WEP

1 WEP

— Écrire une procédure `CRC32:=proc(M)` qui prend comme entrée un message `M` (sous la forme d'une liste de bits) et qui retourne l'encodage de `M` $\left(\text{ i.e. } X^{32}\cdot \texttt{M}(X)\right)$ et le CRC32 de `M`.
— Écrire une procédure `RC4KSA:=proc(K)` qui prend comme entrée une clef `K` et qui retourne le registre RC4 après la phase d'initialisation (RC4-KSA).
— Écrire une procédure `RC4PRGA:=proc(R,t)` qui prend en entrées un registre `R` et un entier `t` et qui retourne une suite chiffrante RC4 de longueur `t` (RC4-PRGA).
— Écrire une procédure `RC4:=proc(M, K)` qui prend comme entrées un message `M` (sous la forme d'une liste d'octets) et une clef `K` et qui retourne un chiffrement RC4 de `M`.
— Écrire une procédure `RandomIV:=proc()` qui retourne une liste de 24 bits aléatoires.
— Écrire une procédure `Trame:=proc(M, K)` qui prend comme entrées un message `M` (sous la forme d'une liste de bits) et une clef `K` de 40 bits (aussi sous forme d'une liste) et qui retourne une *trame*, i.e. :
  — un IV, que l'on note `IV`, associé à la trame.
  — un CRC32 de `M` que l'on note `CRC`.
  — un chiffrement RC4 – en mode WEP – de $\texttt{M}'\,||\texttt{CRC}$, avec `M'` l'encodage de `M`.
— Écrire une procédure `Decrypt:=proc(K,T)` qui prend comme entrées la clef secrète `K`, et qui déchiffre la trame `T`. La fonction retourne un message d'erreur si la trame n'est pas valide.
— Écrire une procédure `Inject:=proc(M',M,T)` qui prend comme entrées un message `M'` (sous la forme d'une liste de bits), un autre message `M`, la trame `T` (i.e. les donnés que retournent la fonction `Trame`) correspondant à `M` et qui renvoie une trame valide pour $\texttt{M}\oplus\texttt{M}'$.
— Écrire les fonctions qui permettent de monter l'attaque ChopChop