**به نام خدا**

**دانشکده‌ی علوم ریاضی**

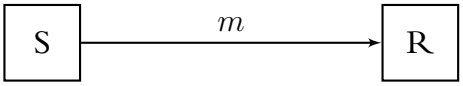
**مقدمه‌ای بر رمزنگاری** ۸ اردیبهشت ۱۳۹۲

# جلسه‌ی ۱۶: کد اصالت‌سنجی پیام

*مدرس:* دکتر شهرام خزائی *نگارنده:* محمد حسین محصولی

## ۱ اصالت پیام

تا به حال تمرکز ما معطوف به رمزنگاری پیام فرستنده به سوی گیرنده بوده است:




در این راستا امنیت رمز استفاده شده را در برابر حمله‌های شنود[۱] تحت سناریوهای مختلف متن اصلی انتخابی[۲] و متن اصلی رمزی[۳] بررسی کرده‌ایم. نکته‌ای که در این میان وجود دارد این است که این حمله‌ها از نوع حمله‌ی *منفعل*[۴] هستند.

**تعریف ۱** *حمله‌ای منفعل است که در آن مهاجم نتواند تغییری در پیام ارسالی از فرستنده به گیرنده ایجاد کند و صرفا تلاش می‌کند تا اطلاعاتی در مورد پیام بدست آورد.*

حمله کننده *فعال*[۵] با دست‌کاری در پیام‌های ارسالی سعی در خراب‌کاری دارد. یک حمله‌ی فعال تحت سناریوی متن رمزی انتخابی علیه یک سیستم رمز در نظر بگیرید. اگر سیستم رمز دارای امنیت متن رمزی انتخابی باشد، حمله کننده قادر نیست متن رمزی را طوری تغییر دهد که تبدیل به متن رمز شده‌ی متناظر با پیام دلخواهش شود. درواقع چنین سیستم رمزی، دارای امنیت قوی‌تری است: مهاجم قادر نیست متن رمز شده را طوری تغییر دهد، به طوری‌که بتواند در مورد متن اصلی متناظر آن اطلاعاتی کسب کند. با این وجود ممکن است مهاجم بتواند متن رمزی بسازد که معتبر باشد و گیرنده بتواند آنرا رمزگشایی کند. چنین ویژگی در بعضی کاربردها می‌تواند خطرناک باشد.

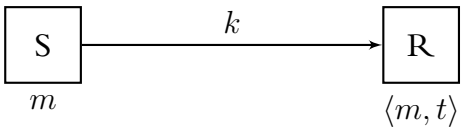
$$\begin{array}{ccc} m & \xrightarrow{\mathsf{Enc}} & c \\ \wr & & \wr \\ m' & & c' \end{array}$$

[۱] eavesdropping
[۲] chosen-plaintext attack
[۳] chosen-ciphertext attack
[۴] passive
[۵] active

سوالی که پیش می‌آید این است که چه کار می‌توان کرد تا گیرنده از اصالت پیام دریافتی مطمئن شود.
اولین چیزی که به ذهن می‌رسد این است که فرستنده باید ویژگی خاصی از پیام اصلی بدست آورده و یا مقداری را از روی آن حساب کند و بجای پیام، مجموعه‌ی پیام و مقدار محاسبه شده را برای ارسال در نظر بگیرد تا اعتبار پیام از روی این مقدار بررسی شود:

$$\langle m, f(m) \rangle$$

در عمل، تابع $f$ می‌تواند یک parity روی بیت‌های $m$ و یا checksum پیام اصلی باشد. اما در این حالات این تابع عمومی[۶] است و مهاجم با اطلاع از تابع $f$ قادر است، حمله مناسبی طراحی کند. برای حل این مشکل در اینجا نیز از یک کلید مشترک بین فرستنده و گیرنده استفاده می‌کنیم. یعنی در پایان پیام یک برچسب[۷] چون $t$ به پیام اضافه می‌کنیم که این برچسب تابعی از پیام اصلی و کلید است:




در نهایت گیرنده با دریافت $t$ بررسی می‌کند که آیا $t$ تابعی معلوم از پیام و کلید است یا خیر و به این ترتیب صحت و اصالت پیام را ملاحظه می‌کند.

## ۲ کد اصالت‌سنجی پیام

**تعریف ۲** *یک سیستم کد اصالت‌سنجی پیام*[۸] *($MAC$) به صورت* $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ *از الگوریتم‌های چندجمله‌ای تصادفی روی فضای $\mathcal{M}$ است که در آن:*

- Gen*: الگوریتم تولید کلید است که از روی پارامتر امنیت $n$ کلید $k$ را تولید می‌کند.*
  *($\mathcal{K}$ را فضای کلید، یعنی مجموعه همه خروجی‌های الگوریتم تولید کلید بگیرید.)*
- Mac*: الگوریتم تولید برچسب است و پیام $m \in \mathcal{M}$ و کلید $k \in \mathcal{K}$ را به برچسب $t \leftarrow \mathsf{Mac}_k(m)$ می‌نگارد.*
- Vrfy*: الگوریتم تایید برچسب است و پیام $m \in \mathcal{M}$، کلید $k \in \mathcal{K}$ و برچسب $t$ را می‌گیرد و خروجی یک (به معنی معتبر) یا صفر (به معنی نامعتبر) بر می‌گرداند. برای هر پیام $m$ و هر کلید $k$ داریم:*

$$\Pr\{k \leftarrow \mathsf{Gen}(1^n) ; t \leftarrow \mathsf{Mac}_k(m) : \mathsf{Vrfy}_k(\langle m, t \rangle) = 1\} = 1 .$$

*الگوریتم تولید برچسب می‌تواند قطعی یا تصادفی باشد. اما الگوریتم تایید برچسب همواره قطعی است.*

---
[۶] public
[۷] tag
[۸] Message Authentication Code

**چگونه می‌توان یک ساختار کد اصالت‌سنجی ساخت؟** برای پاسخ به این سوال ابتدا فرض می‌کنیم که پیام‌های ما طول ثابتی دارند؛ مثلاً $\mathcal{M} = \{۰, ۱\}^n$. ایده‌ی ابتدایی این است که مشابه رمزنگاری از OTP[۹] برای کد اصالت‌سنجی پیام استفاده کنیم، داریم:

- $\mathsf{Gen}(۱^n) : k \leftarrow \{۰, ۱\}^n$
- $t = \mathsf{Mac}_k(m) : t \leftarrow m \oplus k$
- $\mathsf{Vrfy}_k(\langle m, t\rangle) = \begin{cases} ۱, & \text{if } t = m \oplus k \\ ۰, & \text{otherwise} \end{cases}$

**آیا این روش برای ساخت برچسب امن است؟** برای پاسخ به این پرسش ابتدا نیاز داریم که تعریفی دقیقی از امنیت کد اصالت‌سنجی پیام ارائه کنیم که قابلیت‌های حمله‌کننده‌های فعال را در دنیای واقعی مدل می‌کند.

# ۳ امنیت سیستم کد اصالت‌سنجی پیام

حمله به یک سیستم کد اصالت‌سنجی با هدف جعل پیام صورت می‌گیرد. به عبارت دیگر مهاجم برای پیام مورد نظرش یک برچسب معتبر می‌سازد تا گیرنده به هنگام بررسی اصالت پیام، آن را معتبر تشخیص دهد. برای تعریف امنیت چنین سیستمی، ابتدا نیاز است تا آزمایش مناسبی طراحی کرد تا توان حمله‌ی مهاجم[۱۰] به این سیستم را مدل کند. طراحی این آزمایش به صورت یک بازی بین مهاجمی چون $\mathcal{A}$ و یک چالشگر[۱۱] فرضی اجرا می‌شود. در این آزمایش بررسی می‌شود که آیا مهاجم می‌تواند برای پیامی که قبلا برچسب آن را ندیده است، برچسب معتبری تولید کند یا نه. این آزمایش که با $\mathsf{MacForge}_{\mathcal{A},\Pi}$ نشان داده می‌شود، به صورت زیر اجرا می‌شود:

۱. چالشگر یک کلید $k$ تولید می کند:
$$k \leftarrow \mathsf{Gen}(۱^n)$$

۲. به مهاجم دسترسی اوراکلی به $\mathsf{Mac}_k(\cdot)$ داده می‌شود تا پیام‌های مختلف را بررسی کرده و در نهایت پس از پرسمان[۱۲]های لازم، یک زوج $\langle m, t\rangle$ خروجی دهد:
$$\langle m, t\rangle \leftarrow \mathcal{A}^{\mathsf{Mac}_k(\cdot)}(۱^n)$$

مجموعه‌ی پرسمان‌های مهاجم به هنگام بررسی را $Q$ می‌نامیم. خروجی آزمایش که با متغیر تصادفی $\mathsf{MacForge}_{\mathcal{A},\Pi}(n)$ نشان داده می‌شود یک در نظر گرفته می‌شود اگر و فقط اگر مهاجم موفق به جعل یک برچسب معتبر برای پیامی که قبلاً پرسمان نکرده است شود. به طور دقیق‌تر:

$$\mathsf{MacForge}_{\mathcal{A},\Pi}(n) = \begin{cases} ۱, & \text{if } m \notin Q \wedge \mathsf{Vrfy}_k(\langle m, t\rangle) = ۱ \\ ۰, & \text{otherwise} \end{cases}$$

[۹] One Time Pad
[۱۰] adversary
[۱۱] challenger
[۱۲] query

**تعریف ۳** *یک سیستم کد اصالت‌سنجی پیام دارای امنیت جعل‌ناپذیری[۱۳] است، هرگاه برای هر مهاجم تصادفی چون $\mathcal{A}$ که در زمان چندجمله‌ای اجرا می‌شود، یک تابع ناچیز چون $\varepsilon(n)$ یافت شود به طوری که داشته باشیم:*

$$\Pr\{\mathsf{MacForge}_{\mathcal{A},\Pi}(n) = 1\} \leqslant \varepsilon(n)$$

**نکته ۱** *تعریف فوق حمله بازپخش[۱۴] را مدل نمی‌کند. به عبارت دیگر اگر زوج $\langle m, t\rangle$ توسط فرستنده مجاز برای گیرنده ارسال شود، مهاجم می‌تواند بعدا همین پیام را از طرف فرستنده برای گیرنده ارسال کند و گیرنده آنرا بپذیرد. این بدین معنی نیست که حمله بازپخش در عمل مهم نیست؛ برعکس، حمله بازپخش در عمل می‌تواند بسیار خطرناک باشد. اما برای مقابله با آن باید چاره‌های دیگری اندیشید (مانند استفاده از مهر زمانی[۱۵]) که از موضوع این درس خارج است.*

اکنون باید واضح باشد که چرا پیشنهاد قسمت قبلی برای کد اصالت‌سنجی دارای امنیت لازم نیست. برای اثبات دقیق ادعا مهاجم $\mathcal{A}$ را به صورت زیر می‌سازیم که در آزمایش همواره برنده می‌شود (خروجی آزمایش با احتمال یک برابر ۱ می‌شود). مهاجم ابتدا پرسمان داخواه $m_1$ را به اوراکل $\mathsf{Mac}_k(\cdot)$ ارسال می‌کند و برچسب $t_1$ را دریافت می‌کند. سپس زوج $\langle m, t\rangle = \langle m_1 \oplus 1^n, t_1 \oplus 1^n\rangle$ را به خروجی می‌دهد. به وضوح داریم:

$$\Pr\{\mathsf{MacForge}_{\mathcal{A},\Pi}(n) = 1\} = 1 \ .$$

## ۱.۳ امنیت جعل‌ناپذیری کامل

فرض کنید $q$ یک عدد طبیعی ثابت و $\mathcal{T}$ فضای همه برچسب‌های ممکن باشد. می‌توان امنیت جعل‌ناپذیری کامل $q$-پیامی را بدین صورت تعریف کرد: احتمال موفقیت هر مهاجم با توان محاسباتی دلخواه که حداکثر به تعداد $q$ پرسمان انجام می‌دهد، حداکثر برابر با $\frac{1}{|\mathcal{T}|}$ باشد.

**سؤال ۴** *نشان دهید چگونه می‌توان با استفاده از یک چندجمله‌ای درجه $q$ روی یک میدان $|\mathcal{T}|$ عضوی می‌توان به امنیت جعل‌ناپذیری کامل $q$- پیامی دست یافت.*

## ۲.۳ امنیت جعل‌ناپذیری قوی

در آزمایش جعل‌ناپذیری، مجموعه‌ی همه زوج پرسمان‌های ارسالی و برچسب‌های دریافتی توسط مهاجم به هنگام بررسی را $Q'$ بنامید. یک خروجی دیگر که با متغیر تصادفی $\mathsf{SMacForge}_{\mathcal{A},\Pi}(n)$ نشان داده می‌شود برای آزمایش در نظر بگیرید که برابر یک در نظر گرفته می‌شود اگر و فقط اگر مهاجم موفق شود به یکی از دو هدف زیر دست یابد:

- جعل یک برچسب معتبر برای پیامی که قبلاً پرسمان نکرده است،
- جعل یک برچسب جدید برای پیامی که قبلاً پرسمان کرده است.

[۱۳] unforgeability
[۱۴] replay attack
[۱۵] timestamp

به طور دقیق‌تر:

$$\mathsf{SMacForge}_{\mathcal{A},\Pi}(n) = \begin{cases} ۱, & \text{if } \langle m,t\rangle \notin Q' \wedge \mathsf{Vrfy}_k(\langle m,t\rangle) = ۱ \\ ۰, & \text{otherwise} \end{cases}$$

**تعریف ۵** *یک سیستم کد اصالت‌سنجی پیام دارای امنیت جعل‌ناپذیری قوی[۱۶] است، هرگاه برای هر مهاجم تصادفی چون $\mathcal{A}$ که در زمان چندجمله‌ای اجرا می‌شود، یک تابع ناچیز چون $\varepsilon(n)$ یافت شود به طوری که داشته باشیم:*

$$\Pr\{\mathsf{SMacForge}_{\mathcal{A},\Pi}(n) = ۱\} \leqslant \varepsilon(n)$$

**قضیه ۱** *امنیت جعل‌ناپذیری قوی، امنیت جعل‌ناپذیری را نتیجه می‌دهد.*

**قضیه ۲** *اگر الگوریتم تولید برچسب قطعی باشد، امنیت جعل‌ناپذیری قوی و امنیت جعل‌ناپذیری معادلند.*

# ۴ ساخت یک کد امن اصالت‌سنجی پیام

در ابتدا بررسی می‌کنیم که چگونه می‌توان یک MAC مناسب برای رشته‌های با طول ثابت، مثلا مجموعه رشته‌های $n$-بیتی ساخت. فرض کنید مجموعه توابع $f_k : \{۰,۱\}^{|k|} \to \{۰,۱\}^{|k|}$ خانواده‌ای از توابع شبه‌تصادفی باشند. سیستم کد اصالت‌سنجی پیام $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ را روی $\mathcal{M} = \{۰,۱\}^n$ اینگونه تعریف می‌کنیم:

- $\mathsf{Gen}(۱^n) : k \leftarrow \{۰,۱\}^n$
- $\mathsf{Mac}_k(m) : \langle m, f_k(m)\rangle$
- $\mathsf{Vrfy}_k(\langle m,t\rangle) = \begin{cases} ۱, & \text{if } t = f_k(m) \wedge m,t \in \{۰,۱\}^n \\ ۰, & \text{otherwise} \end{cases}$

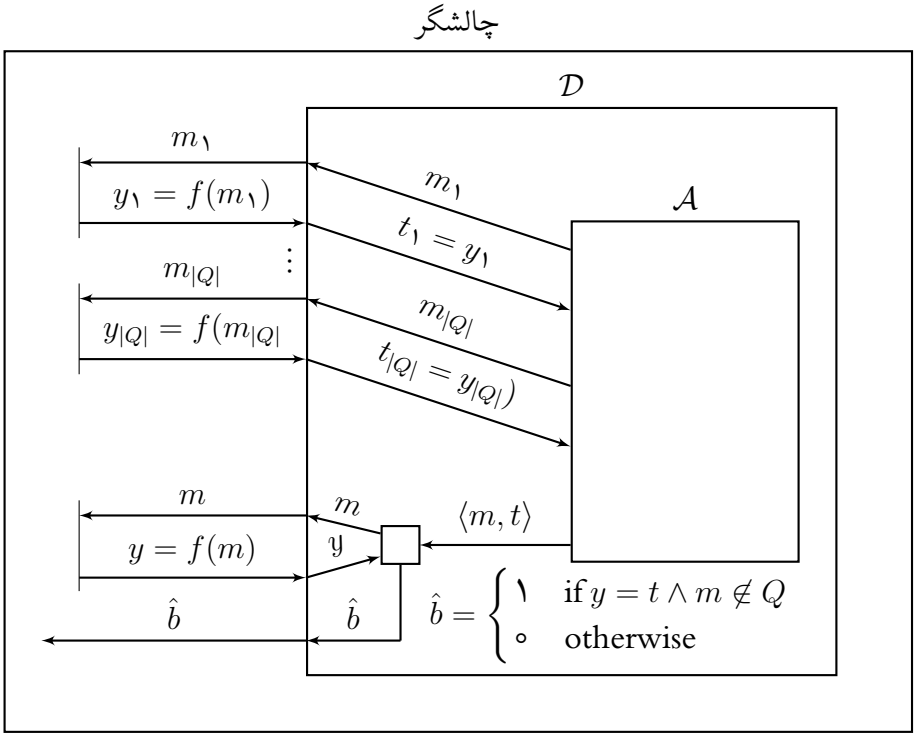
**قضیه ۳** *سیستم کد اصالت‌سنجی فوق دارای امنیت جعل‌ناپذیری است.*

**برهان.** از برهان خلف برای اثبات کمک می‌گیریم. فرض کنید این سیستم جعل‌پذیر است. پس یک مهاجم چندجمله‌ای تصادفی چون $\mathcal{A}$ و یک تابع قابل‌توجه (غیرناچیز) مثل $\mu(\cdot)$ جود دارد که:

$$\Pr\{\mathsf{MacForge}_{\mathcal{A},\Pi} = ۱\} = \mu(n) \qquad (۱)$$

در اینجا با روش کاهش[۱۷] می‌توان به کمک این مهاجم یک تمایزگر $\mathcal{D}$ برای تمایز تابع $f_k$ از توابع تصادفی ساخت:

[۱۶] strong unforgeability
[۱۷] reduction




در واقع تمایزگر $\mathcal{D}$ در آزمایشی شرکت می‌کند که دسترسی اوراکلی به تابع $f$ مورد آزمون دارد و باید تعیین کند که آیا این تابع همان $f_k$ است (که کلید $k$ کاملاً تصادفی انتخاب شده است) یا یک تابع کاملاً تصادفی است (که از بین همه توابعی که $n$ بیت را به $n$ بیت می‌نگارند به تصادف انتخاب شده است). برای تشخیص این مساله تمایزگر $\mathcal{D}$ از مهاجم $\mathcal{A}$ کمک می‌گیرد و آزمایش $\mathsf{MacForge}_{\mathcal{A},\Pi}(n)$ را برای او شبیه‌سازی می‌کند. تمایزگر $\mathcal{D}$ با دریافت هر پیام مانند $m_i$ از مهاجم $\mathcal{A}$ باید برچسب متناظر با آن، $t_i$، را به وی برگرداند. بدین منظور تمایزگر $\mathcal{D}$ که خود در حال اجرای آزمایش تمایز $f_k$ از یک تابع کاملاً تصادفی با چالشگر خود است، پرسمان $m_i$ را برای اوراکل $f$ ارسال می‌کند و پاسخ $y_i = f(m_i)$ را دریافت می‌کند. دقت کنید که تمایزگر $\mathcal{D}$ نمی‌داند که $y_i$ خروجی تابع $f_k$ است یا خروجی یک تابع کاملا تصادفی. با این وجود، تمایزگر $\mathcal{D}$ مقدار $y_i$ را به عنوان برچسب $t_i$ برای مهاجم $\mathcal{A}$ ارسال می‌کند. در نهایت نیز مهاجم $\mathcal{A}$ زوج پیام و برچسب جعلی متناظر را به صورت $\langle m, t \rangle$ به $\mathcal{D}$ می‌گرداند. حال تمایزگر $\mathcal{D}$ باید یک بیت $\hat{b}$ تولید کند. بدین منظور، تمایزگر $\mathcal{D}$ پرسمان $m$ را برای اوراکل $f$ ارسال می‌کند و پاسخ $y = f(m)$ را دریافت می‌کند. حال تمایزگر $\mathcal{D}$ (با پیش‌فرض اینکه $f = f_k$) بررسی می‌کند که آیا $\langle m, t \rangle$ یک زوج پیام و برچسب معتبر است یا خیر. به طور دقیق‌تر تمایزگر $\mathcal{D}$ بیت خروجی $\hat{b}$ را برابر یک قرار می‌دهد اگر و فقط اگر پاسخ $y$ دریافتی از اوراکل $f$ با برچسب جعلی $t$ برابر باشد و مهاجم $\mathcal{A}$ قبلاً درخواست برچسب پیام $m$ را نداده باشد. با توجه به روند فوق و رابطه (۱) داریم:

$$|\Pr\{k \leftarrow \{0,1\}^n : \mathcal{D}^{f_k(\cdot)}(1^n) = 1\} - \Pr\{f \leftarrow \mathsf{RF}_n : \mathcal{D}^{f(\cdot)}(1^n) = 1\}| \;=\; |\mu(n) - 2^{-n}|$$

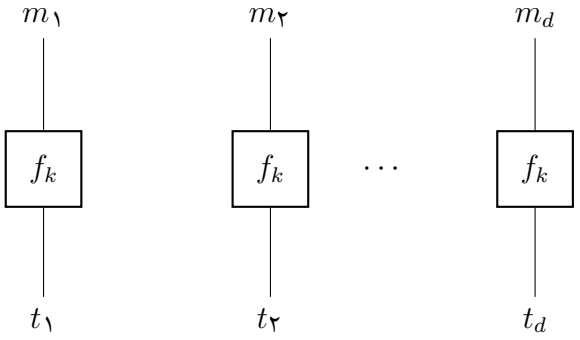
با توجه به غیرناچیز بودن $\mu(\cdot)$، احتمال فوق نیز غیرناچیز است و این یعنی تمایزگر $\mathcal{D}$ می‌تواند تابع شبه‌تصادفی $f_k$ را از توابع $n$ بیتی تصادفی تمیز دهد که این مساله امری است متناقض. پس سیستم کد اصالت‌سنجی داده شده، امن است. ■

# ۵ کد اصالت‌سنجی روی پیام‌های با طول متغیر

تا به حال سیستم‌های کد اصالت‌سنجی را روی پیام‌های با طول ثابت در نظر گرفتیم. اما در واقعیت طول پیام ثابت نیست. باز هم از یک تابع شبه‌تصادفی با طول کلید، طول ورودی و طول خروجی $n$-بیت استفاده می‌کنیم.
سیستم کد اصالت‌سنجی پیام $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ را روی رشته‌های به طول دلخواه اینگونه می‌سازیم:

**تلاش اول.** پیام ورودی $M$ را به بلوک‌های $n$ بیتی تقسیم کرده و برای آخرین بلوک (که می‌تواند تهی باشد) با اضافه کردن رشته $۱۰\cdots۰$ طول این بخش را به $n$ می‌رسانیم. یعنی در پایان پیام رشته $۱$ را قرار داده و بعد با افزودن تعدادی $۰$ (کمترین تعداد ممکن) در انتها، طول رشته حاصل را به مضربی از $n$ می‌رسانیم. سپس در این روش کد مربوط به هر بلوک را به کمک $f_k$ تولید می کنیم:

$$\mathrm{pad}(M) = M||۱۰\cdots۰ = m_۱m_۲\cdots m_d, |m_i| = n$$
$$t_i = f_k(m_i)$$
$$\langle t_۱, \cdots, t_d\rangle \leftarrow \mathsf{Mac}_k(M)$$




اما در این روش مهاجم با دانستن کد مربوط به یک پیام، می‌تواند جای دو بلوک پیام را عوض کرده و کد مربوط به پیام دیگری را جعل کند. به طور دقیق‌تر، مهاجم پس از پرسمان پیام $M_۱$ و دریافت برچسب آن، یک برچسب جعلی $T$ برای پیام $M$ به صورت زیر جعل می‌کند:

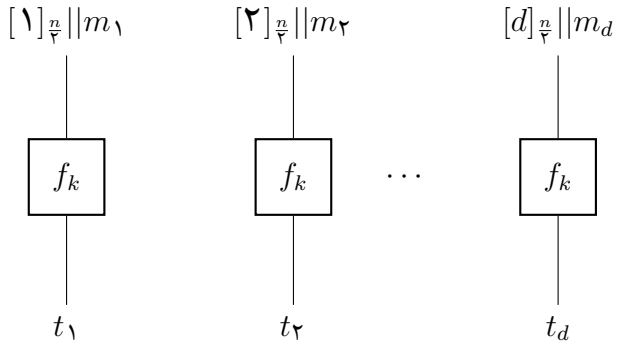
$$M_۱ = m_۱m_۲\cdots m_d \;\Rightarrow\; T_۱ = \langle t_۱, t_۲, \cdots, t_d\rangle$$

$$\langle M, T\rangle : \quad M = m_۲m_۱\cdots m_d \;\Rightarrow\; T = \langle t_۲, t_۱, \ldots, t_d\rangle$$

**سؤال ۶** *فرض کنید خروجی* $\mathsf{Mac}_k$ *را برابر* $t_۱ \oplus t_۲ \oplus \cdots \oplus t_d$ *قرار دهیم. یک حمله برای جعل برچسب بیابید.*

**تلاش دوم.** برای حل مشکل روش قبلی، برای هر بلوک، شماره‌ی بلوک را نیز با پیام اصلی به عنوان ورودی تابع شبه‌تصادفی همراه می‌کنیم. نماد $[i]_n$ برای نمایش باینری عدد $i$ توسط یک رشته $n$-بیتی را در نظر بگیرید:

$$\mathrm{pad}(M) = M||۱۰\cdots۰ = m_۱m_۲\cdots m_d, |m_i| = \frac{n}{۲}$$
$$t_i = f_k([i]_{\frac{n}{۲}}||m_i)$$
$$\langle t_۱, \cdots, t_d\rangle \leftarrow \mathsf{Mac}_k(M)$$




برای این طرح، حمله قبلی دیگر کارساز نیست اما این روش نیز جعل‌پذیر است. پرسمان $M_1$ و برچسب جعلی $T$ برای پیام جعلی $M$ را به صورت زیر در نظر بگیرید:

$$M_1 = 0^{\frac{n}{4}}||1\,0^{\frac{n}{4}-1} \quad \Rightarrow \quad T_1 = \langle t_1, t_2, t_3 \rangle$$

$$\langle M, T \rangle : \quad M = 0^{\frac{n}{4}} \quad \Rightarrow \quad T = \langle t_1, t_2 \rangle$$

در این حالت نیز اگر برچسب خروجی الگوریتم $\mathsf{Mac}_k$ به صورت $t_1 \oplus t_2 \oplus \cdots \oplus t_d$ تعریف شود، باز هم حمله کار می‌کند. پرسمان‌های $M_1$، $M_2$ و $M_3$، و برچسب جعلی $T$ برای پیام $M$ را به صورت زیر در نظر بگیرید:

$$\begin{array}{lll} M_1 = 0^{\frac{n}{4}}||1\,0^{\frac{n}{4}-1} & \Rightarrow & T_1 = t_1 \oplus t_2 \oplus t_3 \\ M_2 = 0^{\frac{n}{4}} & \Rightarrow & T_2 = t_1 \oplus t_2 \\ M_3 = 1^{\frac{n}{4}} & \Rightarrow & T_3 = t'_1 \oplus t_2 \end{array}$$

$$\langle M, T \rangle : \quad M = 1^{\frac{n}{4}}||1\,0^{\frac{n}{4}-1} \quad \Rightarrow \quad T = T_1 \oplus T_2 \oplus T_3$$

به‌وضوح $T$ یک برچسب معتبر برای پیام $M$ است.

$$T = T_1 \oplus T_2 \oplus T_3 = t'^1 \oplus t^2 \oplus t^3$$

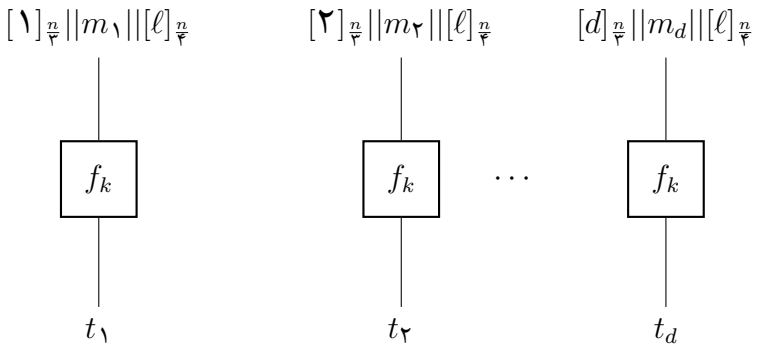
**تلاش سوم.** مشکل روش فوق نیز با افزودن طول پیام اصلی به ورودی تابع شبه‌تصادفی حل می‌شود:

$$|m| = \ell$$

$$\mathrm{pad}(M) = M||1\,0\cdots 0 = m_1 m_2 \cdots m_d, |m_i| = \frac{n}{3}$$

$$t_i = f_k([i]_{\frac{n}{4}}||m_i||[\ell]_{\frac{n}{4}})$$

$$\langle t_1, \cdots, t_d \rangle \leftarrow \mathsf{Mac}_k(M)$$



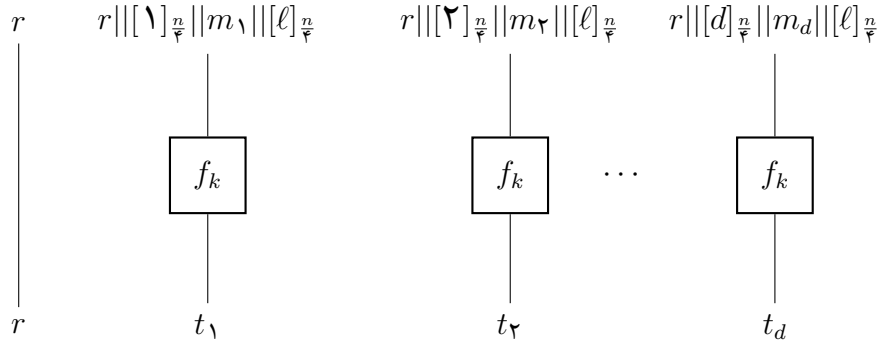

متأسفانه در اینجا نیز با دو پرسمان می‌توان برچسب یک پیام دیگر را جعل کرد. کافیست دو پیام را که فقط در دو بلوک اول متفاوت‌اند در نظر بگیریم:

$$\begin{aligned} M_1 = m_1 m_2 m_3 \cdots m_d &\Rightarrow T_1 = \langle t_1, t_2, t_3 \cdots t_d \rangle \\ M_2 = m'_1 m'_2 m_3 \cdots m_d &\Rightarrow T_2 = \langle t'_1, t'_2, t_3 \cdots t_d \rangle \end{aligned}$$

$$\langle M, T \rangle : \quad M = m'_1 m_2 m_3 \cdots m_d \Rightarrow T = \langle t'_1, t_2, t_3 \cdots t_d \rangle$$

**تلاش چهارم.** در نهایت با اضافه کردن یک رشته‌ی تصادفی به تمام بلوک‌ها، می‌توان یک سیستم کد اصالت‌سنجی پیام با امنیت جعل‌ناپذیری ارائه کرد:

$$|m| = \ell$$
$$\mathrm{pad}(M) = M||10\cdots0 = m_1 m_2 \cdots m_d, |m_i| = \frac{n}{4}$$
$$r \leftarrow \{0,1\}^{\frac{n}{4}}$$
$$t_i = f_k(r||[i]_{\frac{n}{4}}||m_i||[\ell]_{\frac{n}{4}})$$
$$\langle t_1, \cdots, t_d \rangle \leftarrow \mathsf{Mac}_k(M)$$




می‌توان ثابت کرد که طرح فوق یک کد اصالت‌سنجی دارای امنیت جعل‌ناپذیری است.
بدین ترتیب توانستیم برای پیام‌های به طول متغیر، یک سیستم کد اصالت‌سنجی پیام با امنیت جعل‌ناپذیری بسازیم. اما در این روش طول برچسب حاصل شده بسیار زیاد و چند برابر طول پیام اصلی است؛ این مساله باعث می‌شود که

عملا این سیستم برای دریافت و ارسال پیام ناکارآمد باشد. راه حلی که ممکن است به ذهن برسد تعریف برچسب به صورت $\langle r, t_1 \oplus t_2 \oplus \cdots \oplus t_d \rangle \leftarrow \mathsf{Mac}_k(m)$ است. آیا به نظر شما این طرح جدید امن است؟ حتی در صورت مثبت بودن جواب، طرح بازهم ناکارآمد است زیر مستلزم استفاده از حدود $\frac{۴|m|}{n}$ بار استفاده از تابع شبه‌تصادفی است. هدف ما دستیابی به طرح‌های امنی است که حدود $\frac{|m|}{n}$ بار تابع شبه‌تصادفی را فراخوانی می‌کند. یک روش مرسوم طراحی MACهای امن استفاده از توابع چکیده‌ساز[۱۸] است که در جلسات آینده در مورد آنها بحث خواهیم کرد.

[۱۸] hash functions