مقدمه‌ای پیشرفته بر رمزنگاری ۱۴ آذر ۱۳۹۱

جلسه‌ی ۱۴: توابع شبه‌تصادفی

مدرس: شهرام خزائی *نگارنده: مقصود پرویز*

۱ توابع تصادفی و شبه‌تصادفی

۱.۱ توابع تصادفی

در جلسات پیش دیدیم که برای داشتن سیستمی که دارای امنیت چند پیامی باشد، می‌توان از توابع تصادفی استفاده کرد. حال ببینیم توابع تصادفی چه هستند. هرگاه روی مجموعه توابع $f : \{0,1\}^n \to \{0,1\}^n$ توزیع یکنواخت قرار دهیم هر برآمد[1] یک تابع تصادفی خواهد بود. RF_n را توزیع یکنواخت توابع تصادفی $f : \{0,1\}^n \to \{0,1\}^n$ می‌گیریم. می‌دانیم تعداد کل توابع که n بیت را به n بیت می‌نگارند برابر است با 2^{n2^n} (چرا؟). بنابراین می‌توان این توزیع را U_{n2^n} نیز در نظرگرفت و لذا هر برآمد از U_{n2^n} یک تابع تصادفی است. توابع تصادفی $f : \{0,1\}^n \to \{0,1\}^n$ را به دو طریق می‌توان توصیف کرد.

۱) توصیف ترکیبیاتی (یکباره): در واقع در این نوع توصیف، تابع f را به عنوان یک جدول (آرایه‌ی) بزرگ، که مقادیر f را ذخیره می‌کند، می‌نگریم و در این حالت $f(x)$ مقدار ذخیره شده در x امین، n بیتی است.

0101...	1101...	0010...	...	0100...

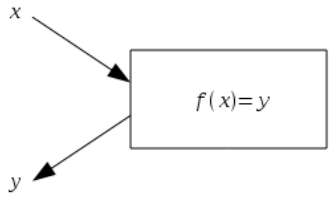
شکل ۱: جدول مقادیر f

بنابراین در این حالت طول توصیف f برابر $n2^n$ خواهد بود. در نتیجه 2^{n2^n} تابع تصادفی $f : \{0,1\}^n \to \{0,1\}^n$ وجود دارند. به زبان ساده در این توصیف کل جدول مقادیر تابع به دشمن داده می شود.

۲) توصیف محاسباتی: در این توصیف تابع به عنوان ماشینی نگریسته می‌شود که با گرفتن ورودی، خروجی را به صورت تصادفی انتخاب می‌کند و جواب‌های قبلی را پیگیری می‌کند. در واقع اگر $x \in \{0,1\}^n$ را به عنوان ورودی دریافت کند، در صورتی که این مقدار را قبلا دریافت نکرده باشد، آنگاه مقدار $y \leftarrow \{0,1\}^n$ را تولید و سپس $f(x) = y$ را ذخیره می‌کند. اگر x را قبلا دیده باشد با یک نگاه به جدولی که قبلا ذخیره کرده است، مقدار $f(x)$ را خارج می‌کند. به زبان ساده هرگاه دشمن خروجی تابع را برای یک ورودی درخواست کرد، یک مقدار تصادفی

[1] sample

انتخاب و به دشمن داده می‌شود. برای این که شرط تابع بودن برقرار شود باید مقدار تابع ذخیره شود تا اگر دشمن بار دیگر همان ورودی را داد، خروجی متناظر را تحویل بگیرد. می‌توان نشان داد که دو توصیف بالا از توابع به یک



شکل ۲: محاسبه و ذخیره ی f

توزیع یکسان می‌رسند.
مشکل توابع تصادفی این است که طبق تعریف، توصیف طولانی دارند (یعنی برای ذخیره‌ی آن‌ها حافظه‌ی زیادی نیاز است) بنابراین نمی‌توان آن‌ها را در طرح‌های رمزنگاری به کار برد. به همین منظور توابع شبه‌تصادفی را تعریف می‌کنیم که مانند توابع تصادفی به نظر می‌رسند اما توصیف کوتاهی دارند.

۲.۱ تمایزناپذیری اُراکلی

به طور نیمه شهودی یک تابع شبه‌تصادفی برای هر دشمن nuPPT شبیه یک تابع تصادفی به نظر می‌رسد. برای تعریف این مفهوم نیاز به مفاهیمی مانند دسترسی اُراکلی به یک تابع داریم. به طور شهودی، اُراکل، ماشینی است که می‌توان از آن سوال پرسید و بی‌درنگ جواب گرفت.

تعریف ۱ *یک ماشین اُراکلی (قطعی، احتمالاتی)، یک ماشین تورینگ با یک نوار اضافی (به نام نوار اُراکل) و دو حالت خاص به نام‌های OI (احضار اراکل[۲]) و OA (ظهور اراکل[۳]) است. محاسبه‌ی یک ماشین اُراکلی قطعی M روی ورودی z و با دسترسی به تابع $f : \{۰,۱\}^* \to \{۰,۱\}^*$ (که آن را اُراکل می‌نامیم) با رابطه‌ی پیکربندی متوالی[۴] تعریف می‌شود. در واقع هرگاه حالت درونی ماشین[۵] با OI متفاوت باشد، همانند ماشین تورینگ زمینه عمل می‌کند. در غیر این صورت هرگاه γ یک پیکربندی با محتوای x روی نوار اُراکل باشد آنگاه γ ثابت می‌ماند، حالت درونی ماشین به OA تغییر می‌کند و محتوای نوار اراکل $f(x)$ خواهد شد. رشته‌ی x، پرسمان[۶] M و $f(x)$ پاسخ اُراکل نامیده می‌شود. ماشین اُراکلی احتمالاتی به گونه‌ای مشابه تعریف می‌شود. توزیع ماشین اُراکلی M روی ورودی z و دسترسی به اُراکل f با $M^{f(\cdot)}(z)$ نمایش داده می‌شود.*

تعریف ۲ *گیریم $\{O_n\}_{n\in\mathbb{N}}$ و $\{O'_n\}_{n\in\mathbb{N}}$ خانواده‌هایی باشند که O_n و O'_n توزیع‌های احتمال روی توابع $f : \{۰,۱\}^n \to \{۰,۱\}^n$ هستند. گوییم $\{O_n\}_{n\in\mathbb{N}}$ و $\{O'_n\}_{n\in\mathbb{N}}$ از نظر محاسباتی تمایزناپذیرند هرگاه برای هر nuppt الگوریتم D تابع ناچیز ε موجود باشد که برای هر $n \in \mathbb{N}$ ،*

[۲] oracle invocation
[۳] oracle appered
[۴] successive-configuration
[۵] state
[۶] query

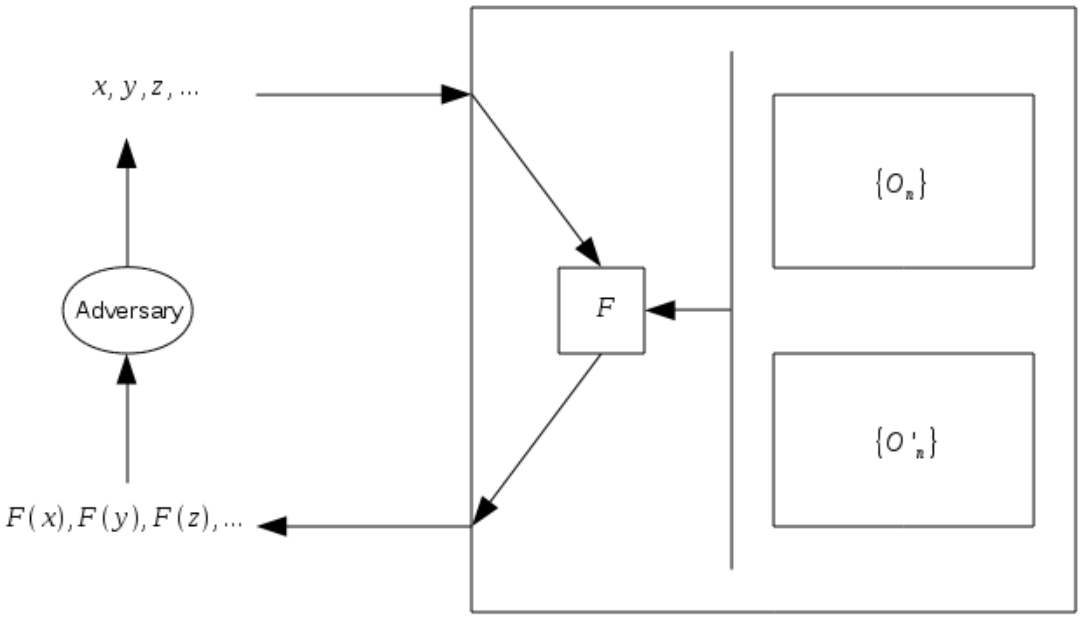
$$|\Pr[f \leftarrow O_n : D^{f(\cdot)}(1^n) = 1] - \Pr[f \leftarrow O'_n : D^{f(\cdot)}(1^n) = 1]| < \varepsilon(n)$$

این را با $\{O_n\}_{n\in\mathbb{N}} \approx \{O'_n\}_{n\in\mathbb{N}}$ *نمایش می‌دهیم.*

نکته ۱ *می‌توان تعریف را به مجموعه توابع* $f : \{0,1\}^{l_1(n)} \to \{0,1\}^{l_2(n)}$ *برای چندجمله‌ای‌های* l_1 *و* l_2 *تعمیم داد.*

نکته ۲ *توجه شود که در اینجا رشته‌ی ورودی به* f *داده می‌شود و* 1^n *داده‌ی کمکی است.*

نکته ۳ *بد نیست در اینجا به تفاوت بین تمایزناپذیری بین دو توزیع و تمایزناپذیری اُراکلی اشاره کنیم. در تمایزناپذیری اُراکلی دسترسی به یک تابع فرض شده است که دشمن می‌تواند به آن ورودی دهد و خروجی را ببیند (شکل ۳).*



شکل ۳: پرسش دشمن و پاسخ اُراکل به آن

می‌توان نشان داد بستار تحت عمل کارا[۷]، لم هایبرید[۸] و لم پیشبینی[۹] در اینجا نیز برقرارند (ثابت کنید!).

۳.۱ توابع شبه‌تصادفی

تعریف ۳ *یک خانواده از توابع* $\{f_s : \{0,1\}^{|s|} \to \{0,1\}^{|s|}\}_{s\in\{0,1\}^*}$ *شبه‌تصادفی است هرگاه*

آ) $f_s(x)$ *را بتوان توسط یک* PPT *الگوریتم با ورودی‌های* s *و* x *محاسبه کرد.*

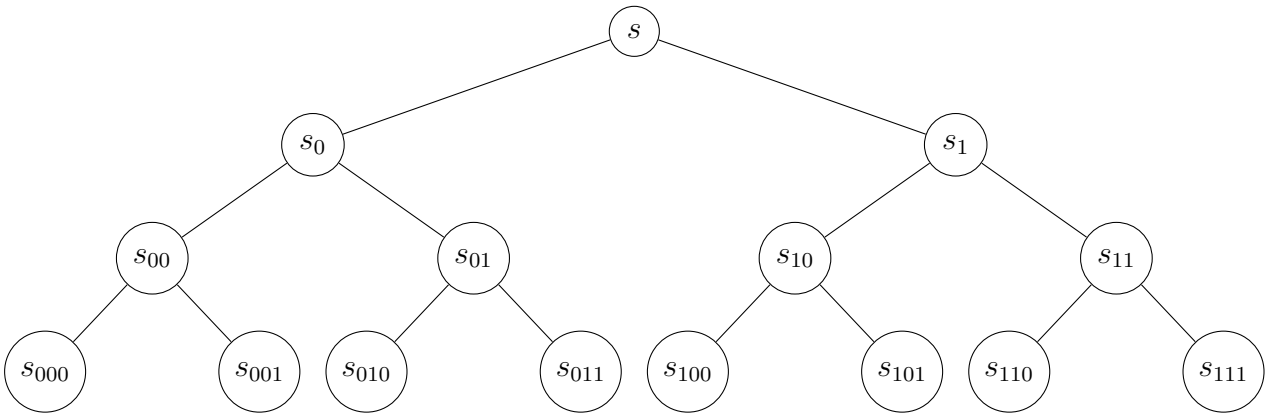
ب) $\{s \leftarrow \{0,1\}^n : f_s\}_{n\in\mathbb{N}} \approx \{F \leftarrow RF_n : F\}_{n\in\mathbb{N}}$ *که در اینجا منظور، تمایزناپذیری اُراکلی است.*

[۷] closure under efficient operations
[۸] hybrid lemma
[۹] prediction lemma

در این تعریف ضروری است که مقدار s در PRF مخفی بماند. زیرا در غیر این صورت f_s از یک تابع تصادفی قابل تمایز است. این به این دلیل است که دشمن با دادن یک x تصادفی به اُراکل و گرفتن مقدار آن، چون خودش هم با داشتن s می‌تواند $f_s(x)$ را محاسبه کند، این دو مقدار را مطابقت می‌دهد. در صورتی که یکسان باشند، با احتمال تقریبا ۱، f_s و در غیر این صورت RF انتخاب شده است.
اگر با دیدن این تعریف فکر می‌کنید که می‌توان از روی تعداد توابع حدس زد که تابع تصادفی است یا از یک خانواده‌ی دیگر انتخاب شده است، باید مفهوم تمایزناپذیری اُراکلی را دوباره بخوانید و آن را با تمایزناپذیری دو توزیع مقایسه کنید.

قضیه ۱ *اگر مولد شبه‌تصادفی وجود داشته باشد، آنگاه توابع شبه‌تصادفی موجودند.*

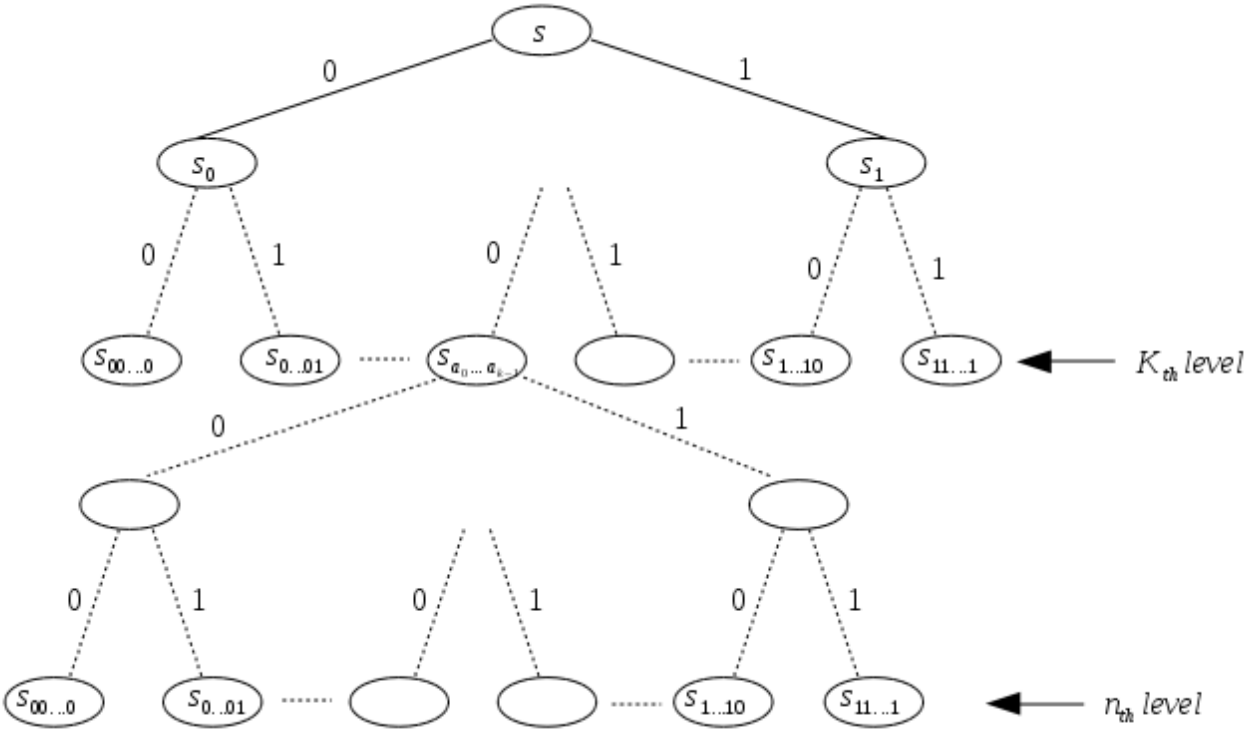
برهان. فرض کنیم g یک مولد شبه‌تصادفی باشد که طول خروجی آن دو برابر طول ورودی‌اش باشد (این مولد قبلا ساخته شده است). بنابراین می‌توانیم بنویسیم $g = g_0||g_1$ که در آن هر دو g_0 و g_1 توابعی هستند که n بیت را به n بیت می‌نگارند. در این صورت تابع f_s را به صورت زیر تعریف می‌کنیم
$f_s(b_0b_1\cdots b_{n-1}) = g_{b_{n-1}}(\cdots(g_{b_1}(g_{b_0}(s)))\cdots)$
در واقع f_s در هر محاسبه نیمی از g را نگه می‌دارد. در نتیجه خروجی‌های ممکن f_s یک درخت تشکیل می‌دهند که برگ‌هایش مقدار خروجی f_s است.



$$f_s(001) = s_{001} = g_1(s_{00}) = g_1(g_0(s_0)) = g_1(g_0(g_0(s)))$$

به وضوح f_s به طور کارا قابل محاسبه است. تمایزناپذیری را با استفاده از لم هایبرید ثابت می‌کنیم. هایبریدها را با استفاده از عمق k از درخت تعریف می‌کنیم. در واقع هایبرید k ام عبارت است از زیر تابعی که با استفاده از ساختار درخت، در یکی از رئوس درخت، که به طور یکنواخت انتخاب شده است، ساخته می‌شود. برای تعریف دقیق ابتدا برای هر انتخاب $s_0, s_1, \ldots, s_{2^k-1} \in \{0,1\}^n$ تابع $f_{s_0,s_1,\ldots,s_{2^k-1}} : \{0,1\}^n \to \{0,1\}^n$ را به صورت زیر تعریف می‌کنیم
$f_{s_0,s_1,\ldots,s_{2^k-1}}(b_0b_1\cdots b_{n-1}) = g_{b_{n-1}}(\cdots(g_{b_{k+1}}(g_{b_k}(s_{b_0b_1\cdots b_{k-1}})))\cdots)$
که در آن $s_{b_0b_1\cdots b_{k-1}} = s_{\sum_{i=0}^{k-1} b_i 2^i}$ سپس قرار می‌دهیم

$H_n^k = f_{U_n^1,\ldots,U_n^{2^k}}$
که در آن U_n^j ها متغیرهای تصادفی مستقل با توزیع یکنواخت روی $\{0,1\}^n$ هستند. به وضوح فضای نمونه‌ای H_n^k خانواده‌ای از توابع با اندازه‌ی حداکثر 2^{n2^k} است. می‌توان این را به این صورت تصور کرد که با k بیت ابتدایی ورودی یکی از s ها در عمق k ام به تصادف و با توزیع یکنواخت انتخاب می‌شود و با باقیمانده‌ی بیت‌ها با استفاده از g، مقدار تابع حساب خواهد شد.



شکل ۴: یکی از اعضای هایبرید k ام

به وضوح در این حالت $H_n^0 = \{s \leftarrow \{0,1\}^n : f_s\}$ و $H_n^n = RF_n$. حال اگر حکم برقرار نباشد (یعنی خانواده‌ی تعریف شده از خانواده‌ی توابع تصادفی تمایزپذیر نباشد) طبق لم هایبرید i ای موجود است که H_n^i و H_n^{i+1} با مزیتی غیر قابل اغماض مانند $\epsilon(n)$ تمایزپذیرند. تفاوت این دو در آن است که گام (منظور همان level در نمودار بالا است) i ام در H_n^i عبارت است از $g(U_n)$ و در H_n^{i+1} عبارت است از U_n (چرا؟ می‌توانید با رجوع به تعریف H_n^i و H_n^{i+1} و نمودار درختی به این پرسش پاسخ دهید). برای اتمام کار یک بار دیگر از لم هایبرید استفاده می‌کنیم (یعنی هایبرید در هایبرید!).
می‌دانیم تعداد پرسش‌های D توسط یک چندجمله‌ای مانند p محدود شده است. هایبرید HH_j^n را به این صورت تعریف می‌کنیم که F ای را از H_n^i انتخاب می‌کند و j پاسخ اول را با F می‌دهد و $p(n) - j$ پاسخ باقیمانده را توسط عضوی از H_n^{i+1} می‌دهد. طبق لم هایبرید j موجود است که D می‌تواند HH_n^j و HH_n^{j+1} را با مزیت $\epsilon(n)/p(n)$ تمایز دهد. تفاوت HH_n^j و HH_n^{j+1} در این است که HH_n^{j+1} ، $j+1$ پاسخ خود را با خروجی

یک مولد شبه‌تصادفی روی مقدار تصادفی انتخاب شده می‌دهد در حالی که HH_n^j، $j+1$ امین پاسخ خود را با یک مقدار تصادفی انتخاب شده می‌دهد. چون پرسش‌های HH_n^j در PPT انجام می‌گیرد، طبق بستار تحت عمل کارا، D شبه‌تصادفی بودن g را نقض می‌کند.