cryptography is about securing communications in the face of malicious adversaries. When describing cryptographic scenarios, Alice and Bob are used to indicate two parties who wish to communicate with one another across some channel, while Eve is a malicious adversary. Eve may attempt to read or modify the data being transmitted.

The main goals of cryptography are to provide:

- **Confidentiality:** Keeps data secret from unauthorized entities.
- **Data integrity:** Ensures data has not been altered by unauthorized means.
- **Data origin authentication:** Determines the sender of data.
- **Non-repudiation:** Provides proof of data origin and integrity, which can be used to prevent senders from disputing a previous action.

3 Symmetric-Key Encryption

3.1 Introduction

A symmetric-key encryption scheme (SKES) consists of:

- M - the plaintext space
- C - the ciphertext space
- K - the key space
- a family of encryption functions - $\forall k \in K, E_k : M \to C$
- a family of decryption functions - $\forall k \in K, D_k : C \to M$, such that $\forall k \in K, \forall m \in M, D_k(E_k(m)) = m$

The idea behind using SKES to achieve confidentiality is:

1. Alice and Bob agree on a secret key k (note that this cannot be done over an unsecured channel, or Eve may read the key).
2. Alice computes $c = E_k(m)$, and sends Bob ciphertext c, which is an encrypted form of the plaintext m.
3. Bob computes $m = D_k(c)$ to retrieve the plaintext m.

3.2 SKES Security

A security model defines the computational power of an adversary, and how they interact with the communicating parties. It is generally assumes that the adversary knows everything about the SKES besides the value of k.

3.2.1 Types of Attacks (Adversary Interactions)

- **Passive Attacks:**
  - Ciphertext-only attack - Eve only knows some ciphertext generated by Alice and Bob.
  - Known-plaintext attack - Eve knows some plaintext and the corresponding ciphertext.
- **Active Attacks:**
  - Chosen-plaintext attack - Eve can choose some plaintext and obtain the corresponding ciphertext.
- **Other Attacks:**

- Clandestine attacks - Bribery, blackmail, etc.
- Side-channel attackers - Involves monitoring of encryption and decryption equipment.

3.2.2 Computational Power of Adversary

- **Information-theoretic security:** Eve has infinite computational resources.
- **Complexity-theoretic security:** Eve has a polynomial-time Turing Machine.
- **Computational Security:** Eve has some fixed amount of computing power.

3.2.3 Goal of Adversary

The adversary may try to:

1. Recover the secret key.
2. Systematically recover plaintext from ciphertext, though without necessarily knowing the key.
3. Learn some information about plaintext based on its ciphertext (other than its length).

If an adversary can succeed at 1 or 2, then the SKES is said to be completely/totally insecure. If none of these goals can be achieved, the SKES is said to be semantically secure.

3.2.4 Secure SKES

A SKES is said to be secure if it is semantically secure when attacked with a chosen-plaintext attack by a computationally-bounded adversary.

3.3 Desirable Properties for SKES

- Efficient algorithms should be known for encryption and decryption functions.
- The secret key should be large enough to render exhaustive key search infeasible, though still relatively small.
- The scheme should be secure against everyone, even against the designers of the system (no security through obscurity).

3.4 Security Level

A cryptographic schema is said to have a security level of l bits if the fastest known attack scheme takes around 2^l operations. For context, in this course, we consider 2^{40} operations very feasible, 2^{80} barely feasible, and 2^{128} infeasible. Security levels of 128 bits are desirable in practice.

3.5 Some Simple Ciphers

3.5.1 Simple Substitution Cipher

The simple substitution cipher involves replacing units of plaintext with ciphertext. For example, if the key is that a maps to Y and b maps to Z, then $m = abbba$ results in $c = YZZZY$.

This cipher is totally insecure against a chosen-plaintext attack, since the mapping can be easily determined. When the mapping can be determined using character frequency analysis, it is also totally insecure against a ciphertext-only attack.

3.5.2 Polyalphabetic Cipher

The idea behind a polyalphabetic cipher is to allow a unit of plaintext to be encrypted to one of many possible ciphertext units. A common example is the Vigenere cipher, which uses modulo addition with a key. For example if the message is *thisisamessage* and the key is $CRYPTO$, and the ciphertext is computer as follows:

m =	t	h	i	s	i	s	a	m	e	s	s	a	g	e
k =	C	R	Y	P	T	O	C	R	Y	P	T	O	C	R
c =	V	Y	G	H	B	G	C	D	C	H	L	O	I	V

The Vigenere cipher is totally insecure against a chosen-plaintext attack, since the key can easily be figured out through modulo arithmetic.

3.5.3 One-Time Pad

The one-time pad uses modulo addition with a key at least as long as the message itself. For example, if the message is *message* and the key is $SMFJDLG$, the ciphertext can be computer as follows:

m =	m	e	s	s	a	g	e
k =	S	M	F	J	D	L	G
c =	F	Q	X	C	D	R	K

The one-time pad is like the Vigenere cipher, but a different key is used to encrypt each message. This makes the one-time pad semantically secure against a ciphertext-only attack, even by an adversary with unlimited resources. However, the one-time pad is impractical due to:

1. Its long key length
2. Key reuse introducing a vulnerability (Since $c_1 = m_1 + k$ and $c_2 = m_2 + k$, $c_1 - c_2 = m_1 - m_2$, so if one message is known, another can be computed.

3.6 Stream Ciphers

As an alternative to using a random key, a pseudorandom bit generator (PRBG) can be used to generate a key. The PBRG is seeded with a secret value shared between Alice and Bob, and then generates new "random-looking" keys based on its previous state.

3.6.1 RC4 Cipher

RC4 is a simple, fast cipher for which no catastrophic weakness has been found. It consists of two components, namely a key scheduling algorithm and a keystream generator.

The high-level idea of the key scheduling algorithm is to generate a random-looking permutation of 0 to 255 (stored in S) based on the secret key (K).

```
for i from 0 to 255
  S[i] := i // initialize to identity permutation

j := 0
for i from 0 to 255
  j := j + S[i] + K[i mod keylength] mod 256
  swap(S[i], S[j])
```

The keystream generator accepts the permutation of S resulting from the key scheduling algorithm as input. The keystream generator continues to modify S while producing a bytestream.

```
i := 0
j := 0
while we need to generate another keystream byte
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap(S[i], S[j])
  t := S[i] + S[j] mod 256
  output S[t]
```

The keystream bytes are modulo added (i.e XORed) to the plaintext bytes to produce ciphertext bytes. There are known biases in the first few bytes of the keystream, so these bytes should be discarded.

RC4 was used for applications such as SSL and WEP. More recently, several weaknesses have been discovered, so RC4 is no longer widely used.

3.7 Block Ciphers

Block ciphers break up a plaintext into fixed-size blocks, and then encrypt the blocks one at a time. In contrast, stream ciphers encrypt the plaintext one character at a time (i.e. blocks are of length 1).

Some desirable properties of block ciphers are:

- **Diffusion:** Each ciphertext bit should depend on all plaintext bits.
- **Confusion:** The relationship between key bits and ciphertext bits should be complicated.
- **Algorithm simplicity**
- **Efficiency:** High speed of encryption and decryption.

3.7.1 Feistel Ciphers

Feistel ciphers are a class of block ciphers with the following parameters:

- n - half the block length
- h - number of encryption rounds
- l - key length
- $M = \{0,1\}^{2n}, C = \{0,1\}^{2n}, K = \{0,1\}^{l}$

A key scheduling algorithm determines h subkeys $k_1, k_2, ..., k_h$ from a key k. Each subkey k_i defines a component function $f_i : \{0,1\}^n \to \{0,1\}^n$. Feistel ciphers encrypt by iteratively applying these component functions as follows:

- Plaintext is $m = (m_0, m_1)$
- Round 1 of encryption results in (m_1, m_2), where $m_2 = m_0 \oplus f_1(m_1)$
- Round 2 of encryption results in (m_2, m_3), where $m_3 = m_1 \oplus f_2(m_2)$
- . . .
- Ciphertext is $c = (m_h, m_{h+1})$

Decryption applies these functions in reverse order.

New Data Seal (NDS) is a type of Feistel cipher with $n = 64$ and $h = 16$. k is a randomly selected function that converts one byte to another. Since there are 256 unique bytes, there are $256^{256} = 2^{2048}$ unique keys, so $l = 2048$. Subkeys k_i are simply k itself. The component function $f : \{0,1\}^{64} \to \{0,1\}^{64}$ is defined as follows:

1. Divide input z into 8 bytes $(z_1, z_2, ..., z_8)$.
2. Divide each byte z_j into two nibbles (n_1^j, n_2^j).
3. Apply some publicly known operations S_0 to n_1^j and S_1 to n_2^j to produce new nibbles p_1^j and p_2^j.
4. Let z^* be the byte obtained by taking the first bit of every byte. Compute $t = k(z^*)$.
5. If the jth bit of t is 1, then swap p_1^j and p_2^j.
6. Permute the resulting 64 bits by some publicly known operation P.

The key of NDS can be determined after a chosen-plaintext attack of around 32,000 carefully selected inputs. Thus, NDS is totally insecure, which underlines the importance of using different subkeys for different rounds of a Feistel cipher.

Data Encryption Standard (DES) is a type of Feistel cipher with $n = 32$, $h = 16$ and $l = 56$. Unlike NDS, DES selects different subkeys for each round of encryption; each subkey is a selection of 48 bits of k.

Although DES has been important in the development of modern cryptography, its short key length makes it susceptible to brute-force attacks. Another problem is the small block size ($2 \cdot 32 = 64$), which means that a collision between ciphertext blocks is expected approximately every $\sqrt{2^{64}} = 2^{32}$ blocks.

3.7.2 Multiple Encryption

Multiple encryption involves encrypting a message multiple times. For example, if DES encryption is done twice with independent keys, then $l = 2 * 56 = 112$, seemingly reducing the feasibility of exhaustive key search.

However, so-called meet-in-the-middle attacks can be used to greatly reduce the security level of the encryption scheme. For example, if $c = E_{k2}(E_{k1}(m))$, then instead of performing 2^{2l} operations to find the key (k_1, k_2), the result of $E_{k2}^{-1}(c)$ can be saved for all 2^l possible values of k_2. Then, $E_{k1}(m)$ can be run for all 2^l possible values of k_1, and then matches can be searched for such that $E_{k1}(m) = E_{k2}^{-1}(c)$. Thus, the security level of double encryption is roughly $2 \cdot 2^l = 2^{l+1}$, making it not much more secure than single encryption.

Triple encryption is more resistant to meet-in-the-middle attacks, requiring around $2^{2*56} = 2^{112}$ steps to break (i.e. "middle" here means two levels deep on one side and one level deep on the other). In light of its enhanced security, triple DES actually has some uses in production.

3.7.3 Modes of Operation

Modes of operation concern how to use a block cipher to encrypt some plaintext messages $m = m_1 m_2 ... m_t$. The simplest mode is called electronic codebook (ECB) mode, where plaintext blocks map to ciphertext blocks independently of one another. Since identical plaintexts result in identical ciphertexts, ECB mode is not semantically secure against chosen-plaintext attacks.

Cipher block chaining (CBC) mode involves each block of plaintext being XORed with the one encrypted before it. Since identical blocks of plaintext no longer map the the same ciphertext, CBC mode is secure against chosen-plaintext attacks.

3.8 Advanced Encryption Standard (AES)

AES is the most widely used production SKES. Established in 2001 by the US National Institute of Standards and Technology (NIST), AES is a block cipher using 128 bit blocks and supports key sizes of 128, 192, and 256 bits.