UNIVERSITÀ DEGLI STUDI "ROMA TRE"
CORSO DI STUDI IN MATEMATICA
IN450 - INFORMATICA 6
ALGORITMI PER LA CRITTOGRAFIA – A.A. 2016-2017

M. PEDICINI

ESONERO DEL 12/01/2017 – TEMPO 3H00

COGNOME _____ NOME _____ MATRICOLA _____

Esercizio 1. *(20pt) (KeyExpansion/AES-SubBytes)*

Sia $p(x) = x^4 + x + 1$ si consideri la S-box di AES/Rijndael adattata opportunamente a 4 bit e che incorpora le operazioni sul campo finito $\mathbb{F}_{2^4} = \mathbb{F}_2[x]/p(x)$.

Eseguire un round di cifratura di AES del blocco $X =$ `0x0220023332111313` *ponendo la chiave $k =$* `0x0DEADC0DEBADC0DE` *(*`0x` *è la notazione per indicare il sistema di numerazione esadecimale):*

(1) Adattare opportunamente l'algoritmo AES alle dimensioni del campo discutendo la scelta effettuata (in particolare nel caso della SubBytes*).*

Nel seguito le funzioni SubBytes*,* ShiftRows*,* MixColumns *e* AddKey *sono quelle dell'algoritmo adattato:*

a. Calcolare $S_1 =$ SubBytes(X);
b. Calcolare $S_2 =$ ShiftRows(S_1);
c. Calcolare $S_3 =$ MixColumns(S_2);
d. Calcolare $S_4 =$ AddKey(S_3, key_0) dove key_0 è il primo elemento del KeySchedule *calcolato a partire dalla chiave k.*

Esercizio 2. *(7pt) (Differential Cryptanalysis)*

Calcolare la tabella delle caratteristiche differenziali per la S-BOX $f : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ definita da:

$$f(x_1, x_2, x_3, x_4) = (x_1\, x_3, x_3x_4 + 1, x_3 + x_2 + 1, x_1 + x_2 + x_3).$$

Esercizio 3. *(15pt) (Hash Functions) Avendo presente l'Algoritmo di Merkle-Damgaard (vedi pagina seguente) che definisce una funzione di hash iterata a partire da una funzione di compressione $C : \{0,1\}^{m+t} \to \{0,1\}^m$ con $t \geq 2$.*

(1)
- *Mostrare che per ogni n, il parametro d è non negativo e che $d < t$. Mostrare quindi che la rappresentazione binaria di d potrà sempre essere rappresentata in $t - 1$ bits.*
- *Al fine di definire al punto 2, un hash function simile a quella di Merkle-Damgaard, si trovi uno schema alternativo di codifica per d in modo che almeno un bit di y_k sia un uno.*

(2) posto $t < 2^m$ sia $w =$ "la rappresentazione binaria di d in m bits" : descrivere una variante dell'algoritmo di Merkle-Damgaard e dimostrare che se poniamo

$$z_1 := w||0||y_1$$

e al tempo stesso riduciamo il numero di iterazioni al processamento dell'ultimo blocco (ovvero y_k) allora la sicurezza dell'algoritmo non cambia.

Fare attenzione che gli ultimi d bit meno significativi di y_k e che gli m bit più significativi di z_1 potrebbero essere uguali ad una qualsiasi sequenza binaria.

Per comodità è riportato l'algoritmo 4.6 del libro da modificare:

1: ***function*** MERKLE-DAMGARD(x)
2: $n := |x|$
3: $k := \lceil n/(t-1) \rceil$
4: $d := k(t-1) - n$
5: ***for*** i *in* $1 \ldots k-1$ ***do***
6: $y_i := x_i$
7: ***end for***
8: $y_k := x_k || 0^d$
9: $y_{k+1} :=$ *rappresentazione binaria di* d
10: $z_1 := 0^{m+1} || y_1$
11: $g_1 := C(z_1)$
12: ***for*** i *in* $1 \ldots k$ ***do***
13: $z_{i+1} := g_i || 1 || y_{i+1}$
14: $g_{i+1} := C(z_{i+1})$
15: ***end for***
16: $h(x) := g_{k+1}$
17: *return* $(h(x))$
18: ***end function***
