

Les conversions classiques entre types de données

1 Les données

1.1 Présentation du problème

En cryptographie moderne, la plupart des primitives de chiffrement s'appliquent à des nombres entiers (chiffrements arithmétiques) ou à des blocs de bits (opérations booléennes). En revanche les données qu'on chiffre sont des fichiers qui en général ont des structures liées à des représentations sous forme d'octets (textes par exemple). Il est donc nécessaire pour s'affranchir de ces différences de prévoir des procédures standards de traduction permettant de passer d'une représentation à une autre.

Nous définissons dans la suite les fonctions suivantes, conformes au standard ISO/IEC 18033-2.

$OS2BSP(u)$	Octet String to Bit String Procedure
$BS2OSP(b)$	Bit String to Octet String Procedure
$BS2IP(b)$	Bit String to Integer Procedure
$I2BSP(x, l)$	Integer to Bit String Procedure
$OS2IP(u)$	Octet String to Integer Procedure
$I2OSP(x, l)$	Integer to Octet String Procedure

1.2 Les bits

Notons \mathbb{F}_2 l'alphabet à deux éléments $\{0, 1\}$. Chacun de ces deux éléments est appelé **bit**. On parle ainsi du bit 0 ou du bit 1. Quand cela sera nécessaire, on utilisera certaines opérations classiques sur \mathbb{F}_2 (par exemple le "xor" (addition modulo 2), le "or" (borne supérieure), le "and" (multiplication ou aussi borne inférieure)). Un **mot** de longueur l construit sur cet alphabet est une **suite finie de l bits**. En anglais on appelle une telle suite finie un *bit string*.

1.3 les octets

Un **octet** est une suite finie de 8 bits. On notera \mathbb{F}_{2^8} l'alphabet constitué des 256 octets possibles. Quand c'est utile on peut définir des opérations sur \mathbb{F}_{2^8} . En particulier le "xor" (ou exclusif bit à bit). Un **mot** de longueur l construit sur cet alphabet est une **suite finie d'octets** (*octet string* en anglais).

1.4 Les entiers

Les entiers qui servent en cryptographie sont en général des grands nombres, et sont traités en informatique grâce à des bibliothèques spécifiques (exemples : BigInteger en Java, zz en NTL etc.).

2 Les diverses conversions

2.1 Suite d'octets \leftrightarrow suite de bits

- La fonction *OS2BSP* (Octet String to Bit String Procedure) transforme une suite d'octets en suite de bits. Cette transformation est triviale, une suite $(u_i)_{0 \leq i \leq l-1}$ de l octets est transformée en une suite $(b_k)_{0 \leq k \leq 8l-1}$ de $8l$ bits en mettant les l octets à la suite les uns des autres.

$$b_k := u_{\lfloor k/8 \rfloor} [k \bmod 8].$$

- La fonction *BS2OSP* (Bit String to Octet String Procedure) transforme une suite finie b de bits de longueur $8l$ en l'unique suite finie u d'octets de longueur l telle que $OS2BSP(u) = b$. Si la suite initiale de bits n'est pas de longueur multiple de 8, il faut s'y ramener avant d'utiliser la fonction *BS2OSP*. Il y a diverses façons de faire, notamment rajouter des bits 0 jusqu'à tomber sur un multiple de 8.

2.2 Suite de bits \leftrightarrow entiers

- La fonction *BS2IP* (Bit String to Integer Procedure) transforme une suite de bits $b = b_0 b_1 \dots b_{l-1}$ en un entier x suivant la formule :

$$BS2IP(b) = \sum_{i=0}^{l-1} b_{l-i-1} 2^i.$$

- La fonction *I2BSP* (Integer to Bit String Procedure) prend en entrée 2 entiers $x, l \geq 0$ et renvoie l'unique suite b de bits, de longueur l telle que $BS2IP(b) = x$, si cette suite existe (c'est-à-dire si on n'a pas pris l trop petit). Si la suite n'existe pas la fonction renvoie un symbole d'erreur. On notera $Oct(x) = I2BSP(x, 8)$. Si $x > 255$ cette fonction renvoie une erreur.

2.3 Suite d'octets \leftrightarrow entiers

- La fonction *OS2IP* (Octet String to Integer Procedure) transforme une suite d'octets $u = u_0 u_1 \dots u_{l-1}$ en un entier x suivant la formule :

$$OS2IP(u) = \sum_{i=0}^{l-1} u_{l-i-1} 256^i.$$

Il est équivalent de dire :

$$OS2IP(u) = BS2IP(OS2BS(u)).$$

- La fonction *I2OSP* (Integer to Octet String Procedure) prend en entrée 2 entiers $x, l \geq 0$ et renvoie l'unique suite u d'octets, de longueur l telle que $OS2IP(u) = x$, si cette suite existe (c'est-à-dire si on n'a pas pris l trop petit). Si la suite n'existe pas la fonction renvoie un symbole d'erreur.

*Auteur : Ainigmatias Cruptos
Diffusé par l'Association ACrypTA*