HAI709I : Fondements cryptographiques de la sécurité, Université de Montpellier, 2023

1 11/09/2023. Lecture 1.

In this lecture we discussed the simplest cryptographic protocol: encryption with a symmetric key. We assume that the Sender (*Alice*) wants to transmit a message to the Receiver (*Bob*) via a non-protected channel. The opponent (attacker, *Eve*) can intercept the data sent via the channel.

We assume that the clear message m is taken form the space of all possible messages $\mathcal{M}$. We assume that the message is random, i.e., there is a probability distribution on $\mathcal{M}$, and we choose from this set a random element m according to this distribution.

Alice and Bob fix in advance a general rules of communication, which are called an *encryption scheme*. The description of an encryption scheme consists of three spaces (finite sets) and three algorithms. The spaces are $\mathcal{M}$ (the space of clear messages, a.k.a. plaintexts, *messages clairs*), $\mathcal{E}$ (the space of encoded messages, *messages chiffrés*), and $\mathcal{K}$ (the space of secret keys, *clefs secretes*). The scheme involves three algorithms $\langle \texttt{Gen}, \texttt{Enc}, \texttt{Dec} \rangle$ that are used as follows:

- a randomized algorithm `Gen()` without arguments produces a random element k in a space $\mathcal{K}$ (this k is called a *secret key*); we assume that the produced distribution of probabilities on $\mathcal{K}$ is independent on the distribution of clear messages on $\mathcal{M}$;
- a deterministic (or, in some cases, randomzied) algorithm $\texttt{Enc}(m, k)$ with two arguments

$$\texttt{Enc} \ : \ \mathcal{M} \times \mathcal{K} \to \mathcal{E}$$

  (it takes a clear message m and a secret key k and produces the encrypted message);
- a deterministic algorithm $\texttt{Dec}(e, k)$ with two arguments computes a mapping

$$\texttt{Dec} \ : \ \mathcal{E} \times \mathcal{K} \to \mathcal{M}$$

  (it takes an encrypted message and a secret key and produces the clear message).

We require that for all m and for all k

$$\texttt{Dec}(\texttt{Enc}(m, k), k) = m \tag{1}$$

(i.e., encoding and then decoding with the same key gives the initial clear message).

In practice such a scheme is understood in a natural way: first of all, Alice and Bob fix a scheme $\langle \texttt{Gen}, \texttt{Enc}, \texttt{Dec} \rangle$; then, before the communication session begins, they use $\texttt{Gen}()$ to sample a common secret key k; when Alice wants to communicate a message m, she computes an encrypted message

$$e := \texttt{Enc}(m, k)$$

and transmits e to Bob via a public channel. Bob decodes the message by computing

$$m = \texttt{Dec}(e, k).$$

Condition (1) guarantees that Bob obtains the original clear message m.

The usual requirements called *Kerckhoffs's principle* is that a cryptographic scheme remain secure, even if everything about the scheme, except the key, is known in full detail to the opponent. The principle is the opposite of *security through obscurity*. In our setting, Kerckhoffs's principle means that the opponent may know in advance the spaces $\mathcal{M}, \mathcal{K}, \mathcal{E}$ and the full description of the algorithms $\texttt{Gen}, \texttt{Enc}, \texttt{Dec}$. On the other hand, the value of the secret key k should be known only to Alice and Bob.

What does it mean that a cryptographic scheme is *secure*? A scheme $\langle \texttt{Gen}, \texttt{Enc}, \texttt{Dec} \rangle$ is called *secure* if the encrypted message e provides to the opponent no information on the clear message m. Moreover, even if the opponent has some *a priori* information on m (e.g., a few first letters of the message, any specific subword, frequencies of letters used in the messages, and so on), e should not give to the opponent any *supplementary* information on m. Let us give a more formal definition.

Definition 1. A scheme $\langle \texttt{Gen}, \texttt{Enc}, \texttt{Dec} \rangle$ is called *secure* if for every probability distribution on $\mathcal{M}$, for all $m \in \mathcal{M}$ and for all $e \in \mathcal{E}$

$$\begin{gathered} \mathrm{Prob}[\text{clear message is equal to } m] \\ \| \\ \mathrm{Prob}[\text{clear message is equal to } m \mid \text{encrypted message is equal to } e]. \end{gathered}$$

That is, the knowledge of the opponent on m (the distribution of probabilities on all possible messages) does not change when the opponent gets the encrypted message e. So if the opponent can come to some conclusions given e, exactly the same conclusions can be done without it.

In what follows we discuss one specific example of a secure encryption scheme.

Vernam's system, a.k.a. *one-time pad*.
We will assume that $\mathcal{M} = \{0,1\}^n$. We let $\mathcal{K} = \mathcal{E} = \{0,1\}^n$ and define a cryptographic scheme as follows:

- $\texttt{Gen()}$ samples a random element of $\mathcal{K}$ with the uniform distribution (every $k \in \mathcal{K}$ is produced with probability $1/2^n$).
- $\texttt{Enc} : \mathcal{M} \times \mathcal{K} \to \mathcal{E}$ is the bitwise XOR between the clear message m and the secret key k; for example, $\underbrace{00110101}_{m} \oplus \underbrace{11001101}_{k} = \underbrace{11111000}_{e}$.
- $\texttt{Enc} : \mathcal{E} \times \mathcal{K} \to \mathcal{M}$ is the bitwise XOR between the encrypted message and the secret key; for example, $\underbrace{11111000}_{e} \oplus \underbrace{11001101}_{k} = \underbrace{00110101}_{m}$.

It is easy to verify that this scheme satisfies (1) since $(m \oplus k) \oplus k = m$ for all strings m and k.

Theorem 1. *Vernam's system is secure in the sense of Definition 1.*

Proof. (sketch) First of all, we observe that

$$\begin{aligned} &\mathrm{Prob}[\text{clear message is equal to } m \text{ and the encrypted is equal to } e] \\ &= \mathrm{Prob}[\text{clear message is equal to } m \text{ and secret key is } k = e \oplus m] \\ &= \mathrm{Prob}[\text{clear message is equal to } m] \cdot \mathrm{Prob}[\text{secret key } k = e \oplus m] \\ &= \mathrm{Prob}[\text{clear message is equal to } m] \cdot \tfrac{1}{2^n}. \end{aligned}$$

On the other hand, for every $e \in \mathcal{E}$

$$\begin{aligned}
&\text{Prob[encrypted message is equal to } e] \\
&= \sum_{k,m \text{ such that } k \oplus m = e} \text{Prob[clear message is equal to } m \text{ and the key is equal to } k] \\
&= \sum_{k,m \text{ such that } k \oplus m = e} \text{Prob[clear message is equal to } m] \cdot \text{Prob[and the key is equal to } k] \\
&= \sum_{m \in \mathcal{M}} \text{Prob[clear message is equal to } m] \cdot \tfrac{1}{2^n} \\
&= \tfrac{1}{2^n} \cdot \sum_{m \in \mathcal{M}} \text{Prob[clear message is equal to } m] = \tfrac{1}{2^n}.
\end{aligned}$$

By the definition of conditional probability we obtain

$$\begin{aligned}
&\text{Prob[clear message is equal to } m \mid \text{encrypted message is equal to } e] \\
&= \frac{\text{Prob[clear message is equal to } m \text{ and encrypted message is equal to } e]}{\text{Prob[encrypted message is equal to } e]} \\
&= \frac{\text{Prob[clear message is equal to } m] \cdot (1/2^n)}{1/2^n} = \text{Prob[clear message is equal to } m].
\end{aligned}$$

The equality between the red and the blue terms is exactly the definition of security (Definition 1 above). □

In Vernam's scheme the size of the secret key (n bits, which corresponds to 2^n possible values) is exactly equal to the size of the clear message. This relation between the size of the key and the size of the message is optimal. More precisely, we have the following theorem.

Theorem 2. *For every scheme* $\langle \texttt{Gen}, \texttt{Enc}, \texttt{Dec} \rangle$ *that is secure in the sense of Definition 1*

$$|\mathcal{K}| \geq |\mathcal{M}|,$$

i.e., the size of the secret key cannot be smaller than the size of the clear message.

We skip the proof (discussed in the class).

In the class we discussed why using the same secret key several times is a bad idea (this implies a loss of security in terms of our theoretical definition of a secure scheme, and it opens a door for various attacks in practice).

Example 1. In this example we let $\mathcal{M} = \mathcal{E} = \{A, B, C, \ldots, Z\}$ (the whole plaintext is a single letter of the latin alphabet), and $\mathcal{K} = \{1, 2, \ldots, 26\}$. We define a cryptographic scheme as follows:

- `Gen()` samples a random element of $\mathcal{K}$ with the uniform distribution (every integer $k = 1, \ldots, 26$ is produced with probability $1/26$).
- The encoding $\texttt{Enc}(m, k)$ maps each letter m of the latin alphabet to the letter which stands in the alphabet k positions later (modulo 26) than m. For example, $\texttt{Enc}(A, 3) = D$, $\texttt{Enc}(F, 5) = K$, $\texttt{Enc}(Y, 4) = C$.
- The decoding $\texttt{Dec}(e, k)$ maps each letter e of the latin alphabet to the letter which appears in the alphabet k positions earlier (again, modulo 26) than e. For example, $\texttt{Dec}(A, 3) = X$, $\texttt{Dec}(F, 5) = A$, $\texttt{Dec}(Y, 4) = U$.

Exercise 1. Prove that the scheme form Example 1 satisfies the definition of a secure encryption scheme.

Example 2. In this example we let $\mathcal{M} = \mathcal{E} = \{A, B, C, \ldots, Z\}^n$ (an n-letter text in the latin alphabet), and $\mathcal{K} = \{1, 2, \ldots, 26\}$. The encoding are defined similarly to Example 1: we apply *the same* shift in the alphabet by k positions to each letter in the clear text; the decoding procedure is the shift in the alphabet by k positions in the opposite direction (applied to each letter in the encoded text). This scheme is *not* secure when $n > 1$.

Example 3. The scheme from Example 2 becomes secure if we let $\mathcal{K} = \{1, 2, \ldots, 26\}^n$ and use individual (chosen independently) shifts k_j for each position j in the clear text $m \in \{A, B, C, \ldots, Z\}^n$.

Example 4 [substitution scheme]. Let $\mathcal{M} = \mathcal{E} = \{A, B, C, \ldots, Z\}^n$ (an n-letter text in the latin alphabet), and let $\mathcal{K}$ consist of all permutations

$$\pi \ : \ \{A, B, C, \ldots, Z\} \to \{A, B, C, \ldots, Z\}.$$

The algorithm `Gen()` chooses such a permutation at random. The encoding procedure applies the chosen permutation π to each letter of he clear message (one and the same permutation is applied to each letter of the message). The decoding procedure applies the inverse permutation π^{-1} to each letter of he encoded message (again, one and the same permutation is applied to position in the message). Historically, this scheme was very popular. However, it is not secure (for large enough n).

Exercise 2 (optional)**.** The *substitution scheme* (with some unknown permutation π) was used to encode a french text (with omitted spaces and punctuation marks). The encoded text is

```
hzkvrynrkzuztzvntagxxeizlzzygynzkivicscelztzvntagxxeiixryucoroe
rnhzuiyecyivlrnhzzyuzcokivlrzehinvbklgsvikarzmrhrlirvzucqgcvyih
uzeenrzynzemrhrlirvzenzkvrynrkzikkivirlkivmrhzeerouzeruzviliuzh
invbklgsvikarzmrhrlirvzzygynzekivtzvntagxxeuiyeegylvirlzfcregyl
hzebelzmzugrlzlvzmilzvrzhhzmzylerygymilazmilrfczmzylryuznarxxvi
whzrrhxiclfcrhyzorszkiehzeznvzlzlfcrhkcreezeiyeryngyjzyrzyllgmw
zvzylvzhzemiryeuzhzyyzmrhinhzxugrlkgcjgrvzyzlvzngmmcyrfczzzlvzl
zyczeiyehzezngcveuzyglzeznvrlzezlzlvznaiyszzgcmgurxrzzicsvzuzen
gvvzekgyuiylerhxiclfcrhegrlikkhrniwhzihingvvzekgyuiynzlzhzsvika
rfczrhxiclfcrhegrlkgvlilrxzlfczegymiyrzmzylgcegyxgynlrgyyzmzyly
zorszkiehzngyngcveuzkhcerzcvekzvegyyzezyxryrhzelyznzeeirvzjchze
nrvngyeliynzefcrzyngmmiyuzylhikkhrnilrgyfczhzebelzmzegrlucyceis
zxinrhzyzuzmiyuiylyrlzyergyuzekvrlyrhingyyireeiynzucyzhgysczezv
rzuzvzshzeigwezvjzvtzvntagxxeryerelzecvhzelvgrekvzmrzveuzeruzvi
lifcregyljzvrliwhzmzylgvrsryicoiegyzkgfczhzelvgreuzvyrzveyzliyl
ihgvekiengylzelzenzfcrzelikkzhzicqgcvuacrkvrynrkzuztzvntagxxeze
lzeezylrzhhzmzylhzuzcorzmz
```

Reconstruct the clear text.

References

[1] J. Katz, Y. Lindell. Introduction to modern cryptography, CRC Press, 2021 [sections 1.2, 2.1–2.3].