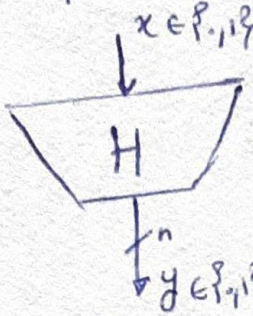
## ① توابع چکیده ساز (Hash Functions):

گوییم $H: \{0,1\}^* \to \{0,1\}^n$ یک تابع چکیده ساز است هرگاه هر پیام ورودی با طول دلخواه را به یک خروجی با طول ثابت (چکیده‌ی پیام) به طور کارایی تصویر کند.




برخی از کاربردهای توابع چکیده ساز عبارتند از:

۱ در تولید اعداد شبه تصادفی،

۲ در پروتکل های احراز هویت،

۳ در احراز یکپارچگی (جامعیت) پیام،

۴ در تولید اعداد شبه تصادفی،

۵ در ساختار ذخیره و بازیابی جداول درهم ساز (Hash Table)،

۶ در پروتکل های تعهد،

۷ بروز رسانی پایگاه داده مربوط به آنتی ویروس ها.

با این حال، تابع چکیده ساز مورد نظر ما در حوزه رمزنگاری باید برخوردی باشد که از روی خروجی آن نتوان ورودی را محاسبه کرد یا از روی ورودی، مقدار خروجی را کنترل کرد.

به طور کلی، از یک تابع چکیده ساز ایده آل انتظار می رود که ویژگی های زیر را داشته باشد:

۱ برخورد تابی (collision Resistant): بدین معناست که برای یک تابع درهم ساز H داده شده، پیدا کردن دو پیام متمایز $m$ و $m'$ که چکیده یکسانی داشته باشند، سخت باشد. به عبارت دیگر از نظر محاسباتی نتوان پیام های $m \neq m'$ را یافت که برای آنها $H(m) = H(m')$ باشد.

به این ویژگی برخورد تاب قوی (strong collision Resistant) نیز اطلاق می‌شود.

۲) پیش تصویر تابی (Preimage Resistant): بدین معناست که برازای y تصادفی داده شده در برد تابع H، پیدا کردن پیام m ای که برای آن داشته باشیم $H(m)=y$، سخت باشد.
به این ویژگی، خاصیت یک طرفه (one way) نیز اطلاق می‌شود.

۳) پیش تصویر دوم تابی (Second Preimage Resistant): بدین معناست که برای یک پیام m داده شده، پیدا کردن پیام $m' \neq m$ که $H(m)=H(m')$ باشد، سخت باشد.
به این ویژگی برخورد تاب ضعیف (Weak Collision Resistant) نیز اطلاق می‌شود.

نکته: مفاهیم بالا را می‌توان به صورت یک بازی (Game) بین مهاجم (Adversary) و چالشگر (challenger) نیز تعریف کرد. به عنوان تمرین این کار را انجام دهید.

## ② دلالت‌ها و جداسازی‌های میان مفاهیم فوق

در این بخش، سعی بر آن است تا ارتباط بین ویژگی‌های مطرح برای توابع درهم‌ساز را بررسی و آن‌ها را برحسب ویژگی‌هایشان کلاس‌بندی کنیم. برای این منظور اختصارهای زیر را برای ویژگی‌های مذکور در نظر می‌گیریم:

۱) Collision Resistant = CR

۲) Preimage Resistant = PIR

۳) Second Preimage Resistant = SPIR

دلالت‌ها (Implications):

۱) اگر تابع H ویژگی CR را داشته باشد آنگاه تابع H ویژگی SPIR را نیز خواهد داشت.

۲) اگر تابع H ویژگی CR را داشته باشد آنگاه تابع H ویژگی PIR را نیز خواهد داشت.

۳ اگر تابع H دیترمینر SPIR را داشته باشد آنگاه تابع H دیترمینر PIR را نیز خواهد داشت.

تمرین: هریک از گزاره های بالا را اثبات کنید.

جداسازی ها (separations):

۱ اگر تابع H دیترمینر PIR را داشته باشد، لزوماً اینطور نیست که دیترمینر CR را نیز داشته باشد

۲ اگر تابع H دیترمینر SPIR را داشته باشد، لزوماً اینطور نیست که دیترمینر CR را نیز داشته باشد

۳ اگر تابع H دیترمینر PIR را داشته باشد، لزوماً اینطور نیست که دیترمینر SPIR را نیز داشته باشد

اثبات ۱ و ۳: فرض کنید تابع H دیترمینر CR را دارد (طبق دلالت های بین H و دیترمینر SPIR, PIR را نیز داریم). حال تابع H' را به صورت زیری سازیم اگر H' یک بیت اضافه تر از H در ورودی می گیرد:

$$H'(\underbrace{ab\,x_2\cdots x_n}_{\text{رشته } n+1 \text{ بیتی}}) = H(\underbrace{a\oplus b\,x_2\cdots x_n}_{\text{رشته } n \text{ بیتی}})$$

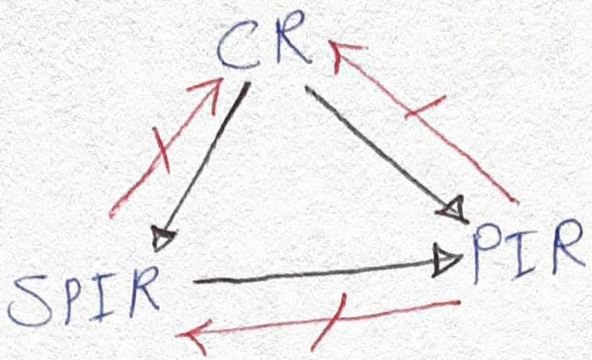
واضح است که همچنان PIR بر تابع H' حفظ می شود اما به وضوح دیترمینر های SPIR و دیترمینر CR برقرار نیست.

اثبات ۲: فرض کنید تابع H دیترمینر CR را دارد (پس H دیترمینر PIR و SPIR را نیز دارا است). حال تابع H' را به صورت زیری سازیم:

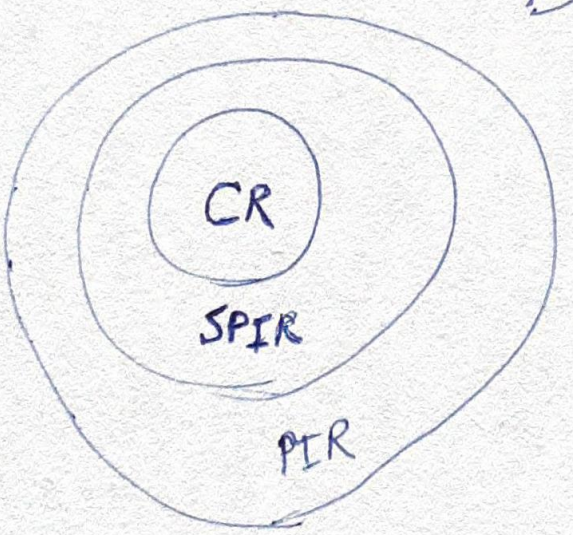
$$H'(x) = \begin{cases} 0^n & \text{if } (x=m \text{ or } x=m') \text{ //where } m\neq m' \in \{0,1\}^* \\ H(x) & \text{O.W} \end{cases}$$

واضح است که همچنان دیترمینر SPIR بر تابع H' حفظ می شود اما به وضوح دیترمینر CR را به سبب عدم کد کردن $m, m'$ ندارد.

جمع بندی روابط: به طور خلاصه می توان روابط بین و تئوری ها را به صورت زیر نشان داد:




کلاس بندی توابع درهم ساز برحسب و تئوری ها: برحسب روابط فوق، می توان توابع درهم ساز را به صورت زیر دسته بندی کرد:




درجه سختی حل مساله و تئوری های فوق:
با توجه به روابط بررسی شده، داریم:

۱ سختی حل مساله SPIR حداقل برسختی حل مساله CR است.

۲ سختی حل مساله PIR حداقل برسختی حل مساله SPIR است.

بنابراین درجه سختی بین سه تئوری به صورت زیر است:

$$PIR > SPIR > CR$$

## فرض قوی / ضعیف:

همانطور که در بخش های قبلی دیدیم، حل مسایل مربوط به ویژگی ها از درجه سختی متفاوتی برخوردار بوده. در حالت کلی اگر امنیت را بر پایه مساله با درجه سختی پایین تری نسبت به دیگر مسایل قرار دهیم، فرض قوی تری را در نظر گرفته ایم به این معنا که چنین مساله ای حل نمی شود. بنابراین برای ویژگی های فوق به ترتیب فرض ضعیف به قوی زیر را می توان در نظر گرفت:

۱ـ مساله PIR،

۲ـ مساله SPIR،

۳ـ مساله CR.

فرض قوی ← PIR > SPIR > CR → فرض ضعیف

نکته: درجه سختی بیشتر ← فرض ضعیف تر

(۳) امنیت توابع درهم ساز

حمله جستجوی کامل (Brute-force attack) به تمامی توابع چکیده ساز قابل اعمال است.

امنیت تابع چکیده ساز در مقابل حمله جست و جوی کامل متناسب با طول خروجی آن است. بنابراین طول خروجی باید به اندازه کافی بزرگ باشد تا تابع چکیده ساز در مقابل حملات زیر امن باشد. فرض کنید H یک تابع چکیده ساز ایده آل با n بیت خروجی باشد.

حمله برخورد: با توجه به تناقض روز تولد، برای آنکه مهاجم بتواند با احتمال بیش از $\frac{1}{2}$ پیام های $m \neq m'$ که $H(m)=H(m')$ پیدا کند، نیاز به حداکثر $2^{n/2}$ بار محاسبه تابع H دارد.

حمله پیش تصویر: در بدترین حالت مهاجم نیاز به حداکثر $2^n$ بار محاسبه تابع H دارد.

حمله پیش تصویر دوم: در بدترین حالت مهاجم نیاز به محاسبه $2^n$ بار محاسبه تابع H دارد.

گوییم یک تابع چکیده ساز شکسته شده است اگر الگوریتمی با پیچیدگی کمتر از حداقل حداکثر حمله ای که برای تابع چکیده ساز ایده آل را تحقق کند.