Corso di Sicurezza e Crittografia
Anno Accademico 2010-2011
Homework I — Soluzioni

Esercizio 1.
Nella codifica ASCII, ogni carattere alfanumerico corrisponde ad una sequenza di 8 bit. Il cifrario a rotazione polialfabetica può essere definito come la tripla $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ dove:

- Gen su input 1^n sceglie casualmente t tale che t sia più piccolo di $p(n)$ dove p è polinomio fissato, e restituisce un elemento casuale di $\{0,1\}^{8t}$, che indichiamo con $k = k_1 \dots k_t$ (dove $|k_i| = 8$);
- $\mathsf{Enc}(k, m)$ è definito solo se $|m|$ è un multiplo di 8, cioè se $m = m_1 \dots m_s$ dove $|m_i| = 8$. In tal caso, $\mathsf{Enc}(k, m) = c_1 \dots c_s$, dove $c_i = m_i \circ k_{(i-1 \ mod \ t)+1}$, dove $\circ$ è un'operazione di rotazione verso l'alto che lavora a gruppi di 8 bit;
- Dec è definito in modo analogo a Enc, ma l'operazione di rotazione $\circ$ è eseguita verso il basso anziché verso l'alto.

Π è un cifrario insicuro secondo tutte le definizioni di sicurezza che abbiamo studiato. Dimostriamo ad esempio che Π non può avere codifiche indistinguibili. Sia $\mathcal{A}$ un avversario che, su input 1^n, restituisce la coppia di messaggi (m_0, m_1) dove $m_0 = 0^{8p(n)} \cdot 0^8$ e $m_1 = 0^{8p(n)} \cdot 1^8$. Ricevuto il crittogramma c, $\mathcal{A}$ procede prima di tutto alla determinazione del periodo t, e può farlo analizzando i primi $8p(n)$ bit di c, visto che questa parte del crittogramma è stata comunque ottenuta a partire da $0^{8p(n)}$, indipendentemente da quale tra m_0 e m_1 sia stato scelto dall'esperimento. A questo punto, l'avversario può analizzare gli ultimi 8 bit di c. Se tali bit corrispondono a quelli in posizione $8(p(n)-t)+1, \dots, 8(p(n)-t)+8$, allora l'avversario restituisce 0, altrimenti restituisce 1. È facile rendersi conto che quest'avversario ha successo con probabilità 1.

Esercizio 2.

- L'algoritmo D_1 non è un generatore pseudocasuale, perché la dimensione dell'output prodotto è identica alla dimensione dell'input.
- L'algoritmo D_2 non è un generatore pseudocasuale, perché un distinguitore $\mathcal{D}$ può essere facilmente costruito nel modo seguente: $\mathcal{D}$ su input $b_1 \dots b_{n+1}$ controlla se $b_{n+1} = \oplus_{i=1}^{n} b_i$ e restituisce il risultato di tale test espresso in forma binaria. È evidente che:

$$Pr(\mathcal{D}(s) = 1) = \frac{1}{2}$$

  se s è scelta casualmente tra tutte le sequenze lunghe $n+1$, mentre

$$Pr(\mathcal{D}(D_2(s)) = 1) = 1$$

  se s è scelta casualmente tra tutte le sequenze lunghe n.
- Anche D_3 non può essere generatore pseudocasuale, in quanto un distinguitore $\mathcal{D}$ per D_3 è facilmente costruibile nel modo seguente: su input $b_1 \dots b_{2n}$, $\mathcal{D}$ controlla se $b_{2i} = b_{2i-1}$ per ogni $i \in \{1, \dots, n\}$. In caso positivo, restituisce 1, altrimenti restituisce 0. Chiaramente:

$$Pr(\mathcal{D}(s) = 1) = \frac{1}{2^n}$$

  se s è scelta casualmente tra tutte le sequenze lunghe $2n$, mentre

$$Pr(\mathcal{D}(D_3(s)) = 1) = 1$$

  se s è scelta casualmente tra tutte le sequenze lunghe n.

Esercizio 3.
Dimostriamo che se Π ha codifiche indistinguibili, allora Π è sicuro rispetto a P ogni predicato P decidibile in tempo polinomiale (deterministico). Facciamo la supposizione, non presente nel testo

dell'esercizio ma del tutto ragionevole, che sia $Pr(P(x))$ che $Pr(\neg P(x))$ siano funzioni limitate inferiormente da una funzione nella forma $\frac{1}{p(n)}$ ($p(n)$ polinomio), dove x varia casualmente su tutte le stringhe binarie in $\mathcal{M}_n$.
Dato un avversario $\mathcal{A}$ che falsifichi il fatto che Π è sicuro rispetto ad un tale predicato P, construiamo un avversario $\mathcal{A}'$ per Π nel senso delle codifiche indistinguibili. $\mathcal{A}'$ procede quindi nel modo seguente:

- Riceve in input 1^n.
- Genera casualmente $n \cdot p(n)$ stringhe casuali $s_1, \ldots, s_{n \cdot p(n)}$ tra tutte quelle in $\mathcal{M}_n$. Controlla per ogni s_i generato in questo modo se vale $P(s_i)$. Se esiste un j tale che $\neg P(s_j)$, allora $m_0 := s_j$, altrimenti $m_0 := s_1$.
- Genera casualmente $n \cdot p(n)$ stringhe casuali $r_1, \ldots, r_{n \cdot p(n)}$ tra tutte quelle in $\mathcal{M}_n$. Controlla per ogni r_i generato in questo modo se vale $\neg P(r_i)$. Se esiste un j tale che $P(r_j)$, allora $m_1 := r_j$, altrimenti $m_1 := r_1$.
- Restituisci all'ambiente (m_0, m_1), il quale restituirà un crittogramma c.
- Passa c ad $\mathcal{A}$, il quale restituisce un bit b.
- Restituisci all'ambiente b.

Chiamiamo R_0 (rispettivamente, R_1) l'evento probabilistici che si verifica quando il messaggio m_0 (risepttivamente, m_1) generato al passo 2 (rispettiavamente, al passo 3) è tale che $P(m_0)$ (rispettivamente, $\neg P(m_1)$). Sia poi R la congiunzione di R_0 e R_1 Osserviamo che

$$\begin{aligned} Pr(R) = 1 - Pr(\neg R) &\geq 1 - Pr(\neg R_0) - Pr(\neg R_1) \\ &\geq 1 - (1 - \frac{1}{p(n)})^{n \cdot p(n)} - (1 - \frac{1}{p(n)})^{n \cdot p(n)} \\ &\geq 1 - (e^{-1})^n - (e^{-1})^n = 1 - \varepsilon(n) \end{aligned}$$

dove ε è una funzione trascurabile. Possiamo quindi dire che:

$$\begin{aligned} Pr(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A}',\Pi}(n) = 1) &\geq Pr(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A}',\Pi}(n) = 1 \wedge R) \\ &= Pr(\mathsf{PrivK}^{\mathsf{eav}}_{P,\mathcal{A},\Pi}(n) = 1 | R) \cdot Pr(R) \\ &\geq (\frac{1}{2} + \varphi(n))(1 - \varepsilon(n)) \end{aligned}$$

dove $\varphi(n)$ non è trascurabile, mentre $\varepsilon(n)$ è trascurabile. È facile verificare che tale quantità è esprimibile come

$$\frac{1}{2} - \frac{1}{2}\varepsilon(n) + \varphi(n) - \varphi(n)\varepsilon(n)$$

e, chiaramente, $\varphi(n) - \frac{1}{2}\varepsilon(n) - \varphi(n)\varepsilon(n)$ non può essere trascurabile. Tutto questo, però, è in contrasto con l'ipotesi che Π avesse codifiche indistinguibili.