Symmetric-key Corruption Detection : When XOR-MACs Meet Combinatorial Group Testing

Lidia Istrate 342C4

XOR-GTM este o clasa de MAC (Message Authentication Code) ce verifica integritatea mesajului dar gaseste si partea mesajului care a fost corupta. Poate fi vazut de asemenea si ca o aplicare a CGT (Combinatorial Group Testing) pentru message authentication.
Incercarile similare pentru acest model au limitari in comunicatie, dar XOR-GTM are un cost de comunicatie mult mai mic, oferind aceeasi capacitate de detectie a coruptiei.

DirectGTM : alegerea matricii de test se face independent de MACk si o matrice d-disjuncta este sugerata sa fie folosita in loc de H, deci costul de comunicatie poate fi redus doar prin gasirea unei matrici d-disjuncte small-row.
Pentru a imbunatati costul de comunicatie, cand authenticatorul ia taguri MAC bazate pe matricea de test H, verificatorul poate folosi orice submatrice a row span-ului lui H pe post de matrice de test virtuala.

Implementarea va presupune 2 cazuri : XOR-GTM cu matrice de test de dimensiune t x m, functiile de hash, de tag, verificare a integritatii, detectie a partii corupte si XOR-GTM-PPI ce foloseste H^R o matrice circulara cu functiile de hash, tag, detectie a partii corupte.

Pseudocod cand este folosita o matrice H de test de dimensiune t x m:

Algorithm
$\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{tag}(M)$

$S \leftarrow \mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M)$
for $i = 1$ **to** t **do**
  $T[i] \leftarrow G^i_{K'}(S[i])$
$T \leftarrow (T[1], \ldots, T[t])$
return T

Algorithm
$\mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M)$

for $i = 1$ **to** t **do**
  $S[i] \leftarrow 0^n$
for $j = 1$ **to** m **do**
  $Z \leftarrow F^j_K(M[j])$
  for $i = 1$ **to** t **do**
    if $\mathbf{H}_{i,j} = 1$
      then $S[i] \leftarrow S[i] \oplus Z$
$S \leftarrow (S[1], \ldots, S[t])$
return S

Algorithm
$\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{verify}(M', T')$

$\widehat{T} \leftarrow \mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{tag}(M')$
if $\widehat{T} = T'$ **return** $\top$
else return $\perp$

Algorithm
$\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{verify\text{-}S}(M', T')$

for $i = 1$ **to** t **do**
  $S' \leftarrow G^{i,-1}_{K'}(T'[i])$
$\widehat{S} \leftarrow \mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M')$
for $i = 1$ **to** v **do**
  $\widehat{S}^R[i] \leftarrow \bigoplus_{j \in R_i} \widehat{S}[j]$
  $(S')^R[i] \leftarrow \bigoplus_{j \in R_i} S'[j]$
for $i = 1$ **to** v **do**
  if $\widehat{S}^R[i] = (S')^R[i]$ **then** $B[i] \leftarrow \top$
  else $B[i] \leftarrow \perp$
$B \leftarrow (B[1], B[2], \ldots, B[v])$
return B

Algorithm
$\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{detect}(M', T')$
$//R_i = \{i\}$ for $i \in [\![t]\!]$

$\mathcal{P} \leftarrow [\![m]\!]$
for $i = 1$ **to** t **do**
  $S' \leftarrow G^{i,-1}_{K'}(T'[i])$
$\widehat{S} \leftarrow \mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M')$
for $i = 1$ **to** v **do**
  $\widehat{S}^R[i] \leftarrow \bigoplus_{j \in R_i} \widehat{S}[j]$
  $(S')^R[i] \leftarrow \bigoplus_{j \in R_i} S'[j]$
for $i = 1$ **to** v **do**
  if $\widehat{S}^R[i] = (S')^R[i]$
    then $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathbf{H}^R_i$
return $\mathcal{P}$

Fig. 1: XOR-GTM using $t \times m$ test matrix $\mathbf{H}$ and extension rule R with v elements.

Pentru a reduce semnificativ memoria ocupata de matricea de test, XOR-GTM-PPI foloseste H^R o matrice circulara.

Algorithm $\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{tag}(M)$:

$S \leftarrow \mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M)$
for $i = 1$ **to** t **do**
  $T[i] \leftarrow G^i_{K'}(S[i])$
$T \leftarrow (T[1], \ldots, T[t])$
return T

Algorithm $\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{hash}(M)$:
$//\ \mathbf{H}_i = \mathbf{H}^R_i$ for all $i \in [\![t]\!]$
$//\ b_i \in (\!|m|\!),\ i \in [\![w]\!]$: $\mathbf{H}^R_{*,1} = \{b_i + 1 : i \in [\![w]\!]\}$
$//\ w = 2^s + 1$

for $i = 1$ **to** t **do**
  $S[i] \leftarrow 0^n$
for $j = 1$ **to** m **do**
  $Z \leftarrow F^j_K(M[j])$
  $\mathcal{I} \leftarrow \{((b_k + (j-1)) \bmod m) + 1 : k \in [\![w]\!]\}$
  for all $i \in \mathcal{I}$ **do** $S[i] \leftarrow S[i] \oplus Z$
$S \leftarrow S([1], \ldots, S[t])$
return S

Algorithm $\mathsf{XOR\text{-}GTM}[F_K, G_{K'}].\mathrm{detect}(M', T')$:
$//c_i \in (\!|m|\!),\ i \in [\![w]\!]$: $\mathbf{H}_1 = \{c_i + 1 : i \in [\![w]\!]\}$

$\mathcal{P} \leftarrow [\![m]\!]$
for $i = 1$ **to** t **do**
  $S' \leftarrow G^{i,-1}_{K'}(T'[i])$
$\widehat{S} \leftarrow \mathsf{XOR\text{-}GTM}[F_K].\mathrm{hash}(M')$
for $i = 1$ **to** v **do**
  $\widehat{S}^R[i] \leftarrow \bigoplus_{j \in R_i} \widehat{S}[j]$
  $(S')^R[i] \leftarrow \bigoplus_{j \in R_i} S'[j]$
for $i = 1$ **to** v **do**
  if $\widehat{S}^R[i] = (S')^R[i]$
    $\mathcal{S} \leftarrow \{((c_j + i - 1) \bmod m) + 1 : j \in [\![w]\!]\}$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{S}$
return $\mathcal{P}$

Fig. 3: XOR-GTM-PPI: XOR-GTM using projective-plane incidence matrix for $\mathbf{H}^R$.