به نام خدا

دانشکده‌ی علوم ریاضی

مقدمه‌ای بر رمزنگاری	دانشجو: علیرضا توفیقی محمدی
تمرین : سری ۴	
مدرّس: دکتر شهرام خزائی	شماره‌ی دانشجویی: ۹۶۱۰۰۳۶۳

مسأله‌ی ۱

آ)

collision resistant است، چراکه اگر m_1, m_2 یافت شود که $H_1(m_1) = H_1(m_2)$ نتیجه می‌گیریم:

$$H(m_1||0) = H(m_2||0)$$

پس H نیز collision resistant نیست.

ب)

collision resistant است، چراکه اگر m_1, m_2 یافت شود که $H_1(m_1) = H_1(m_2)$ نتیجه می‌گیریم:

$$H(m_1||m_1) = H(m_2||m_2)$$

پس H نیز collision resistant نیست.

ج)

collision resistant نیست، برای اثبات فرض کنید H یک تابع هش collision resistant باشد، حال تابع H' با تعریف زیر را در نظر بگیرید:

$$H'(m) = 0^{32}||H(m)$$

این تابع نیز collision resistant است، چرا که اگر m_1, m_2 ای یافت شود که $H'(m_1) = H'(m_2)$ نتیجه می‌گیریم $H(m_1) = H(m_2)$. اما اگر H_3 را با تابع H' تعریف کنیم، همیشه 0^{32} را می‌دهد پس collision resistantنیست.

د)

collision resistant است، چراکه اگر m_1, m_2 یافت شود که $H_1(m_1) = H_1(m_2)$ نتیجه می‌گیریم:

$$H(m_1)||H(0) = H(m_2)||H(0)$$

پس $H(m_1) = H(m_2)$ و در نتیجه H نیز collision resistant نیست.

ه)

collision resistant نیست چراکه $H_5(0^n) = H_5(1^n)$.

و)

collision resistant نیست چراکه $H_6(0^n) = H_6(1^n)$.

مسأله‌ی ۲

اگر m سمپل برداریم، $\binom{m}{3} = O(m^3)$ تا سه‌تایی داریم که هر کدام به احتمال 2^{-2n} یک 3-collision اند، پس برای اینکه یک 3-collision پیدا کنیم لازم است m ما آنقدر زیاد باشد که امیدریاضی 3-collisionها حداقل یک شود، پس خواهیم داشت:

$$cm^3 \times 2^{-2n} \geq 1 \implies m = O(2^{\frac{2n}{3}})$$

پس به $O(2^{\frac{2n}{3}})$ تا سمپل نیاز داریم.

مسأله‌ی ۳

حمله کننده A را در نظر بگیرید که ابتدا 0^{2n} را پرسیده و $\langle a, b\rangle$ را دریافت می‌کند، سپس 1^{2n} را می‌پرسد و $\langle c, d\rangle$ را دریافت می‌کند، طبق الگوریتم مکمان می‌دانیم که $F_k(0^n) = a, F_k(F_k(1^n)) = d$ پس $\mathrm{Mac}_k(0^n 1^n) = \langle a, d\rangle$ و در نتیجه حمله کننده $0^n 1^n$ و $\langle a, d\rangle$ را خروجی می‌دهد و با مزیت یک مک را جعل می‌کند.

مسأله‌ی ۴

چون کلید AES ۱۲۸ بیت است، پس نیمه‌ی اول IV است و نیمه‌ی دوم برابر با $\mathsf{Enc}_k(\mathrm{IV} \oplus m) = c$ است، چون می‌خواهیم متن رمزی را جوری تغییر دهیم که m' شود، کافی است IV را تغییر دهیم که m' را بسازیم، برای

اینکار $m \oplus m'$ را حساب کرده و با IV ایکس‌اور می‌کنیم، در نتیجه اگر متن جدید را دیکریپت کند خواهد داشت:

$$\mathsf{Dec}_k(c) \oplus \mathrm{IV}' = \mathsf{Dec}_k(c) \oplus (\mathrm{IV} \oplus (m \oplus m')) = \mathrm{IV} \oplus m \oplus (\mathrm{IV} \oplus (m \oplus m'))$$

همچنین ایکس‌اور پد شده‌ی دو پیام برابر با 00000000000000004000000000000000 است و در نتیجه پیام جعل شده برابر با 20814804c1767293bd9f1d9cab3bc3e7 ac1e37bfb15599e5f40eef805488281d است.

مسأله‌ی ۵

اگر $t(n) = O(\log n)$ باشد، تعداد کل مک‌های مختلفی که قابل تولید اند برابر با $2^{t(n)} = O(n)$ خواهد بود، در نتیجه، اگر یک رشته‌ی تصادفی را انتخاب کرده و یک مک تصادفی برای آن خروجی دهیم، به احتمال حداقل $\frac{1}{2^{t(n)}} = \frac{1}{O(n)} = \Omega(n^{-1})$ درست جواب می‌دهیم که احتمال موفقیت غیرناچیز است، پس حمله کننده‌ای که بدون استفاده از اوراکل، تصادفی جواب دهد به احتمال غیرناچیز جعل می‌کند و در نتیجه MAC ما امن نیست.

مسأله‌ی ۶

یک خانواده از توابع $\{h_k : \{0,1\}^{m(k)} \to \{0,1\}^{l(k)}\}$ که توسط الگوریتم چند جمله‌ای G تولید شده‌اند را pre-image resistant hash functions گوییم اگرکه، $\forall k : |m(k)| > |l(k)|$ و h_k در زمان چند جمله‌ای بر حسب k قابل محاسبه باشد و برای هر حمله‌کننده‌ی PPT مثل A تابع ناچیز $\epsilon(\cdot)$ وجود داشته باشد که:

$$\forall n : \Pr[k \leftarrow G(1^n); y \leftarrow \{0,1\}^{m(k)}; x \leftarrow A(k, 1^n, h(y)); h_k(x) = h(y)] < \epsilon(n)$$

یک خانواده از توابع $\{h_k : \{0,1\}^{m(k)} \to \{0,1\}^{l(k)}\}$ که توسط الگوریتم چند جمله‌ای G تولید شده‌اند را second pre-image resistant hash functions گوییم اگرکه، $\forall k : |m(k)| > |l(k)|$ و h_k در زمان چند جمله‌ای بر حسب k قابل محاسبه باشد و برای هر حمله‌کننده‌ی PPT مثل A تابع ناچیز $\epsilon(\cdot)$ وجود داشته باشد که:

$$\forall n : \Pr[k \leftarrow G(1^n); x \leftarrow \{0,1\}^{m(k)}; x' \leftarrow A(k, 1^n, x); h_k(x') = h_k(x) \wedge x \neq x'] < \epsilon(n)$$

حال ثابت می‌کنیم اگر خانواده‌ای از هش‌ها second pre-image resistant نباشند، collision resistant نیز نیستند، برای اثبات کافی است حمله‌کننده‌ی A را که به احتمال غیرناچیز second pre-image پیدا می‌کند را در نظر بگیریم، حال حمله کننده‌ی A' را می‌سازیم که به شکل زیر عمل می‌کند:

$$A'(k, 1^n) : x \leftarrow \{0,1\}^{m(k)}; x' \leftarrow A(k, 1^n, x); \text{outputs}(x, x')$$

در واقع A' یکی از رشته‌ها را تصادفی ساخته و second pre-image آن‌را از A می‌گیرد و به احتمالی که A پیری‌امیج پیدا می‌کند، کالیژن پیدا می‌کند.
همچنین اگر خانواده‌ای از رمزها pre-image resistant نباشد، second pre-image resistant نیز نیست، برای اثبات فرض کنید حمله کننده‌ی A برای خانواده‌ی هش‌های h_k وجود دارد که با احتمال غیرناچیز pre image پیدا می‌کند. حال حمله کننده‌ی A' را می‌سازیم که second pre-image پیدا کند. به این شکل که:

$$A'(k, 1^n, x) : y \leftarrow h_k(x); \textbf{outputs} A(k, 1^n, y)$$

که با احتمال احتمال second pre-image پیدا می‌کند.

مسأله‌ی ۷

اولا G_1 collision resistant است، چراکه اگر پیدا کردن کالیژن برایش سخت نباشد، m_1, m_2 ای پیدا می‌کنیم که

$$G_1(m_1) = G_1(m_2) \implies H_2(H_1(m_1)) = H_2(H_1(m_2))$$

در نتیجه یا $H_1(m_1) = H_1(m_2)$ که در نتیجه H_1 collision resistant نیست یا $H_1(m_1) \neq H_1(m_2)$ که در نتیجه H_2 collision resistant نیست.
همچنین G_2 نیز collision resistant است، چراکه اگر پیدا کردن کالیژن برایش سخت نباشد، m_1, m_2 ای پیدا می‌کنیم که

$$G_2(m_1) = G_2(m_2) \implies H_1(m_1)||H_1(m_1) = H_1(m_2)||H_1(m_2) \implies H_1(m_1) = H_1(m_2)$$

و در نتیجه H_1 نیز collision resistant نیست.
حال اگر H_2 collision resistant نباشد، برای G_2 تغییری ایجاد نمی‌شود چراکه به H_2 ربطی ندارد، اما G_1 لزوما collision resistantبودنش را از دست نمی‌دهد، چرا که نیاز به دوپیام c_1, c_2 داریم که $H_2(c_1) = H_2(c_2)$ و c_1, c_2 جوابی برای H_1 باشند. اما ممکن است اصلا کالیژنی که می‌توانیم پیدا کنیم برای طولی بیشتر از طول خروجی H_1 باشد.
اما G_3 collision resistant نیست، چرا m_1, m_2 که $H_2(m_1) = H_2(m_2)$ شود، داریم:

$$H_1(H_2(m_1)) = H_1(H_2(m_2)) \implies G_3(m_1) = G_3(m_2)$$

.

مسأله‌ی ۸

آ)

فرض کنید H collision resistant نباشد، پس حمله کننده‌ی PPT مثل A موجود باشد که

$$Pr[k \leftarrow \mathsf{Gen}(1^n); (x_1, x_2) \leftarrow A(k, 1^n); x_1 \neq x_2 \wedge H^k(x_1) = H^k(x_2)]$$

غیرناچیز باشد، چون $H^{s_1,s_2}(x) = H^{s_1,s_2}(y) \implies H_1^{s_1}(x) = H_1^{s_1}(y) \wedge H_2^{s_2}(x) = H_2^{s_2}(y)$ پس اگر حمله کننده‌ی A_1 را بسازیم که به صورت زیر کار کند:

$$A_1(s_1, 1^n) : s_2 \leftarrow \mathsf{Gen}_2(1^n); x_1, x_2 \leftarrow A(\langle s_1, s_2 \rangle, 1^n) \text{outputs} x_1, x_2$$

حال A_1 یک حمله کننده با مزیت غیرناچیز برای H_1 است، همچنین به طریق مشابه

$$A_2(s_2, 1^n) : s_1 \leftarrow \mathsf{Gen}_1(1^n); x_1, x_2 \leftarrow A(\langle s_1, s_2 \rangle, 1^n) \text{outputs} x_1, x_2$$

نیز یک حمله کننده با مزیت غیرناچیز برای H_2 است. پس اگر H collision resistant نباشد، هر دوی H_1, H_2 collision resistantنیستند، پس اگر یکی collision resistant باشد، H نیز collision resistant است.

ب)

مسأله‌ی ۹

مسأله‌ی ۱۰

مسأله‌ی ۱۱

در این مسئله برای سادگی $E_x(y)$ را الگوریتم AES روی y با کلید x در نظر گرفته و $D_x(y)$ را الگوریتم AES^{-1} روی y با کلید x در نظر می‌گیریم.

آ)

باید x_1, y_1, x_2, y_2 پیدا کنیم که:

$$E_{y_1}(x_1) \oplus y_1 = E_{y_2}(x_2) \oplus y_2 \iff x_2 = D_{y_2}(E_{y_1}(x_1) \oplus y_1 \oplus y_2)$$

پس با انتخاب مقادیر دلخواه برای x_1, y_1, y_2 می‌توان x_2 را محاسبه کرد و یک کالیژن پیدا کرد.

ب)

باید x_1, y_1, x_2, y_2 پیدا کنیم که:

$$E_{x_1}(x_1) \oplus y_1 = E_{x_2}(x_2) \oplus y_2 \iff E_{x_1}(x_1) \oplus y_1 \oplus E_{x_2}(x_2) = y_2$$

پس پس با انتخاب مقادیر دلخواه برای x_1, y_1, x_2 می‌توان y_2 را محاسبه کرد و یک کالیژن پیدا کرد.

مسأله‌ی ۱۲

مسأله‌ی ۱۳

آ)

در زمزنگاری داریم:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F_{k_i}(R_{i-1})$$

پس در تابع وارون خواهیم داشت:

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus F_{k_i}(L_i)$$

پس می‌توان L_0, R_0 را از روی L_2, R_2 محاسبه کرد.

ب)

برای ورودی 0^{64} خروجی برابر با $F_{k_1}(0^{32})||F_{k_2}(F_{k_1}(0^{32}))$ خواهد بود و برای $0^{32}1^{32}$ خروجی برابر با $F_{k_1}(0^{32})||F_{k_2}(1^{32} \oplus F_{k_1}(0^{32}))$ است، پس کافی است چک کنیم کدام گزینه ایکس‌اور نیمه‌های اولشان برابر با 1^{32} خواهد شد. که مورد سوم چنین است، پس مورد سوم پاسخ است.

ج)

کافی است حمله کننده 0^{64} و $0^{32}1^{32}$ را به اوراکل بدهد و ۳۲ بیت اول خروجی را باهم ایکس‌اور کند، اگر 1^{32} شد ۱ بیرون دهد و در غیر اینصورت ۰ بیرون دهد.

اگر تابع E باشد به احتمال یک، یک بیرون داده و اگر تابع یک تابع تصادفی باشد به احتمال 2^{-32} یک بیرون می‌دهد و در نتیجه مزیت حمله کننده برابر با $1 - 2^{-32}$ است.