



Applied Cryptography

CO 487



Alfred Menezes

Preface

Disclaimer Much of the information on this set of notes is transcribed directly/indirectly from the lectures of CO 487 during Winter 2021 as well as other related resources. I do not make any warranties about the completeness, reliability and accuracy of this set of notes. Use at your own risk.

For any questions, send me an email via <https://notes.sibeliusp.com/contact/>.

You can find my notes for other courses on <https://notes.sibeliusp.com/>.

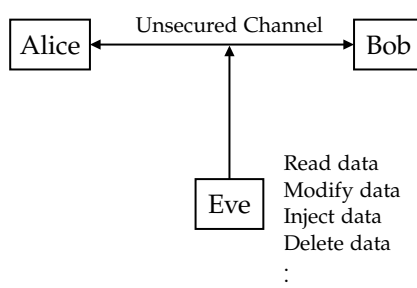
Sibelius Peng

Contents

Preface	1
1 Introduction	3
1.1 Secure Web Transactions	4
1.2 The TLS Protocol	4
1.3 Cryptography in Context	5
2 Symmetric-Key Cryptography	6
2.1 Basic concepts	6
2.1.1 The Simple Substitution Cipher	7
2.1.2 Polyalphabetic Ciphers	9
2.2 The One-Time Pad	10

Introduction

Cryptography is about securing communications in the presence of *malicious* adversaries.



Note that even if we call him “Eve”, he can do more than eavesdropping... The adversary is malicious, powerful and unpredictable.

Fundamental Goals of Cryptography

1. Confidentiality: Keeping data secret from all but those authorized to see it.
2. Data integrity: Ensuring data has not been altered by unauthorized means.
3. Data origin authentication: Corroborating the source of data.
4. Non-repudiation: Preventing an entity from denying previous commitments or actions.

Some examples of unsecured channel:

- Secure Browsing: The Internet
- Online Shopping: The Internet
- Automatic Software Upgrades: The Internet
- Cell Phone Service: Wireless
- Wi-Fi: Wireless
- Bluetooth: Wireless
- Messaging: Wired/Wireless

Communicating Parties

Alice and Bob are two communicating devices.

Alice	Bob	Communication channel
person	person	telephone cable
person	person	cellular network
person	web site	internet
iPhone	wireless	router wireless
iPhone	headphones	wireless
iPhone	service provider	cellular network
your car's brakes	another car	wireless
smart card	bank machine	financial network
smart meter	energy provider	wireless
military commander	satellite	space

1.1 Secure Web Transactions

Transport Layer Security (TLS): The cryptographic protocol used by web browsers for secure web transactions for secure access to amazon, gmail, hotmail, facebook etc.

TLS is used to assure an individual user (called a client) of the authenticity of the web site (called the server) he or she is visiting, and to establish a secure communications channel for the remainder of the session.

Symmetric-key cryptography: The client and server a priori share some secret information k , called a key.

They can subsequently engage in secure communications by encrypting their messages with AES and authenticating the resulting ciphertexts with HMAC.

How do they establish the shared secret key k ?

Public-key cryptography: Communicating parties a priori share some authenticated (but non-secret) information.

To establish a secret key, the client selects the secret session key k , and encrypts it with the server's RSA public key. Then only the server can decrypt the resulting ciphertext with its RSA private key to recover k .

How does the client obtain an authentic copy of the server's RSA public key?

Signature scheme: The server's RSA public key is signed by a Certifying Authority using the RSA signature scheme.

The client can verify the signature using the Certifying Authority's RSA public verification key which is embedded in its browser. In this way, the client obtains an authentic copy of the server's RSA public key.

1.2 The TLS Protocol

1. When a client first visits a secured web page, the server transmits its certificate to the client.
 - The certificate contains the server's identifying information (e.g., web site name and URL) and RSA public key, and the RSA signature of a certifying authority.
 - The certifying authority (e.g., Verisign) is trusted to carefully verify the server's identity before issuing the certificate.
2. Upon receipt of the certificate, the client verifies the signature using the certifying authority's public key, which is embedded in the browser. A successful verification confirms the authenticity of the server and of its RSA public key.
3. The client selects a random session key k , encrypts it with the server's RSA public key, and

transmits the resulting ciphertext to the server.

4. The server decrypts the ciphertext to obtain the session key, which is then used with symmetric-key schemes to encrypt (e.g. with AES) and authenticate (e.g. with HMAC) all sensitive data exchanged for the remainder of the session.
5. The establishment of a secure link is indicated by a closed padlock in the browser. Clicking on this icon reveals the server's certificate and information about the certifying authority.

TLS is one of the most successful security technologies ever deployed. But is TLS really secure?

There are many potential security vulnerabilities:

1. The crypto is weak (e.g., AES, HMAC, RSA).
2. Quantum attacks on the underlying cryptography.
3. Weak random number generation.
4. Issuance of fraudulent certificates
 - In 2001, Verisign erroneously issued two Class 3 code-signing certificates to a person masquerading as a Microsoft representative.
 - Mistake due to human error.
5. Software bugs (both inadvertent and malicious).
6. Phishing attacks.
7. TLS only protects data during transit. It does not protect your data when it is collected at the server.

Many servers store large amounts of credit card data and other personal information.

8. The National Security Agency (NSA)

1.3 Cryptography in Context

Cybersecurity is comprised of the concepts, technical measures, and administrative measures used to protect networks, computers, programs and data from deliberate or inadvertent unauthorized access, disclosure, manipulation, loss or use. Also known as information security. Cybersecurity includes the study of computer security, network security and software security.

Note that Cryptography \neq Cybersecurity.

- Cryptography provides some mathematical tools that can assist with the provision of cybersecurity services. It is a small, albeit an indispensable, part of a complete security solution.
- Security is a chain
 - Weak links become targets; one flaw is all it takes.
 - Cryptography is usually not the weakest link. However, when the crypto fails the damage can be catastrophic.

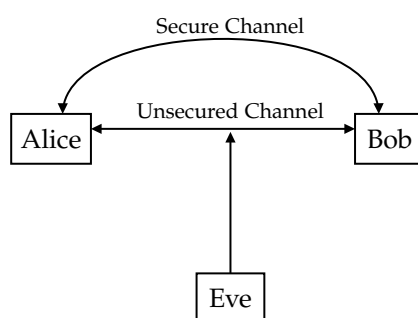
Symmetric-Key Cryptography

2.1 Basic concepts

symmetric-key encryption scheme

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: $E_k : M \rightarrow C, \forall k \in K$,
- a family of decryption functions: $D_k : C \rightarrow M, \forall k \in K$, such that $D_k(E_k(m)) = m$ for all $m \in M, k \in K$.



1. Alice and Bob agree on a secret key $k \in K$ by communicating over the secure channel.
2. Alice computes $c = E_k(m)$ and sends the ciphertext c to Bob over the unsecured channel.
3. Bob retrieves the plaintext by computing $m = D_k(c)$.

Some examples:

- WWII: Enigma Machine, Alan Turing,
https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma
- WWII: Lorenz Machine, Bill Tutte, Colossus
<http://tinyurl.com/COBillTutte>, <http://billtuttememorial.org.uk/>

2.1.1 The Simple Substitution Cipher

- M = all English msgs.
- C = all encrypted msgs.
- K = all permutations of the English alphabet.
- $E_k(m)$: Apply permutation k to m , one letter at a time.
- $D_k(c)$: Apply inverse permutations k^{-1} to c , one letter at a time.

Example:

$k =$

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>
<i>D</i>	<i>N</i>	<i>X</i>	<i>E</i>	<i>S</i>	<i>K</i>	<i>O</i>	<i>J</i>	<i>T</i>	<i>A</i>	<i>F</i>	<i>P</i>	<i>Y</i>	<i>I</i>	<i>Q</i>	<i>U</i>	<i>B</i>	<i>R</i>	<i>Z</i>	<i>G</i>	<i>V</i>	<i>C</i>	<i>H</i>	<i>M</i>	<i>W</i>	<i>L</i>

Encryption: $m = \text{the big dog}$, $c = E_k(\text{the big dog}) = \text{GJS NTO EQO}$.

Decryption: $c = \text{GJS NTO EQO}$, $m = E_k^{-1}(\text{GJS NTO EQO}) = \text{the big dog}$.

Is the simple substitution cipher a secure SKES? but wait, what does it mean for a SKES to be secure? We need a **security definition**.

1. What are the computational powers of the adversary?
2. How does the adversary interact with the two communicating parties?
3. What is the adversary's goal?

Security model: Defines the computational abilities of the adversary, and how she interacts with the communicating parties.

Basic assumption: The adversary knows everything about the SKES, except the particular key k chosen by Alice and Bob. (Avoid security by obscurity!!) Security should only depend on the secrecy of the secret keying material, not on the secrecy of the algorithms that describe the cryptographic scheme.

Let's now consider the computational power of the adversary.

- **Information-theoretic security:** Eve has infinite computational resources.
- **Complexity-theoretic security:** Eve is a 'polynomial-time Turing machine'.
- **Computational security:** Eve has 36,768 Intel E5-2683 V4 cores running at 2.1 GHz at her disposal. See: **Graham** in the basement of MC

We say: Eve is "*computationally bounded*".

In this course, we will be concerned with computational security.

Next consider how the adversary can interact with Alice and Bob.

- Passive attacks:
 - **Ciphertext-only attack:** The adversary knows some ciphertext (that was generated by Alice or Bob).
 - **Known-plaintext attack:** The adversary also knows some plaintext and the corresponding ciphertext. (stronger because more info than before)
- Active attacks:
 - **Chosen-plaintext attack:** The adversary can also choose some plaintext and obtains the corresponding ciphertext. This is at least as strong as known-plaintext attack because the attacker gets to choose the plain text for which it is given the corresponding ciphertext.
- Other attacks (not considered in this course):

- Clandestine attacks: bribery, blackmail, etc.
- Side-channel attacks: monitor the encryption and decryption equipment (timing attacks, power analysis attacks, electromagnetic-radiation analysis, etc.)

Finally, we will consider the adversary's goal (from strongest to weakest).

1. Recover the secret key.
2. Systematically recover plaintext from ciphertext (without necessarily learning the secret key).
3. Learn some partial information about the plaintext from the ciphertext (other than its length).

If the adversary can achieve 1 or 2, the SKES is said to be **totally insecure** (or **totally broken**).

If the adversary cannot learn any partial information about the plaintext from the ciphertext (except possibly its length), the SKES is said to be **semantically secure**. So the ciphertext hides all the meaning about the corresponding plaintext.

security of SKES

A symmetric-key encryption scheme is said to be **secure** if it is semantically secure against chosen-plaintext attack by a computationally bounded adversary.

Note that the definition captures the computational abilities of the adversary, namely some concrete upper bound on its computational resources. It captures how the adversary interacts with Alice and Bob, namely by mounting a chosen-plaintext attack. And finally, it captures the goal of the adversary, namely breaking semantic security. Note also that in the definition the attacker's capabilities are *as strong as possible*, namely a chosen-plaintext attack and its goal is *as weak as possible* namely, breaking semantic security.

From the definition, we can also deduce what it means to break an encryption scheme.

1. The adversary is given a challenge ciphertext c (generated by Alice or Bob using their secret key k).
2. During its computation, the adversary can select plaintexts and obtains (from Alice or Bob) the corresponding ciphertexts.
3. After a feasible amount of computation, the adversary obtains some information about the plaintext m corresponding to the challenge ciphertext c (other than the length of m).

If the attacker can win this game specified in these three steps, then we'll say that the encryption scheme is insecure.

Desirable Properties of a SKES

1. Efficient algorithms should be known for computing E_k and D_k (i.e., for encryption and decryption).
2. The secret key should be small (but large enough to render exhaustive key search infeasible).
3. The scheme should be secure.
4. The scheme should be secure even against the designer of the system.

The last point is to ensure that the designer hasn't somehow inserted a backdoor into the encryption scheme whereby the adversary can learn a secret keying material by looking at ciphertext, while this method would be hard to find by other people. This fourth property can sometimes be very difficult to obtain in practice.

Now we can see whether the simple substitution cipher is secure: Totally insecure against a chosen-

plaintext attack. This is because the adversary can simply give the plaintext message that's comprised of the English alphabet from a to z to Alice for encryption. Alice returns to the adversary the ciphertext, which it turns out is the secret key itself. And so by simply giving Alice a short plaintext and getting back the corresponding ciphertext the adversary has learnt Alice's secret key.

What about security under a ciphertext-only attack? Is exhaustive key search possible?

- Given sufficient amounts of ciphertext c , decrypt c using each possible key until c decrypts to a plaintext message which “makes sense”.
- In principle, 30 characters of ciphertext are sufficient on average to yield a unique plaintext that is a sensible English message. In practice, a few hundred characters are needed.

Exhaustive search:

- Number of keys to try is $26! \approx 2^{88}$
- If the adversary uses 10^6 computers, each capable of trying 10^9 keys per second, then exhaustive key search takes about 10^4 years.
- So, exhaustive key search is infeasible.

In this course,

- 2^{40} operations is considered very easy.
- 2^{56} operations is considered easy.
- 2^{64} operations is considered feasible.
- 2^{80} operations is considered barely feasible.
- 2^{128} operations is considered infeasible.

The Bitcoin network is presently performing hash operations at the rate of $2^{66.4}$ per second (or $2^{91.3}$ per year).

The Landauer limit from thermodynamics suggests that exhaustively trying 2^{128} symmetric keys would require $\gg 3000$ gigawatts of power for one year (which is $\gg 100\%$ of the world's energy production). http://en.wikipedia.org/wiki/Brute-force_attack

security level

A cryptographic scheme is said to have a **security level** of ℓ bits if the fastest known attack on the scheme takes approximately 2^ℓ operations.

As of the year 2021, a security level of 128 bits is desirable in practice.

Of course, simple frequency analysis of ciphertext letters can be used to recover the key. One would simply find the most frequently occurring ciphertext letter; this would most likely correspond to the English letter e which is the most frequently occurring letter in the English alphabet. Hence, the simple substitution cipher is totally insecure even against a ciphertext-only attack.

2.1.2 Polyalphabetic Ciphers

Basic idea: Use several permutations, so a plaintext letter is encrypted to one of several possible ciphertext letters.

Example: Vigenère cipher

Secret key is an English word having no repeated letters. E.g., $k = \text{CRYPTO}$.

Example of encryption:

$$\begin{array}{rcccccccccccccccc}
 m & = & t & h & i & s & i & s & a & m & e & s & s & a & g & e \\
 + & k & = & C & R & Y & P & T & O & C & R & Y & P & T & O & C & R \\
 \hline
 c & = & V & Y & G & H & B & G & C & D & C & H & L & O & I & V
 \end{array}$$

Here, $A = 0, B = 1, \dots, Z = 25$; addition of letters is mod 26.

Decryption is subtraction modulo 26: $m = c - k$.

Frequency distribution of ciphertext letters is flatter (than for a simple substitution cipher).

The Vigenère cipher is totally insecure against a chosen-plaintext attack. Not unexpectedly, the Vigenère cipher is also totally insecure against a ciphertext-only attack.

2.2 The One-Time Pad

Invented by Vernam in 1917 for the telegraph system. The key is a random string of letters. Example of encryption:

$$\begin{array}{rcccccccccccccccc}
 m & = & t & h & i & s & i & s & a & m & e & s & s & a & g & e \\
 + & k & = & Z & F & K & W & O & G & P & S & M & F & J & D & L & G \\
 \hline
 c & = & S & M & S & P & W & Y & P & F & Q & X & C & D & R & K
 \end{array}$$

Note: the key is as long as the plaintext.

The key should not be re-used:

- If $c_1 = m_1 + k$ and $c_2 = m_2 + k$, then $c_1 - c_2 = m_1 - m_2$.
- Thus $c_1 - c_2$ depends only on the plaintext (and not on the key) and hence can leak information about the plaintext.
- In particular, if m_1 is known, then m_2 can be easily computed.

Index

S

security level	9
security of SKES	8
symmetric-key encryption scheme	6