به نام خدا

جلسه هشتم :

- Authenticated Encryption
- Hash Functions Intro.
- Merkle-Damgard Transform
- Hash-and-MAC paradigm

- رمزنگاری تصدیق شده : (Authenticated Encryption (AE))

← هدف ایجاد یک کانال ارتباطی امن با حفظ امانت و پنهان بودن پیام است.

ایده : استفاده از امنیت CCA به همراه MAC

برای ساخت یک سیستم رمزنگاری تصدیق شده، ابتدا باید امنیت جعل‌ناپذیری را برای یک سیستم رمزنگاری تعریف کنیم.

- آزمایش جعل‌ناپذیری برای یک سیستم رمزنگاری : $Enc\text{-}Forge_{A,\Pi}(n)$

1. $k \leftarrow Gen(1^n)$
2. $c \leftarrow A^{Enc_k(\cdot)}(1^n)$
3. $m := Dec_k(c)$, $Q := \{all\ queries\ asked\}$.

خروجی آزمایش برابر ۱ است اگر و تنها اگر $m \neq \perp$ و $m \notin Q$

تعریف : یک سیستم رمز متقارن Π را جعل‌ناپذیر می‌نامیم اگر برای هر مهاجم PPT مانند A ، تابع ناچیز negl موجود باشد به طوری که :

$$Pr[Enc\text{-}Forge_{A,\Pi}(n) = 1] \leq negl(n)$$

← می‌توان تعریف قوی‌تری را در نظر گرفت که در آن مهاجم A به اوراکل رمزگشایی نیز دسترسی اوراکلی دارد.

- تعریف رمزنگاری تصدیق شده : یک سیستم رمزنگاری متقارن را یک سیستم رمزنگاری تصدیق شده می‌نامیم اگر دارای امنیت CCA داشته و جعل‌ناپذیر باشد.

- ساختار کلی AE :

می‌خواهیم با استفاده از یک سیستم رمزنگاری امن و یک سیستم کد اصالت‌سنجی، یک سیستم رمزنگاری تصدیق شده بسازیم.

* هر ترکیبی از این دو سیستم منجر به یک سیستم رمزنگاری تصدیق شده نمی‌شود.

← فرض کنید $\Pi_E=(Enc,Dec)$ یک سیستم رمزنگاری CPA-امن و $\Pi_M=(Mac,Verify)$ یک سیستم MAC باشد. (الگوریتم تولید کلید در هر ۲ سیستم، انتخاب کلید به صورت تصادفی می‌باشد)

۳ راه برای ترکیب سیستم رمز و کد اصالت‌سنجی با دو کلید مستقل k_E و k_M وجود دارد :

۱- Encrypt-and-authenticate :

$$c \leftarrow Enc_{k_E}(m) \quad , \quad t \leftarrow Mac_{k_M}(m)$$

* یک MAC امن تضمینی برای پنهان نگه داشتن پیام ندارد.

اگر یک MAC قطعی مانند CBC-MAC استفاده شود، برچسب برای یک پیام ثابت خواهد بود که باعث می‌شود سیستم به دست آمده از این روش CPA-امن نیز نباشد.

* اگر MAC ها در عمل قطعی هستند.

۲- Authenticate-then-encrypt :

$$t \leftarrow Mac_{k_M}(m) \quad , \quad c \leftarrow Enc_{k_E}(m \| t)$$

* سیستم رمز CBC-mode با پدینگ را در نظر بگیرید: ابتدا پیام پد می‌شود ($m\|t$ پد می‌شود) و پس با استفاده از CBC-mode رمز می‌شود. هنگام رمزگشایی ۲ خطا به وجود می‌آید :

۱- $\tilde{m} := Dec_{k_E}(c)$ ، اگر پدینگ درست نبود، خطای "bad padding" خروجی داده می‌شود.

$Dec'_{k_E,k_M}(c)$:

۲- $\tilde{m} := m\|t$. اگر $Verify_{k_M}(m,t)=1$ ، m را خروجی می‌دهد و در غیر اینصورت "authentication failure" را خروجی می‌دهد.

اگر هکر بتواند این ۲ خطا را از هم تمیز دهد، سیستم رمز امن نخواهد بود.

* حمله های به این شکل در IPsec رخ داده بود.

* حل کردن این مشکل با ایجاد تفاوت بین پیام خطا نیز منجر به ایجاد مشکلات دیگر می‌شود.

← در SSL این تلاش انجام شد اما همچنان حمله Padding-oracle به آن قابل انجام بود.

۳- Encrypt-then-authenticate :

$$c \leftarrow Enc_{k_E}(m) \quad , \quad t \leftarrow Mac_{k_M}(c)$$

* ثابت می‌شود که اگر MAC دارای امنیت قوی باشد، این روش منجر به یک سیستم امن خواهد شد.

امنیت قوی MAC تضمین می‌کند که هیچ مهاجمی قادر به تولید یک متن رمز شده جدید نخواهد بود.

← ساختار بدست آمده از این روش جعل‌ناپذیر خواهد بود.

از آنجایی که برای هر $\langle c,t \rangle$ که مهاجم به اوراکل رمزگشایی می‌فرستد، یا خود پیام را می‌داند و یا خروجی یک پیام خطا خواهد بود، دسترسی به اوراکل رمزگشایی بی استفاده خواهد بود و امنیت CCA تضمین می‌شود.

← اگر Π_E CPA-امن باشد، سیستم نهایی CCA-امن خواهد بود.

* اگر Π_E یک سیستم رمز متقارن CPA-امن باشد، و Π_M یک کد احراز اصالت پیام با امنیت جعل‌ناپذیری قوی باشد، سیستم رمزنگاری بر پایه encrypt-then-authenticate یک سیستم رمزنگاری تضمین شده خواهد بود.

(تمرین: اثبات کنید)

* وجود کلیدهای مستقل ضروری است.

در رمزنگاری، سیستم‌های حفاظت باید از کلیدهای مستقل استفاده کنند.

به عنوان مثال فرض کنید F یک جایگشت شبه‌تصادفی باشد. برای $m \in \{0,1\}^{n/2}$ و $r \in \{0,1\}^{n/2}$ سیستم‌های $Enc_k(m) := F_k(m \| r)$ و $Mac_k(c) = F^{-1}_k(c)$ را در نظر بگیرید. می‌توان نشان داد این سیستم رمزنگاری امنیت CPA دارد و سیستم Mac نیز امن است.

روش encrypt-then-authenticate که برای هر دو سیستم از کلید k استفاده می‌کنه، به صورت زیر خواهد بود:

$$Enc_k(m), Mac_k(Enc_k(m)) = F_k^{-1}(F_k(m||r)) = m||r$$

- ایجاد ارتباط امن:

← از کاربردهای رمزنگاری تصدیق شده.

فرض کنید $\Pi=(Enc,Dec)$ یک سیستم رمز تصدیق شده باشد و A و B با کلید مشترک k می‌خواهند ارتباطی امن داشته باشند.

ایدهٔ ابتدایی استفاده از Π است. اما این روش در مقابل بعضی حملات آسیب پذیر است:

← Re-ordering attack: مهاجم ترتیب پیام‌ها را به نوعی تغییر دهد. این حمله باعث می‌شود که دو طرف ارتباط را به یک شکل نبینند، که می‌تواند موجب بروز مشکل شود.

← Replay attack: مهاجم می‌تواند متن رمزی‌ای که قبلا فرستاده شده است را دوباره بفرستد. این موضوع نیز مانند قبل می‌تواند موجب بروز مشکل شود.

← Reflection attack: مهاجم می‌تواند متن رمزی‌ای که A به B فرستاده است را گرفته و دوباره آن را برای A بفرستد.

* با استفاده از یک شمارنده (counter) می‌توان از دو حمله اول، و با استفاده از یک بیت برای تعیین جهت ارتباط (directionality bit) از حمله آخر جلوگیری کرد.

A		B
$ctr_{A,B}, ctr_{B,A} \leftarrow 0$		$ctr_{A,B}, ctr_{B,A} \leftarrow 0$
$b_{A,B} \leftarrow_R \{0,1\}$ $b_{B,A} \leftarrow \overline{b}_{A,B}$		$b_{A,B}, b_{B,A}$
$c \leftarrow Enc_k(b_{A,B}\|\|ctr_{A,B}\|\|m)$ $ctr_{A,B} \leftarrow ctr_{A,B}+1$	$\xrightarrow{c}$	if $Dec_k(c)=\perp$: reject $b\|\|ctr\|\|m \leftarrow Dec_k(c)$ if $b \neq b_{A,B}$: reject $ctr_{A,B} \leftarrow ctr_{A,B}+1$ output m

- سیستم رمز CCA-امن :

↳ طبق تعریف ، سیستم رمز تصدیق شده ، CCA-امن است .

↳ کاربرد : زمانی که از یک سیستم رمز متقارن برای انتقال کلید استفاده می شود.

* اغلب جاهایی که به سیستم CCA-امن نیاز است ، از سیستم های رمز تصدیق شده استفاده می شود حتی اگر نیازی به جعل ناپذیری سیستم نباشد . در عمل نیز سیستم های رمز تصدیق شده کاراتر هستند .

- توابع چکیده ساز : (Hash Functions)

↳ تابعی که رشته های طولانی را به رشته های کوتاه (digest) نگاشت می کند . از شهود اصلی این توابع آن است که از برخورد (collision) خودداری کند .

* اغلب فرض می شود که توابع چکیده ساز کاملا غیرقابل پیش بینی هستند . (random oracles)
توابع چکیده ساز توابعی هستند که رشته های طولانی را به رشته های کوتاه با طول ثابت فشرده می کنند .

↳ از کاربردهای این توابع در داده ساختارهاست . (در ساخت hash table ها)
یک تابع چکیده ساز "خوب" در داده ساختارها ، برخورد کمی دارد اما در کاربردهای رمزنگاری باید از برخورد خودداری کند .

در داده ساختارها می توان فرض را بر این قرار داد که مجموعه داده ها به صورت مستقل از تابع چکیده ساز انتخاب شده اند ، اما در رمزنگاری مهاجم ممکن است داده ها را طوری انتخاب کند که منجر به برخورد شوند .

- مقاوم در برابر برخورد :

↳ اگر برای هر مهاجم PPT ، پیداکردن یک برخورد برای H ناممکن باشد .

* برخورد وجود دارد اما پیداکردن آن سخت است .

$$\Pr\left[\text{Hash-coll}_{\Lambda,\Pi}(n)=1\right] \le \text{negl}(n)$$

* در عمل، اغلب توابع هش‌هایی هستند که از طریق استاندارد ساخته شده‌اند.

• تعاریف مقدماتی:

- تصادم در یک تابع هش (preimage resistant) : یک هش $H^s(x)=H^s(x')$ است اگر $x \neq x'$ باشد.

- تصادم در برابر پیش‌تصویر (Preimage resistant) : یک هش $H^s(x)=y$ است اگر برای y, s داده شده، یافتن x ی که $H^s(x)=y$ باشد.

- مقاوم در برابر برخورد (Collision resistant) : $\Pi=(Gen,H)$ تابع هش فشرده‌ای است اگر برای هر الگوریتم احتمالی چندجمله‌ای A تابع $negl$ ی وجود داشته باشد که:

$$\Pr\left[s \leftarrow Gen(1^n); (x,x') \leftarrow A(s) : x \neq x' \text{ و } H^s(x)=H^s(x')\right] \le negl(n)$$

یعنی احتمال یافتن برخورد برای H تقریباً غیرممکن است.

* برای ساخت توابع هش مقاوم در برابر برخورد در عمل معمولاً از توابع فشرده‌ساز استفاده می‌شود.

قضیه: اگر (Gen,H) یک PPT باشد که برای هر s تابع H^s یک تابع فشرده‌ساز یک‌به‌یک باشد، آنگاه $\Pi=(Gen,H)$ یک تابع هش مقاوم در برابر برخورد است.

اثبات: یک تابع فشرده‌ساز یک‌به‌یک H^s برای هر s داده شده، یک تابع جایگشت است. در این حالت H یک تابع جایگشت است: برای محاسبه‌ی $H^s(x)$ کافی است x را برگردانیم. مقاومت در برابر برخورد: A یک PPT است که $H^s(x)=H^s(x')$ را با احتمال نامساوی محاسبه می‌کند.

* هر تابع چکیده‌ساز مقاوم در برابر برخورد، در برابر پیش‌تصویر دوم نیز مقاوم است. (ثابت کنید)

* هر تابع چکیده‌ساز مقاوم در برابر پیش‌تصویر دوم، در برابر پیش‌تصویر نیز مقاوم است. (ثابت کنید)

- استفاده از مرکل-دمگارد: Merkle-Damgard

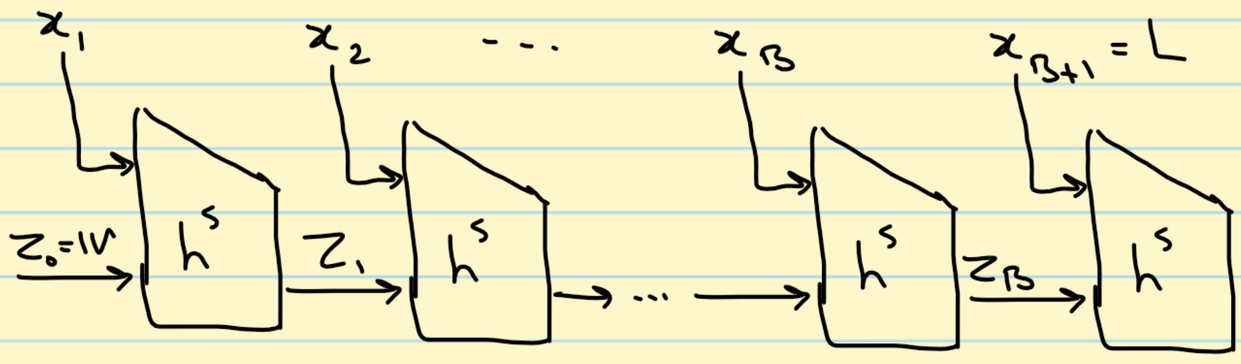
به همراه گسترش دامنه

توابع چکیده‌ساز اغلب با استفاده از توابع فشرده‌ساز مقاوم در برابر برخورد با ورودی‌هایی با طول ثابت، ساخته می‌شوند.

از استفاده مرکل-دمگارد برای ساخت توابع چکیده‌ساز مقاوم در برابر برخورد در MD5 و خانواده SHA استفاده شده است.

فرض کنید (Gen, h) یک تابع چکیده‌ساز برای ورودی‌هایی با طول $2n$ و با طول خروجی n باشد.

فرض کنید $x \in \{0,1\}^*$ با طول $L < 2^n$ ورودی (Gen, H) باشد. قرار می‌دهیم: $B := \lceil \frac{L}{n} \rceil$



* اگر (Gen, h) مقاوم در برابر برخورد باشد، (Gen, H) نیز مقاوم در برابر برخورد خواهد بود.

(تمرین: ثابت کنید)

- کد اصالت‌سنجی با استفاده از توابع چکیده‌ساز:

۱. روش اول: Hash-and-MAC : $\Pi = (Mac, Verify)$ یک MAC برای پیام‌هایی با طول n و $\Pi_H = (Gen_H, H)$ یک تابع چکیده‌ساز با طول خروجی n باشد. کد اصالت‌سنجی $\Pi' = (Gen', Mac', Verify')$ برای پیام‌هایی با طول دلخواه را به شکل زیر می‌سازیم:

$Gen'(1^n)$: $k \xleftarrow{R} \{0,1\}^n$ ، $s \leftarrow Gen_H(1^n)$. $k' := \langle k, s \rangle$

$Mac'(k', m)$: $k' = \langle k, s \rangle$ ، $m \in \{0,1\}^*$. $t \leftarrow Mac_k(H^s(m))$

$Verify'(k', m, t)$: $k' = \langle k, s \rangle$ ، $m \in \{0,1\}^*$. $Verify_k(H^s(m), t) \stackrel{?}{=} 1$: خروجی

* اگر Π یک MAC امن برای پیام‌هایی با طول l ، و Π_H متعادم در برابر برخورد باشد ، ساختار Hash-and-MAC برای پیام‌هایی با طول دلخواه امن است .

(تمرین : اثبات کنید)

۲. روش دوم ، HMAC :