Criptografie și Securitate

- Prelegerea 15 -
HMAC

Adela Georgescu, Ruxandra F. Olimid

Facultatea de Matematică și Informatică
Universitatea din București

Cuprins

Revizuire: MAC-uri

- În prelegerile anterioare am vazut că putem construi MAC-uri sigure pe baza funcțiilor pseudoaleatoare (PRF);

Revizuire: MAC-uri

- În prelegerile anterioare am vazut că putem construi MAC-uri sigure pe baza funcțiilor pseudoaleatoare (PRF);
- Pentru $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ o PRF, defineam un MAC astfel
  - $\mathrm{Mac}(k, m) : t = F_k(m)$;
  - $\mathrm{Vrfy}(k, m, t) = 1$ dacă și numai dacă $t = F_k(m)$ (altfel întoarce 0).

Revizuire: MAC-uri

- În prelegerile anterioare am vazut că putem construi MAC-uri sigure pe baza funcțiilor pseudoaleatoare (PRF);
- Pentru $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ o PRF, defineam un MAC astfel
  - $\mathrm{Mac}(k, m) : t = F_k(m)$;
  - $\mathrm{Vrfy}(k, m, t) = 1$ dacă și numai dacă $t = F_k(m)$ (altfel întoarce 0).
- Această construcție este bună pentru mesaje de lungime mică, dar avem nevoie de construcții de MAC-uri pentru mesaje mult mai mari;

Revizuire: MAC-uri

- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară

Revizuire: MAC-uri

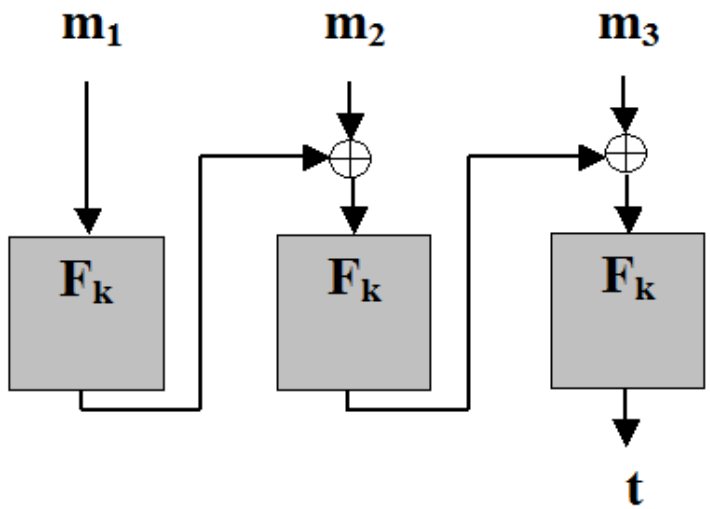
- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară
  - HMAC - pentru protocoale pe Internet: SSL, IPsec, SSH...

Revizuire: MAC-uri

- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară
  - HMAC - pentru protocoale pe Internet: SSL, IPsec, SSH...
- Reamintim construcția CBC-MAC sigură pentru mesaje de lungime fixă:

Revizuire: MAC-uri

- Există două construcții de bază care se folosesc în practică:
  - CBC-MAC - folosit pe larg în industria bancară
  - HMAC - pentru protocoale pe Internet: SSL, IPsec, SSH...
- Reamintim construcția CBC-MAC sigură pentru mesaje de lungime fixă:

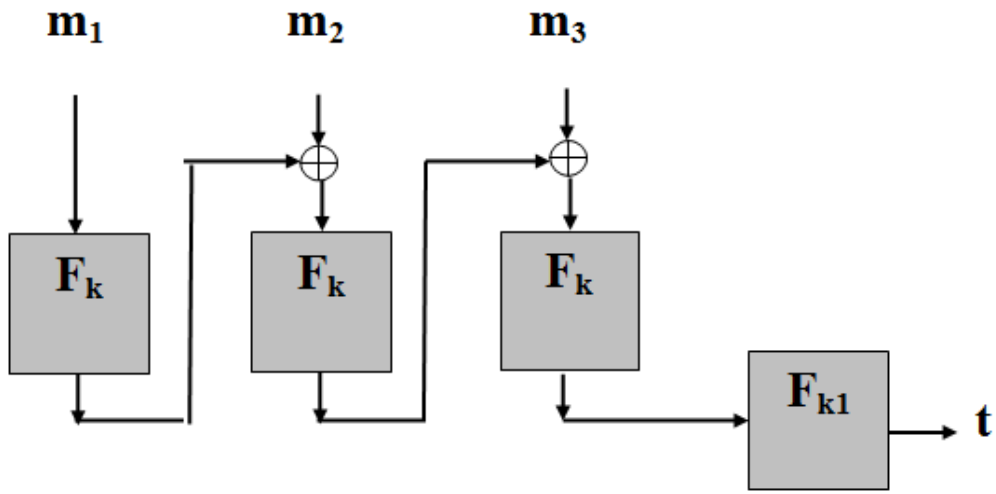


Revizuire: MAC-uri

- Însă pentru mesaje de lungime variabilă, putem reține această construcție ca fiind sigură:

Revizuire: MAC-uri

- Însă pentru mesaje de lungime variabilă, putem reține această construcție ca fiind sigură:



HMAC

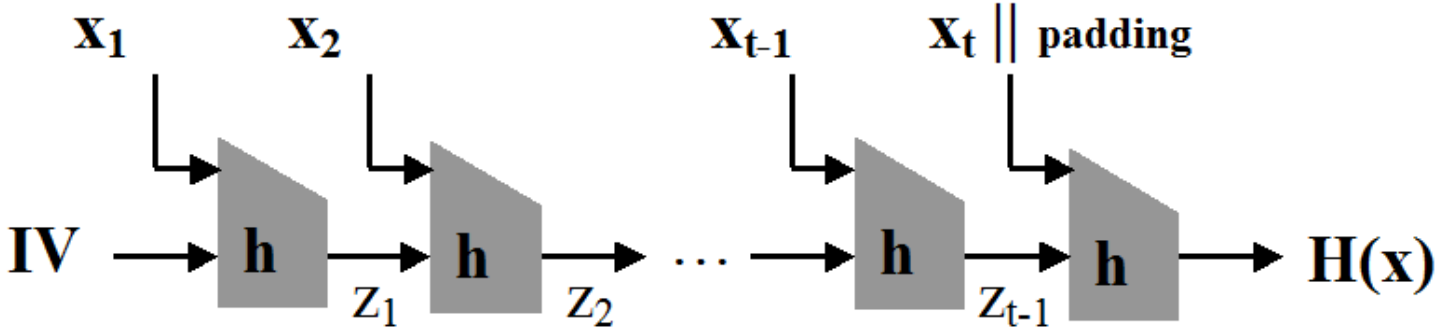
- Am studiat deja funcții hash și transformarea Merkle-Damgård pentru a obține funcții hash (rezistente la coliziuni) cu intrarea de lungime variabilă pornind de la funcții de compresie cu intrarea de lungime fixă;

HMAC

- Am studiat deja funcții hash și transformarea Merkle-Damgård pentru a obține funcții hash (rezistente la coliziuni) cu intrarea de lungime variabilă pornind de la funcții de compresie cu intrarea de lungime fixă;
- Reamintim construcția Merkle-Damgård:

HMAC

- Am studiat deja funcții hash și transformarea Merkle-Damgård pentru a obține funcții hash (rezistente la coliziuni) cu intrarea de lungime variabilă pornind de la funcții de compresie cu intrarea de lungime fixă;
- Reamintim construcția Merkle-Damgård:



HMAC

- Încercăm să construim un MAC direct pornind de la H(·), unde $H(\cdot)$ este funcția hash obținută cu transformarea Merkle-Damgård;

HMAC

- Încercăm să construim un MAC direct pornind de la H(·), unde $H(\cdot)$ este funcția hash obținută cu transformarea Merkle-Damgård;
- Definim $\mathrm{Mac}(k, m)$ astfel: $t = H(k||m)$.

HMAC

- Încercăm să construim un MAC direct pornind de la H(·), unde $H(\cdot)$ este funcția hash obținută cu transformarea Merkle-Damgård;
- Definim $\mathrm{Mac}(k, m)$ astfel: $t = H(k||m)$.
- Întrebare: Este acesta un MAC sigur (nu poate fi falsificat printr-un atac cu mesaj clar ales)?

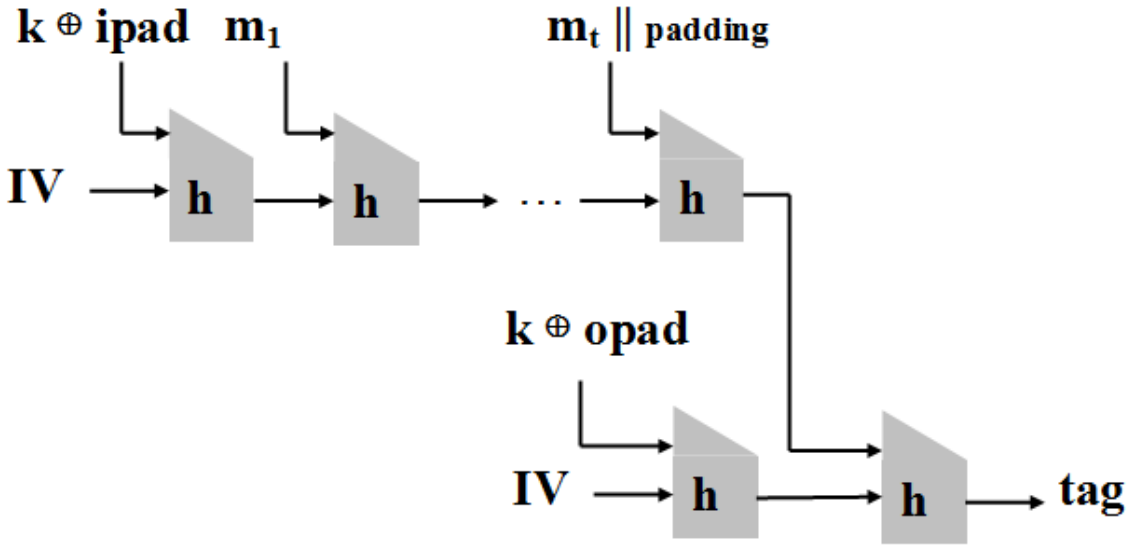
HMAC

- Încercăm să construim un MAC direct pornind de la H(·), unde $H(\cdot)$ este funcția hash obținută cu transformarea Merkle-Damgård;
- Definim $\mathrm{Mac}(k, m)$ astfel: $t = H(k||m)$.
- Întrebare: Este acesta un MAC sigur (nu poate fi falsificat printr-un atac cu mesaj clar ales)?
- Răspuns: NU! Un advesar poate calcula un tag t' pentru un mesaj nou care nu a mai fost autentificat: extinde mesajul anterior cu încă un bloc d, și calculează:

$$H(k||m||d) = h(d||H(k||m))$$

HMAC

- Folosim metoda standardizată HMAC (Hash MAC):



- HMAC se definește astfel:
- $\mathrm{Mac}(k, m)$: $t = H((k \oplus \mathtt{opad}) \,||\, H((k \oplus \mathtt{ipad}) \,||\, m))$;
- $\mathrm{Vrfy}(k, m, t) = 1 \iff t = \mathrm{Mac}(k, m)$.

Notații

- `ipad` și `opad` sunt două constante de lungimea unui bloc m_i
- `ipad` constă din byte-ul $0x5C$ repetat de atâtea ori cât e nevoie;
- `opad` constă din byte-ul $0x36$ repetat de atâtea ori cât e nevoie;
- IV este o constantă fixată.

Securitate HMAC

- Definim $G(k) = h(IV \,||\, (k \oplus \text{opad})) \,||h(IV \,||\, (k \oplus \text{ipad}))$

Securitate HMAC

- Definim $G(k) = h(IV \,||\, (k \oplus \text{opad}))\, ||h(IV \,||\, (k \oplus \text{ipad}))$
- Privind secvența ca $G(k) = k_1||k_2$, dacă G este PRG și $k \leftarrow^R \{0,1\}^n$, deși k_1, k_2 sunt dependente, acestea par alese în mod uniform și independent;

Securitate HMAC

- Definim $G(k) = h(IV \,||\, (k \oplus \text{opad}))\, ||h(IV \,||\, (k \oplus \text{ipad}))$
- Privind secvența ca $G(k) = k_1||k_2$, dacă G este PRG și $k \leftarrow^R \{0,1\}^n$, deși k_1, k_2 sunt dependente, acestea par alese în mod uniform și independent;
- Dacă G este PRG, atunci dăm următorul rezultat de securitate pentru HMAC :

Securitate HMAC

- Definim $G(k) = h(IV \,||\, (k \oplus \text{opad}))\, ||h(IV \,||\, (k \oplus \text{ipad}))$
- Privind secvența ca $G(k) = k_1||k_2$, dacă G este PRG și $k \leftarrow^R \{0,1\}^n$, deși k_1, k_2 sunt dependente, acestea par alese în mod uniform și independent;
- Dacă G este PRG, atunci dăm următorul rezultat de securitate pentru HMAC :

Teoremă

Dacă G este PRG, h prezintă rezistență la coliziuni și MAC-ul $h(k \,||\, m)$ contruit be baza ei este sigur (pentru mesaje de lungime fixă), atunci HMAC este sigur (pentru mesaje de lungime arbitrară) - nu poate fi falsificat printr-un atac cu mesaj ales.

Important de reținut!

- HMAC este un MAC foarte popular folosit în multe protocoale practice precum TLS;
- Construcția lui se bazează pe funcții hash, de exemplu SHA-2 (SHA-256).