06/05/2022

Hash functions and Message Authentication Code

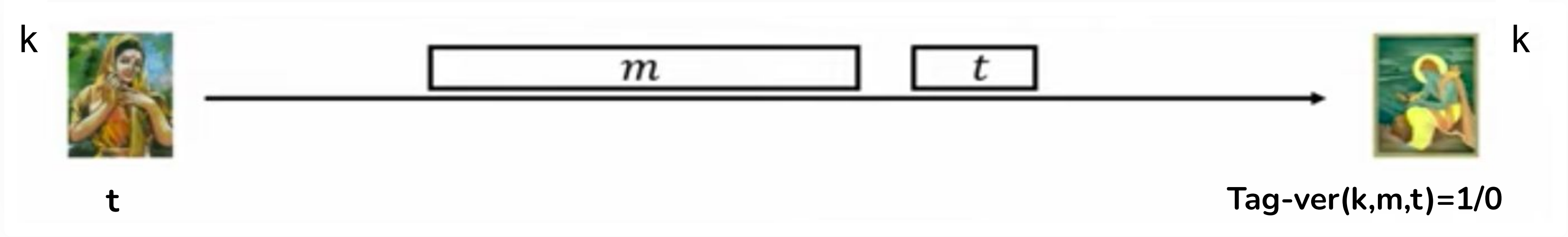
Mentor: Prof. Maheshanand

Samiksha Jain

MSM 3rd year

Enrollment no. 19312028

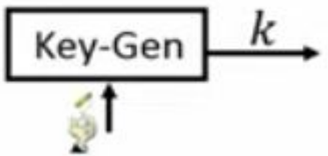
Message Integrity and Authenticity in Symmetric Key Setting



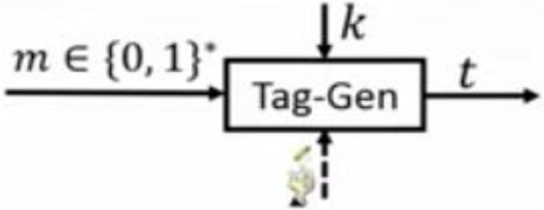
- **How can receiver verify whether the received message was indeed sent by a designated sender? —Message Authenticity**
- **How can receiver verify whether the received message was changed enroute?--- Message Integrity.**
- **Possible Solution: along with the message send a short verification tag.**
- **Message is accepted only if tag verification is successful.**

Message Authentication Code(MAC)

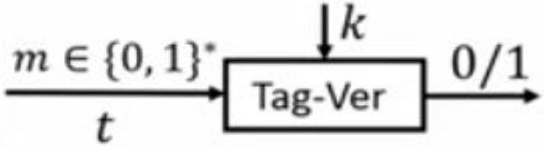
A Mac Π is a collection of three algorithms(key -gen, Tag- gen, Tag-ver).



- ❑ Key-Gen(1^n) $\to k \in$ K
- ❑ K: set of all output keys.
- ❑ Randomized Algorithm



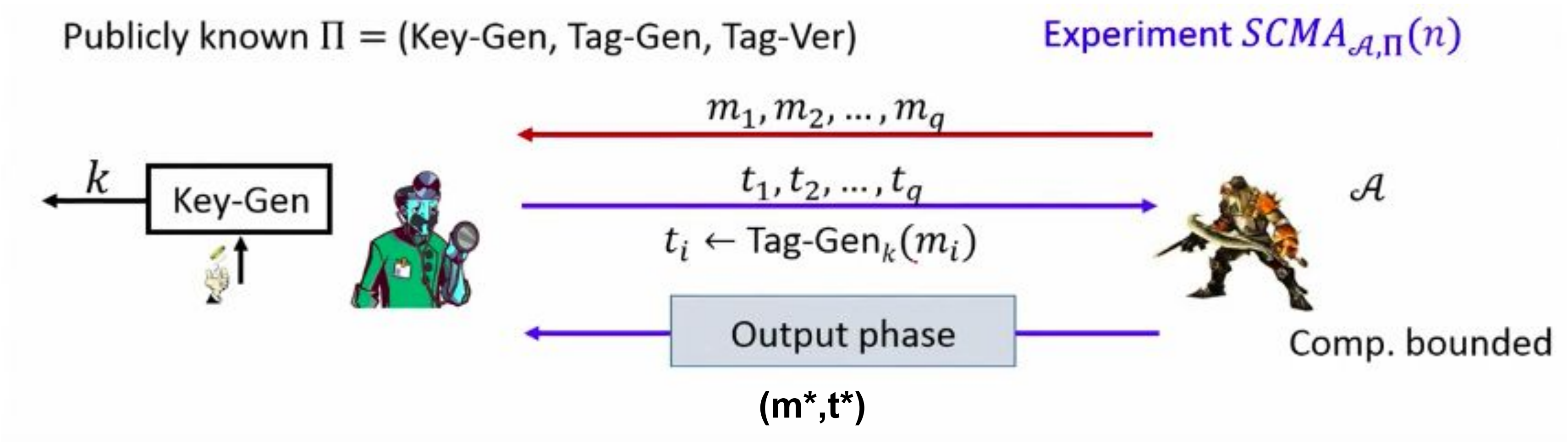
- ❑ Tag-Gen$_k$(m) -> t $\in$ T
- ❑ M: set of all possible messages.
- ❑ T: set of all possible output tags.
- ❑ Need not be randomized



- ❑ Tag-Verk(m,t) $\in$ {0,1}
- ❑ Deterministic Algorithm

❑ Correctness: for every $k \in$ K and m $\in$ M, the following should hold:

Tag-Ver$_k$(m, Tag-Gen$_k$(m)) = 1



A is said to win the experiment if:

$(m^*,t^*) \notin (m_1,t_1),(m_2,t_2)......(m_q,tq)$ and $\text{Tag-Ver}_k(m^*,t^*)=1$

Π is said to be (SCMA) strong chosen message Attack secure, if for every ppt A

$P[\text{A wins the experiment SCMA}_{A,\Pi}(n)] < negl(n)$

Secure MAC for Fixed-length messages from PRF.

- Let $F: \{0,1\}^n \times \{0,1\}^\ell \Rightarrow \{0,1\}^L$ be a secure PRF

$k \in_R \{0,1\}^n$

$x \in \{0,1\}^\ell$

$y = F(k,x) \in \{0,1\}^L$

- Using F, we construct a **deterministic MAC** with:
  - $\mathcal{K} = \{0,1\}^n$
  - $\mathcal{M} = \{0,1\}^\ell$
  - $\mathcal{T} = \{0,1\}^L$

Key-Gen $\xrightarrow{k \in_R \mathcal{K}}$

$m \in \mathcal{M} \rightarrow$ Tag-Gen (key k) $\xrightarrow{t \in \mathcal{T}}$

$t := F_k(m)$

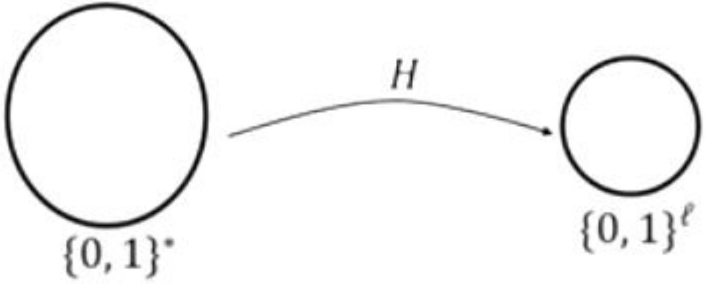
$(m,t) \rightarrow$ Tag-Ver (key k) $\rightarrow$ 1, iff $t = F_k(m)$

Cryptographic Hash functions

Tremendous application, both in symmetric key and public key world.

- **Primary Application :** Data Compression
- **Other applications: MAC, Key-derivation function, de-duplication, etc**

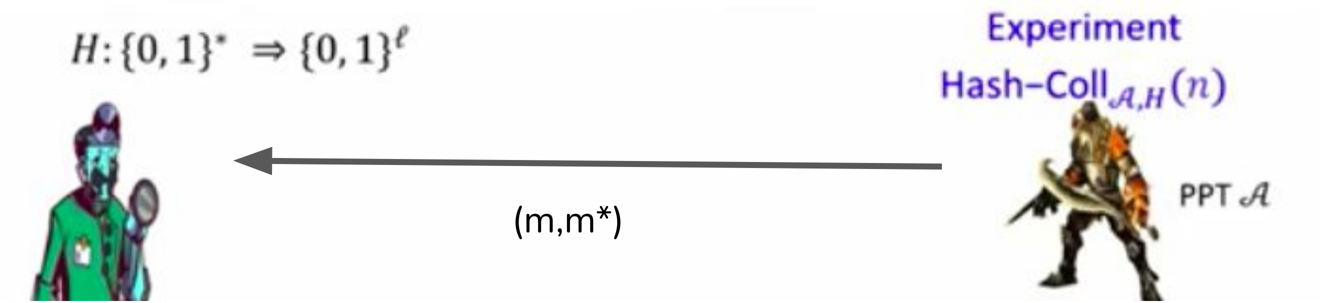
A Many to one function mapping arbitrary length bit strings to fixed length bit strings.



Main security property- Collision Resistant

- **Given a description of H, finding collisions for H must be computationally difficult.**

Collision Resistant Hash Function



- ❑ **H is a CRHF, if for every ppt A in Hash-CollA,H(n) their exists a negligible function negl(n):**
  P[A outputs m,m*: m≠m* and H(m) = H(m*)]<= negl(n)

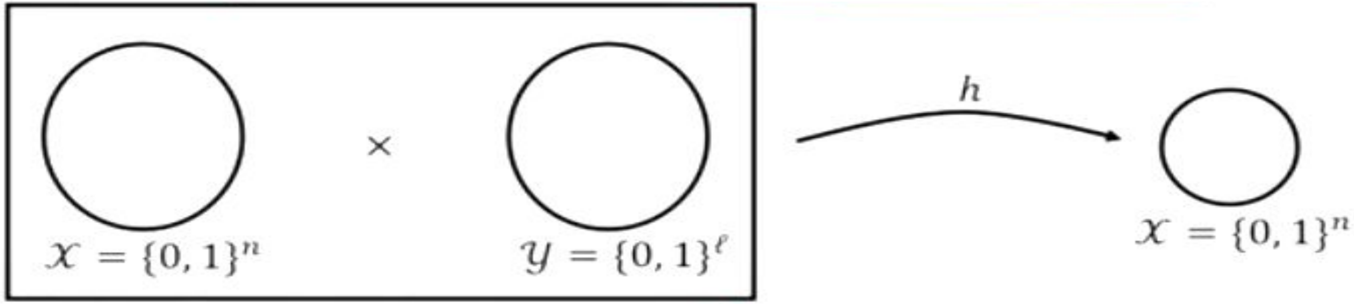
A technical issue with the above definition

- ❑ **Since |{0,1}*|>>{0,1}ℓ , collision must exist(pigeon-hole principle).**
- ❑ **There always exist a constant time A_{coll} hardcoded with a colliding pair(m,m*)**

Merkle Damgård Paradigm for design of CRHF

A well known two- stage approach for designing a CRHF (used in MD5, SHA-256).

- **Stage 1: Design a fixed length, collision resistant , compression function.**



h: $\{0,1\}^{n+\ell} \to \{0,1\}^n$

- **Stage 2: Design a CRHF H_{MD} for arbitrary length messages, using h as a black box.**

  *** Constructing a CRHF H_{MD} : $\{0,1\}^{\leq L}$ -> X, from h: X*Y-> X**

Constructing CRHF HMD :{0,1}<L ->X, from collision resistant h: X x Y->Y

$X=\{0,1\}^n$ $Y=\{0,1\}^\ell$

- For SHA256, n=256 and ℓ = 512

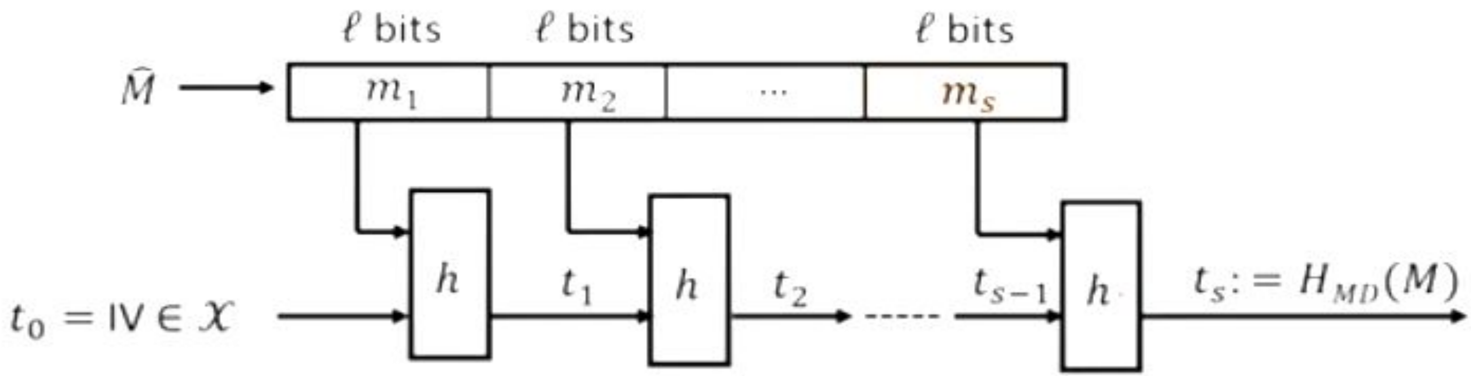
- Step 1: Encode the input $M \in \{0,1\}^{<L}$ for H_{MD}, to make encoded M as a multiple of ℓ bits.

$$M \in \{0,1\}^{\leq L} \xrightarrow{\text{Encode}} \hat{M} = M \,||PB \in \{\{0,1\}^\ell\}^{\leq \frac{L}{\ell}+1}$$

❖ $PB \stackrel{\text{def}}{=} 1000 \ldots 00||\langle s \rangle$, where $\langle s \rangle$ is a **fixed-length bit-string**, representing the **number of ℓ-bit blocks in M**

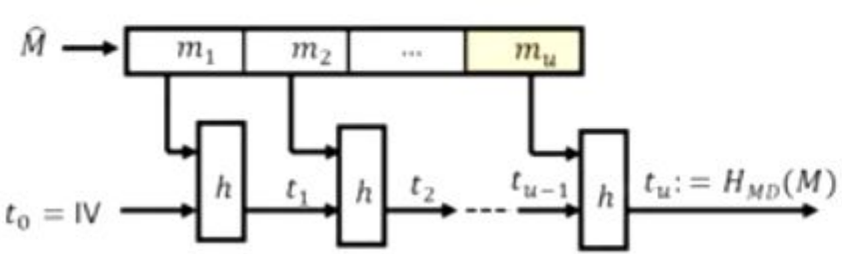
- Typically a 64-bit field -> $L \lesssim 2^{64}.\ell$ bits.
- If L is already a multiple of ℓ, then an additional dummy block added for PB.

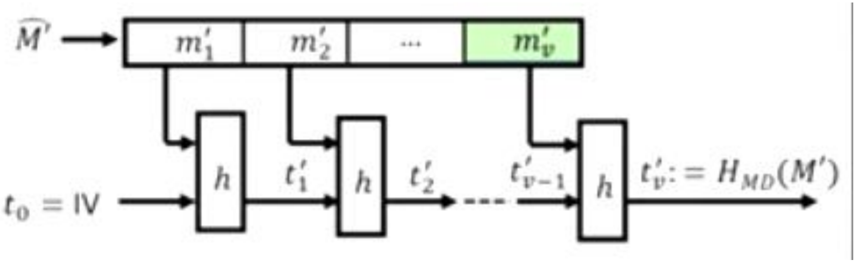
Step 2:Apply function H iteratively over the block of M and the previous outcome of h.



- IV : fixed, publicly known value, (say 0^n), some complicated string.
- Variable $t_0, t_1, \ldots, t_s \in X$ chaining variables.

- If h: X x Y-> X is a collision resistant function, then H_{MD}:{0,1}<L ->X is a CRHF
- Let there exist a ppt A_{MD} which output distinct M, M' ∈{0,1}<L such that P[H_{MD}(M)=H_{MD}(M')] = f(n), where f(n) is a non negligible function.

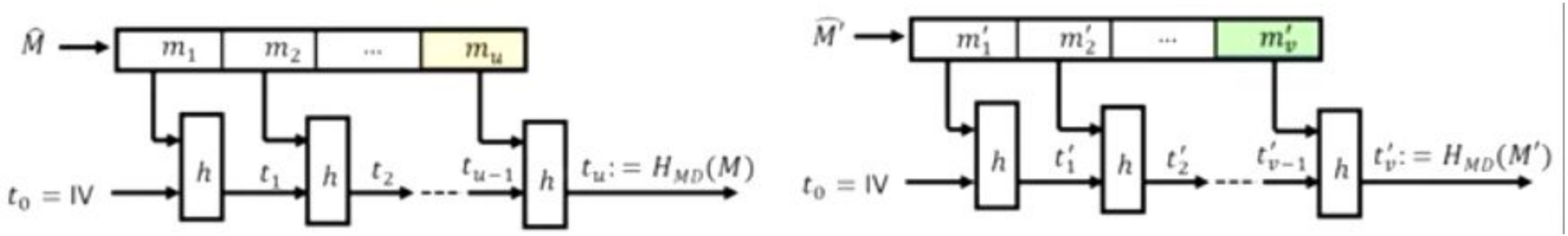




- ❑ Using A_{MD}, we construct a PPT A_h which outputs distinct pairs (t,m)(t*,m*)∈ X x Y:

P[h(t,m)=h(t*,m*)]= f(n)

- ❑ To find collision (t,m) (t*,m*) for h, A_h parses the hash chain H_{MD} and H_{MD}(M') from right to left.



- Since $H_{MD}(M) = H_{MD}(M') \Rightarrow h(t_{u-1}, m_u) = h(t'_{v-1}, m'_v)$

- If $(t_{u-1}, m_u) \neq (t'_{v-1}, m'_v)$, then the pair constitutes a collision for h

- **Else (tu-1, mu) = (t’v-1,m’v): M and M’ contains the same number of blocks–u=v.**
- **Consider the second-to- last invocation of h in the hash-chains.**

❑ $(t_{u-1}, m_u) = (t'_{u-1}, m'_u) : M \neq M'$, but **contains the same number of blocks**

Second-to-last invocation of h in the hash-chain $H_{MD}(M)$ and $H_{MD}(M')$

- ❖ If $(t_{u-2}, m_{u-1}) \neq (t'_{u-2}, m'_{u-1})$, then the pair constitutes a collision for h, as $t'_{u-1} = t_{u-1}$
- ❖ Else $(t_{u-2}, m_{u-1}) = (t'_{u-2}, m'_{u-1})$, with $m_u = m'_u$

➢ Consider the **third-to-last invocation** of h in the hash-chains

❑ The above process of scanning from right to left **will eventually find** an h-collision

➢ Else, we conclude that all the blocks of distinct M, M' are same --- a contradiction

Message Authentication using Hash functions

Mac for arbitrary long messages using a CRHF (Hash-and-Mac paradigm)

- Given an **arbitrary-length message**, compute its **fixed-length Mac-tag** in **two stages**:
  - **Step I: Hash** the arbitrary-length message to a **fixed-length string** using a CRHF
  - **Step II:** Compute the **Mac-tag on the message digest** (output of the CRHF)

Π_{MAC} = (Key-Gen, Tag-Gen, Tag-Ver) with key-space $\mathcal{K}$, message-space $\mathcal{M} = \{0,1\}^{\ell}$ and tag-space $\mathcal{T}$

+

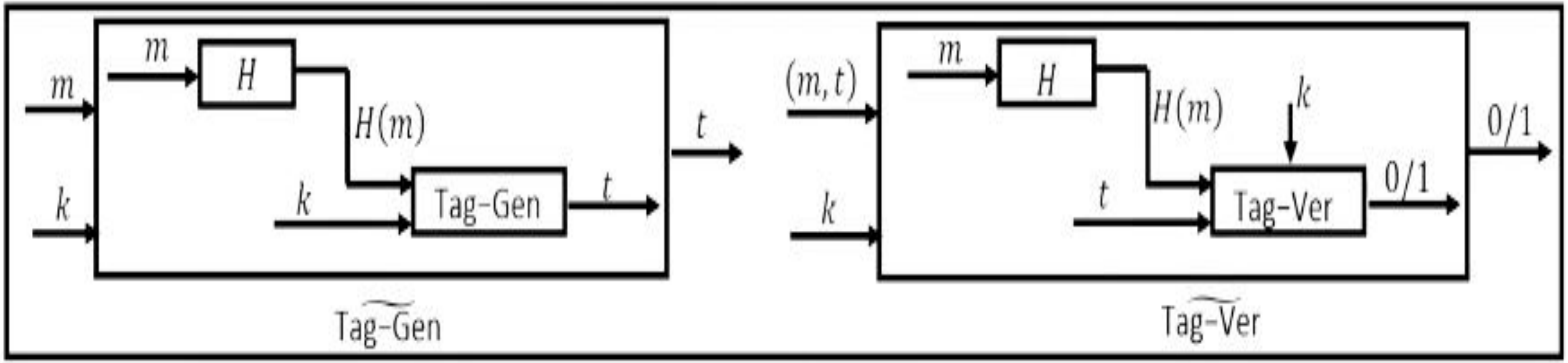
A CRHF $H: \{0,1\}^* \Rightarrow \{0,1\}^{\ell}$

⇒

$\widetilde{\Pi}_{MAC}$ = ($\widetilde{\text{Key-Gen}}$, $\widetilde{\text{Tag-Gen}}$, $\widetilde{\text{Tag-Ver}}$) with key-space $\mathcal{K}$, message-space $\mathcal{M} = \{0,1\}^*$ and tag-space $\mathcal{T}$

Π_{MAC} = (Key-Gen, Tag-Gen, Tag-Ver) with key-space $\mathcal{K}$, message-space $\mathcal{M} = \{0,1\}^{\ell}$ and tag-space $\mathcal{T}$

A CRHF $H: \{0,1\}^{*} \Rightarrow \{0,1\}^{\ell}$



What will I be studying in future?

- Birthday Attacks on cryptographic Hash functions.
- Many more applications of Hash functions
- Random oracle model and authentication Encryption.
- Security analysis of various Hash functions.
- MAC for arbitrary long messages.
- MAC for long messages using CRHF.
- And your suggestions are welcomed.

References

1. Foundations of Cryptography by Prof Ashish Chaudhary.
   Computer science and Engineering.(IIIT Bangalore)

2. Cryptography Theory and Practice (4th edition) by
   Douglas R.Stinson and Maura B. Paterson
   CRC press(Taylor and Francis Group)
   A Champman and Hall book

THANK YOU