New MAC Constructions from (Tweakable) Block Ciphers

Benoît Cogliati[1] Jooyoung Lee[2] Yannick Seurin[3]

[1]UL, Luxembourg

[2]KAIST, Korea

[3]ANSSI, France

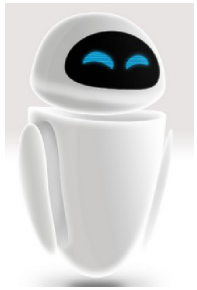
January, 2017 — Early Symmetric Crypto

Outline

Outline

Nonce-Based Message Authentication Codes

(N, M, T)



$T = \mathrm{MAC}_K(N, M)$

$\mathrm{MAC}_K(N, M) = T$?

Security Definition

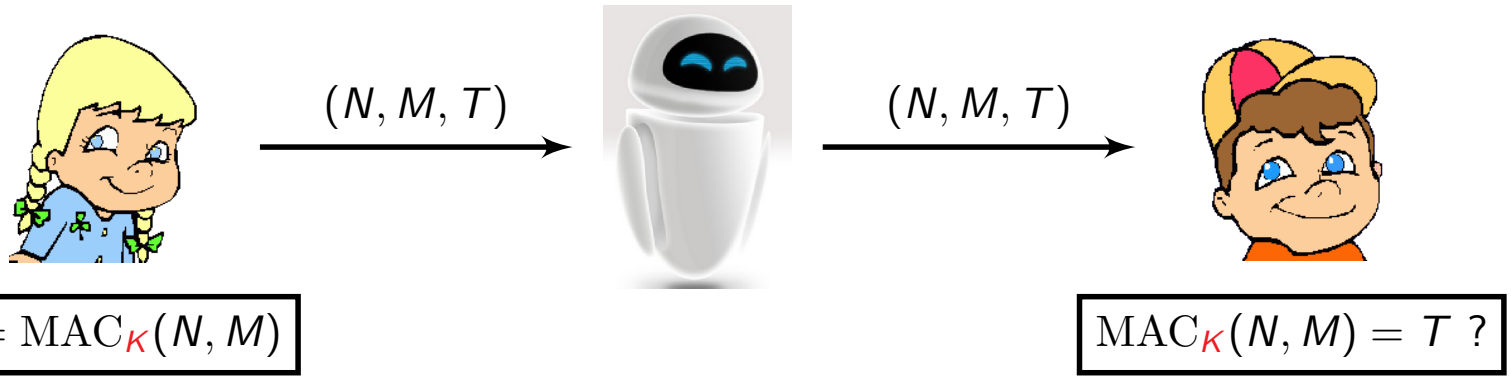
The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(N, M)$
- q_v verification queries (forgery attempts) (N', M', T')

and is successful if one of the verification queries (N', M', T') passes and no previous MAC query (N', M') returned T'.

The adversary is said nonce-respecting if it does not repeat nonces in MAC queries.

Nonce-Based Message Authentication Codes



Security Definition

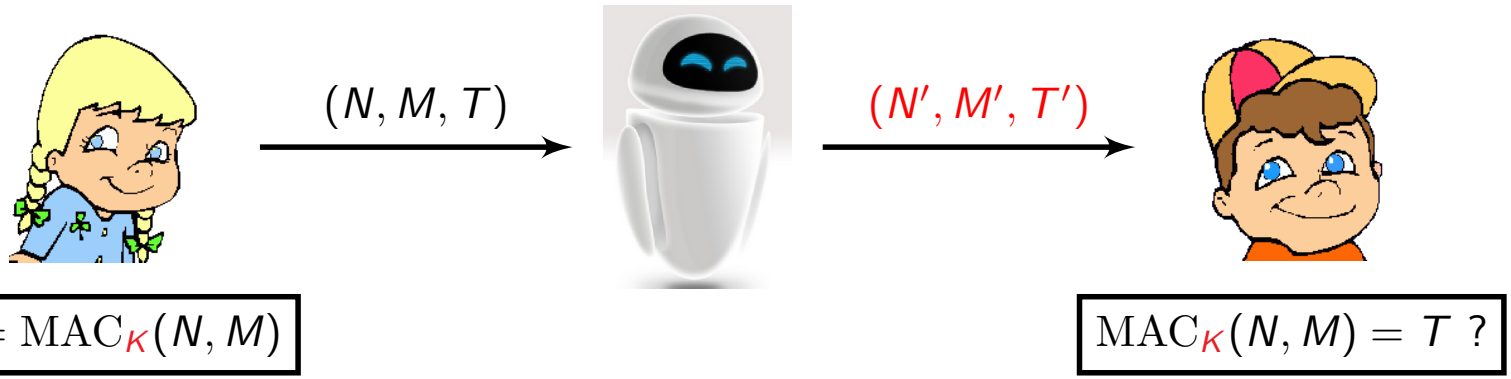
The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(N, M)$
- q_v verification queries (forgery attempts) (N', M', T')

and is successful if one of the verification queries (N', M', T') passes and no previous MAC query (N', M') returned T'.

The adversary is said nonce-respecting if it does not repeat nonces in MAC queries.

Nonce-Based Message Authentication Codes



Security Definition

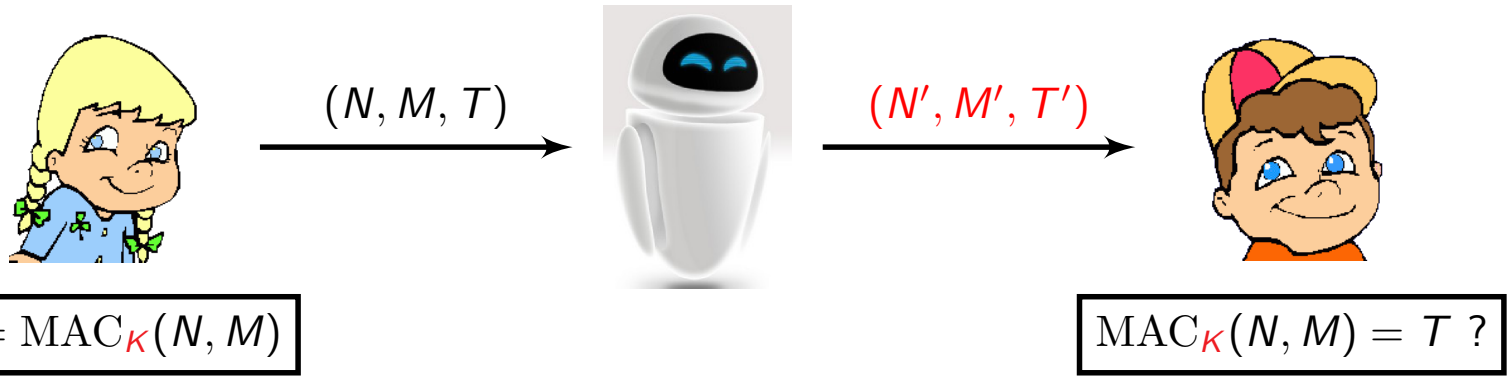
The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(N, M)$
- q_v verification queries (forgery attempts) (N', M', T')

and is successful if one of the verification queries (N', M', T') passes and no previous MAC query (N', M') returned T'.

The adversary is said nonce-respecting if it does not repeat nonces in MAC queries.

Nonce-Based Message Authentication Codes



Security Definition

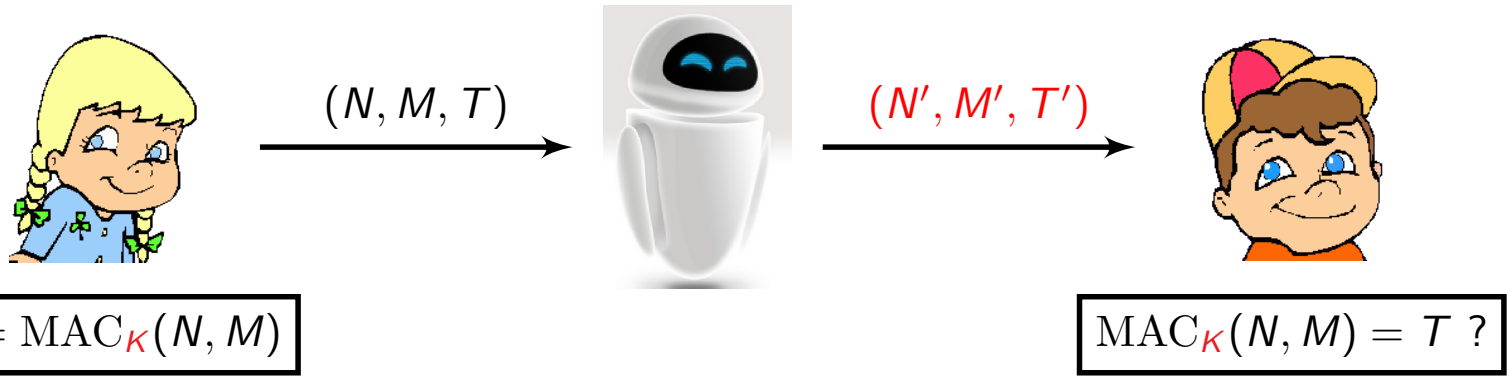
The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(N, M)$
- q_v verification queries (forgery attempts) (N', M', T')

and is successful if one of the verification queries (N', M', T') passes and no previous MAC query (N', M') returned T'.

The adversary is said nonce-respecting if it does not repeat nonces in MAC queries.

Nonce-Based Message Authentication Codes



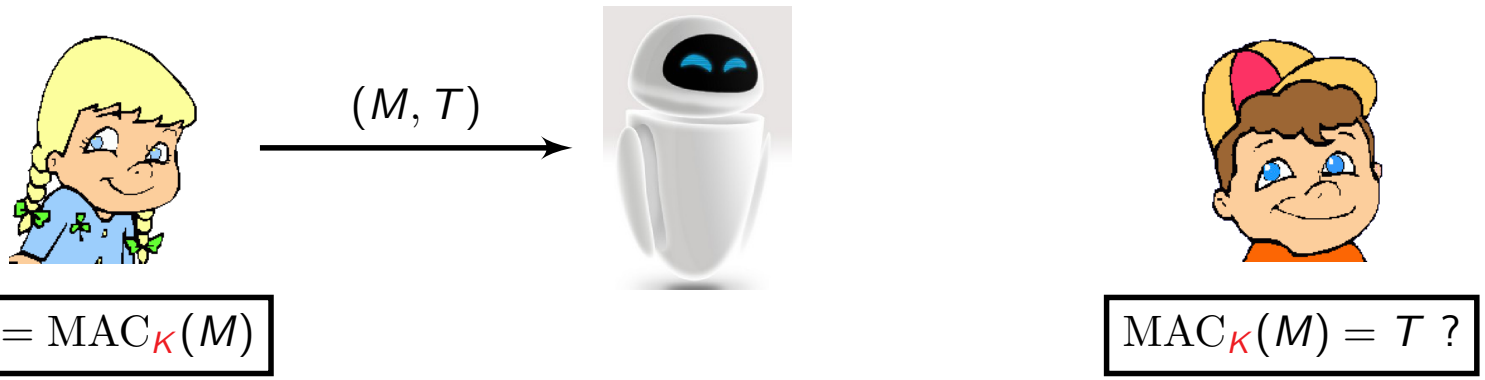
Security Definition

The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(N, M)$
- q_v verification queries (forgery attempts) (N', M', T')

and is successful if one of the verification queries (N', M', T') passes and no previous MAC query (N', M') returned T'.
The adversary is said nonce-respecting if it does not repeat nonces in MAC queries.

Deterministic Message Authentication Codes



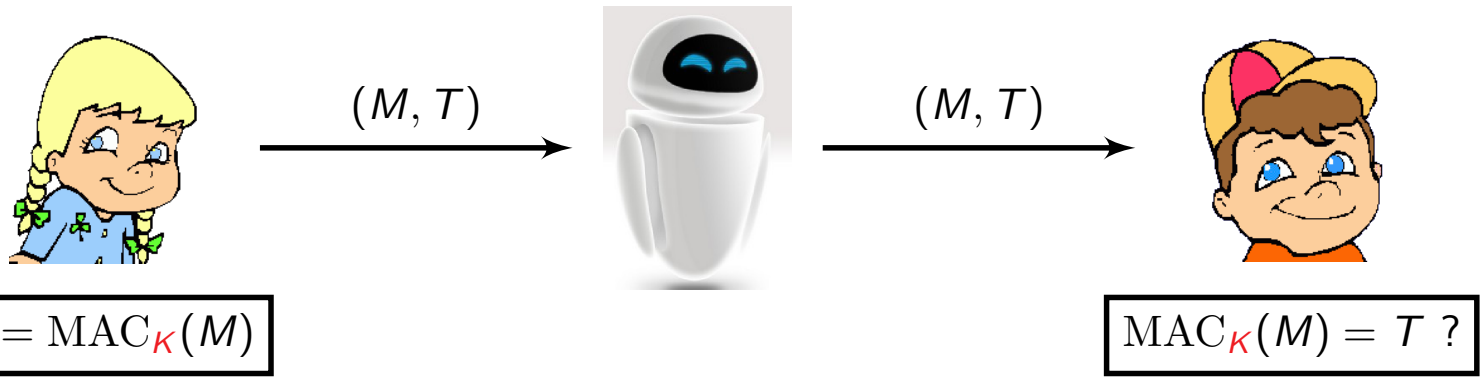
Security Definition

The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(M)$
- q_v verification queries (forgery attempts) (M', T')

and is successful if one of the verification queries (M', T') passes and no previous MAC query M' returned T'.

Deterministic Message Authentication Codes



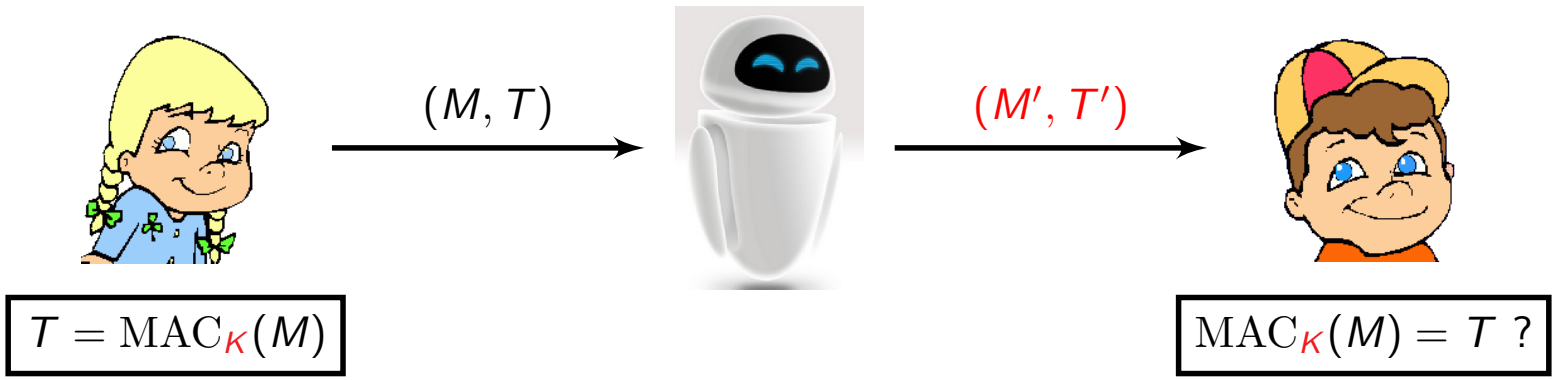
Security Definition

The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(M)$
- q_v verification queries (forgery attempts) (M', T')

and is successful if one of the verification queries (M', T') passes and no previous MAC query M' returned T'.

Deterministic Message Authentication Codes



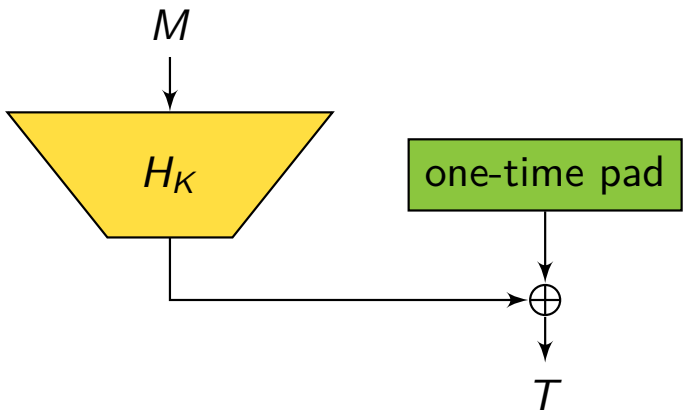
Security Definition

The adversary is allowed

- q_m MAC queries $T = \mathrm{MAC}_K(M)$
- q_v verification queries (forgery attempts) (M', T')

and is successful if one of the verification queries (M', T') passes and no previous MAC query M' returned T'.

Wegman-Carter MACs [GMS74, WC81]



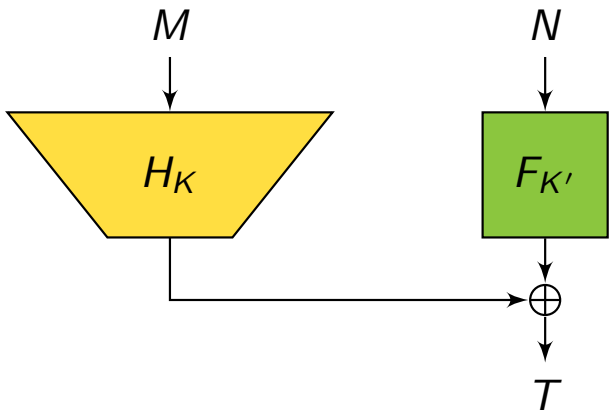
- based on an ε-almost xor-universal (ε-AXU) hash function H:

$$\forall M \neq M', \forall Y, \Pr[K \leftarrow_{\$} \mathcal{K} : H_K(M) \oplus H_K(M') = Y] \leq \varepsilon$$

- in practice, OTPs are replaced by a PRF applied to a nonce N
- H usually based on polynomial evaluation (GCM, Poly1305)
- "optimal" security:

$$\mathbf{Adv}^{\mathrm{MAC}}_{\mathrm{WC}}(q_m, q_v) \leq \varepsilon q_v + \mathbf{Adv}^{\mathrm{PRF}}_{F}(q_m + q_v)$$

Wegman-Carter MACs [GMS74, WC81]



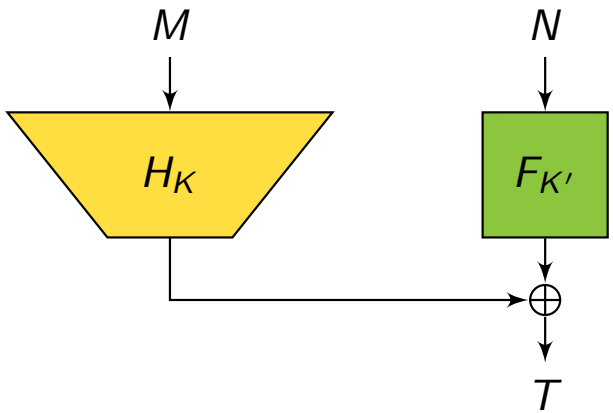
- based on an ε-almost xor-universal (ε-AXU) hash function H:

$$\forall M \neq M', \forall Y, \Pr[K \leftarrow_{\$} \mathcal{K} : H_K(M) \oplus H_K(M') = Y] \leq \varepsilon$$

- in practice, OTPs are replaced by a PRF applied to a nonce N
- H usually based on polynomial evaluation (GCM, Poly1305)
- "optimal" security:

$$\mathbf{Adv}^{\mathrm{MAC}}_{\mathrm{WC}}(q_m, q_v) \leq \varepsilon q_v + \mathbf{Adv}^{\mathrm{PRF}}_{F}(q_m + q_v)$$

Wegman-Carter MACs [GMS74, WC81]



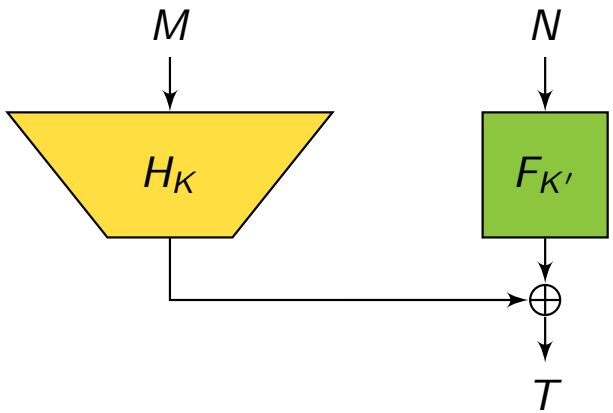
- based on an ε-almost xor-universal (ε-AXU) hash function H:

$$\forall M \neq M', \forall Y, \Pr[K \leftarrow_{\$} \mathcal{K} : H_K(M) \oplus H_K(M') = Y] \leq \varepsilon$$

- in practice, OTPs are replaced by a PRF applied to a nonce N
- H usually based on polynomial evaluation (GCM, Poly1305)
- "optimal" security:

$$\mathbf{Adv}^{\mathrm{MAC}}_{\mathrm{WC}}(q_m, q_v) \leq \varepsilon q_v + \mathbf{Adv}^{\mathrm{PRF}}_{F}(q_m + q_v)$$

Wegman-Carter MACs [GMS74, WC81]



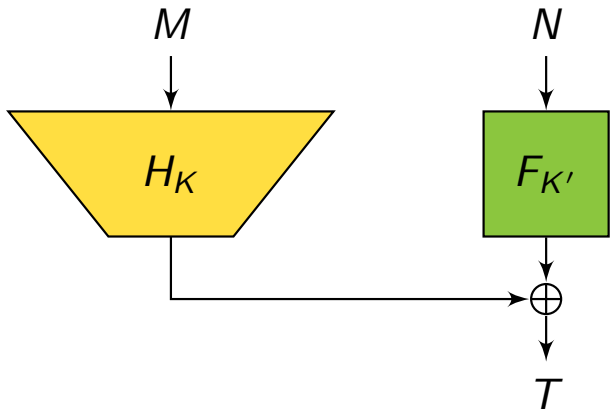
- based on an ε-almost xor-universal (ε-AXU) hash function H:

$$\forall M \neq M', \forall Y, \Pr[K \leftarrow_{\$} \mathcal{K} : H_K(M) \oplus H_K(M') = Y] \leq \varepsilon$$

- in practice, OTPs are replaced by a PRF applied to a nonce N
- H usually based on polynomial evaluation (GCM, Poly1305)
- "optimal" security:

$$\mathbf{Adv}^{\mathrm{MAC}}_{\mathrm{WC}}(q_m, q_v) \leq \varepsilon q_v + \mathbf{Adv}^{\mathrm{PRF}}_{F}(q_m + q_v)$$

The Nonce-Misuse Problem (1/2)

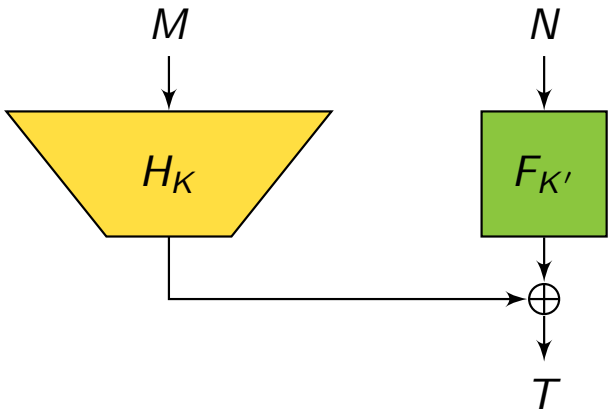


- Wegman-Carter MACs are brittle: a single nonce repetition can completely break security [Jou06, HP08]
- esp. for polynomial-based hashing, i.e., $H_K(M) = P_M(K)$:

$$\begin{cases} P_M(K) \oplus F_{K'}(N) = T \\ P_{M'}(K) \oplus F_{K'}(N) = T' \end{cases} \Rightarrow P_M(K) \oplus P_{M'}(K) = T \oplus T'$$

- solution: extra PRF call (in fact, OK to use a PRP here)

The Nonce-Misuse Problem (1/2)

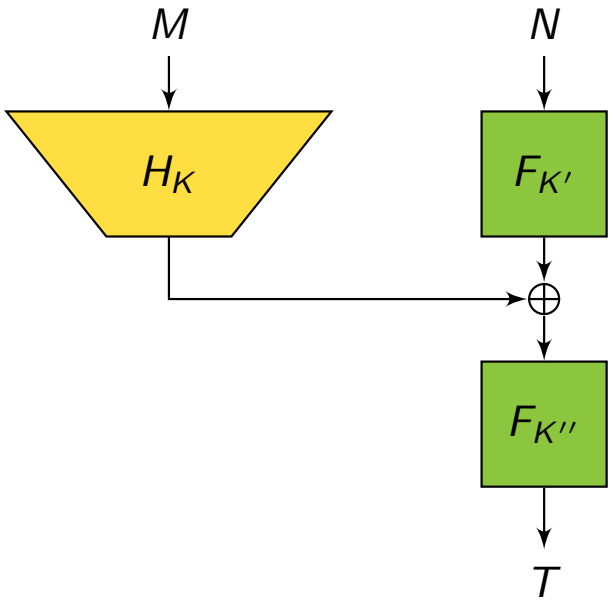


- Wegman-Carter MACs are brittle: a single nonce repetition can completely break security [Jou06, HP08]
- esp. for polynomial-based hashing, i.e., $H_K(M) = P_M(K)$:

$$\begin{cases} P_M(K) \oplus F_{K'}(N) = T \\ P_{M'}(K) \oplus F_{K'}(N) = T' \end{cases} \Rightarrow P_M(K) \oplus P_{M'}(K) = T \oplus T'$$

- solution: extra PRF call (in fact, OK to use a PRP here)

The Nonce-Misuse Problem (1/2)

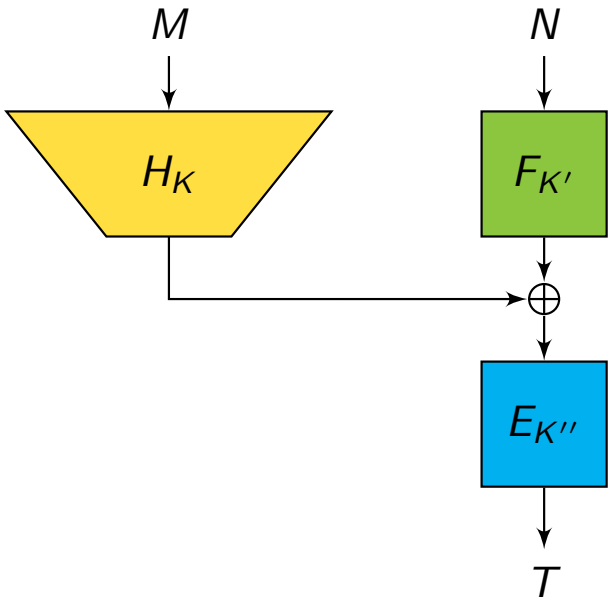


- Wegman-Carter MACs are brittle: a single nonce repetition can completely break security [Jou06, HP08]
- esp. for polynomial-based hashing, i.e., $H_K(M) = P_M(K)$:

$$\begin{cases} P_M(K) \oplus F_{K'}(N) = T \\ P_{M'}(K) \oplus F_{K'}(N) = T' \end{cases} \Rightarrow P_M(K) \oplus P_{M'}(K) = T \oplus T'$$

- solution: extra PRF call (in fact, OK to use a PRP here)

The Nonce-Misuse Problem (1/2)

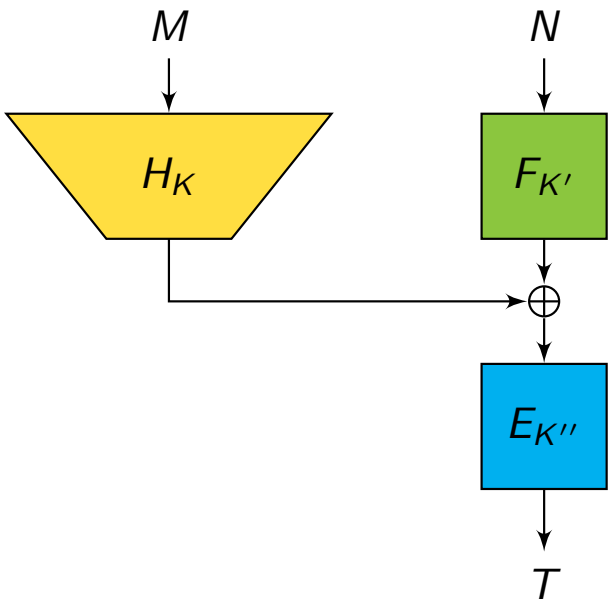


- Wegman-Carter MACs are brittle: a single nonce repetition can completely break security [Jou06, HP08]
- esp. for polynomial-based hashing, i.e., $H_K(M) = P_M(K)$:

$$\begin{cases} P_M(K) \oplus F_{K'}(N) = T \\ P_{M'}(K) \oplus F_{K'}(N) = T' \end{cases} \Rightarrow P_M(K) \oplus P_{M'}(K) = T \oplus T'$$

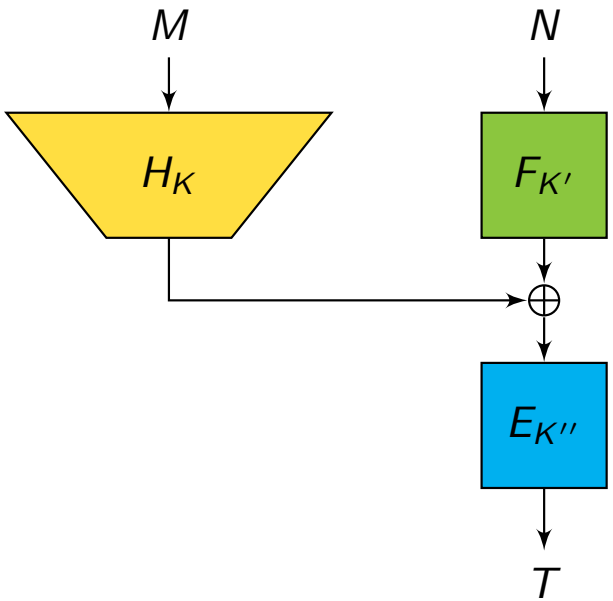
- solution: extra PRF call (in fact, OK to use a PRP here)

The Nonce-Misuse Problem (2/2)



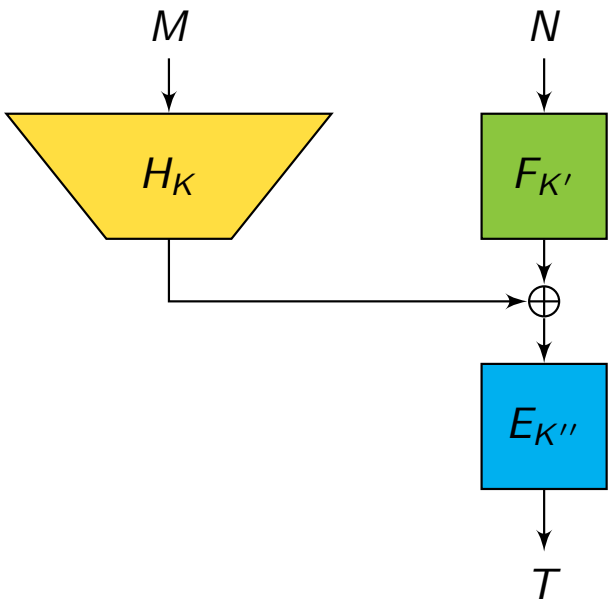
- good security against *nonce-respecting* adversaries ;
- BUT security drops to the birthday bound when a nonce is used twice ;
- same problem when implementing F from a Block Cipher ;
- too simple mixing of the nonce and the hash of the message...

The Nonce-Misuse Problem (2/2)



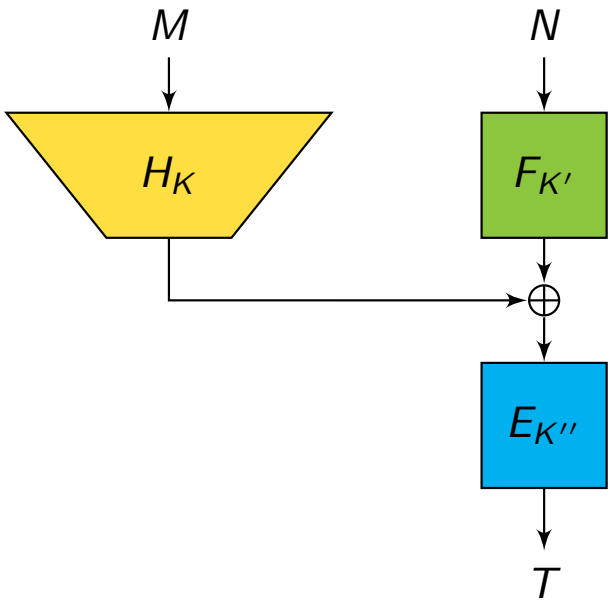
- good security against *nonce-respecting* adversaries ;
- BUT security drops to the birthday bound when a nonce is used twice ;
- same problem when implementing F from a Block Cipher ;
- too simple mixing of the nonce and the hash of the message...

The Nonce-Misuse Problem (2/2)



- good security against *nonce-respecting* adversaries ;
- BUT security drops to the birthday bound when a nonce is used twice ;
- same problem when implementing F from a Block Cipher ;
- too simple mixing of the nonce and the hash of the message...

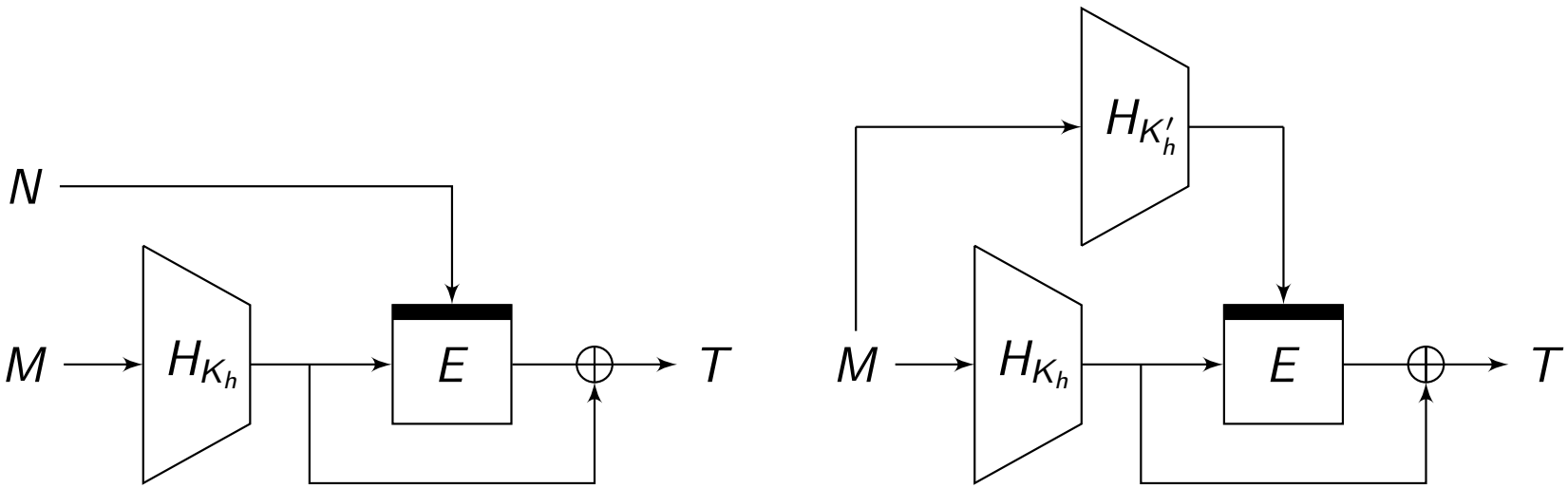
The Nonce-Misuse Problem (2/2)



- good security against *nonce-respecting* adversaries ;
- BUT security drops to the birthday bound when a nonce is used twice ;
- same problem when implementing F from a Block Cipher ;
- too simple mixing of the nonce and the hash of the message...

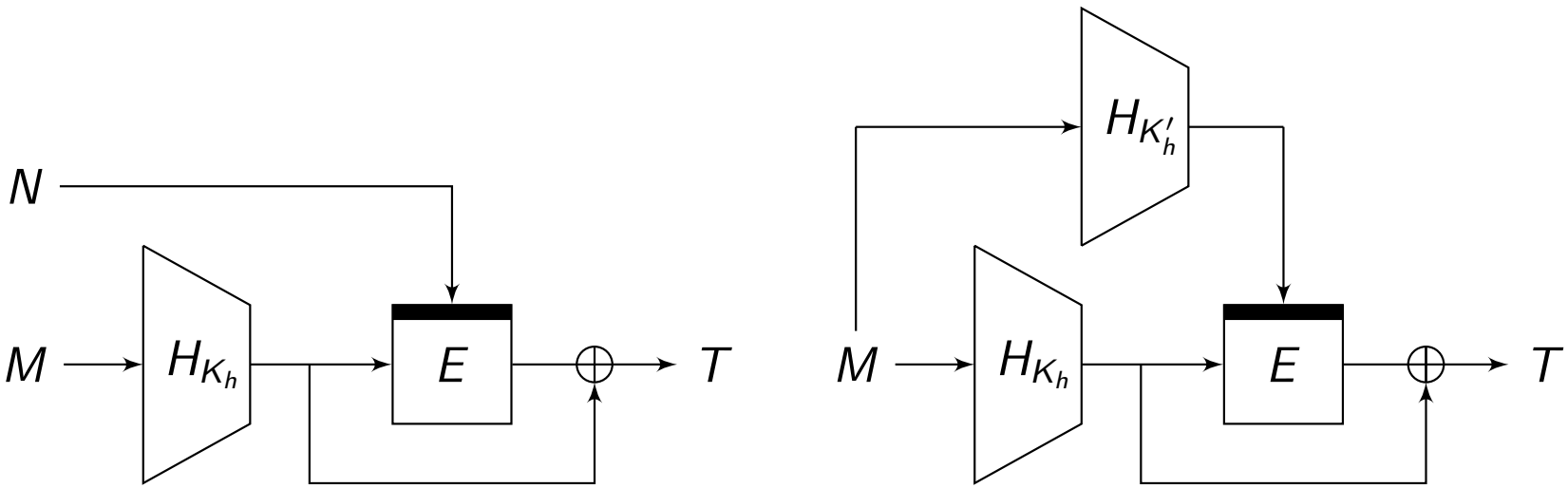
Outline

Our BC Based MACs



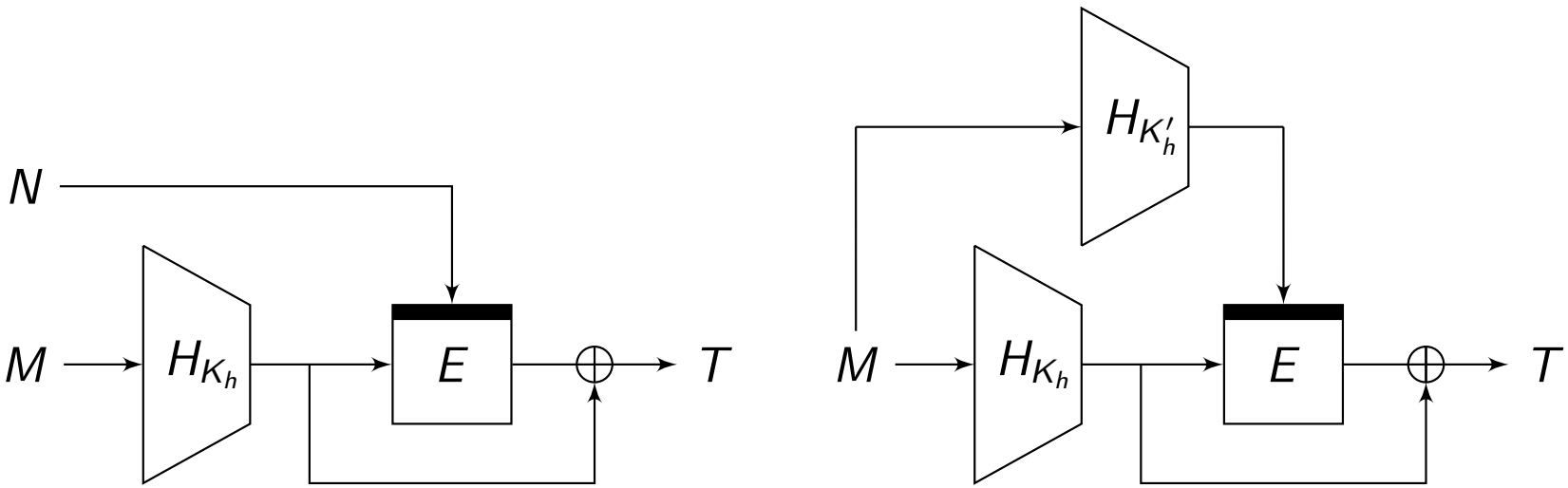
- Dubbed the HENK (*Hash-then-Encrypt with Nonce as Key*) and HEHK (*Hash-then-Encrypt with Hash as Key*) constructions.
- Based on a BC E and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the BC and 1 call (2 for HEHK) to H.
- Provably secure in the Ideal Cipher Model.

Our BC Based MACs



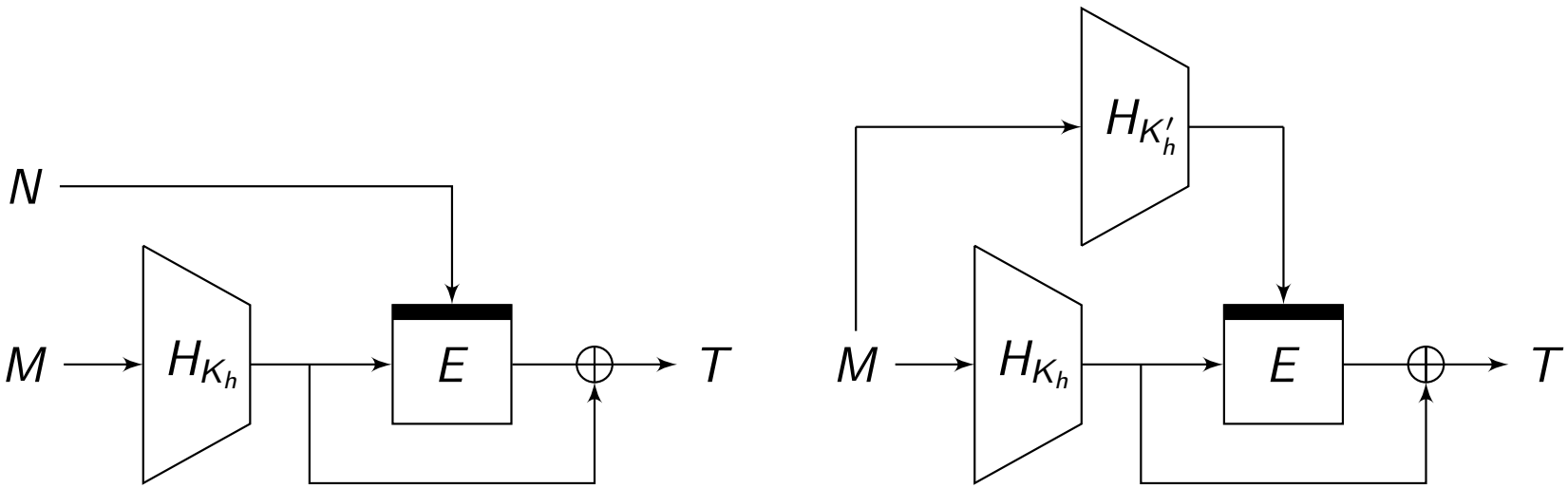
- Dubbed the HENK (*Hash-then-Encrypt with Nonce as Key*) and HEHK (*Hash-then-Encrypt with Hash as Key*) constructions.
- Based on a BC E and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the BC and 1 call (2 for HEHK) to H.
- Provably secure in the Ideal Cipher Model.

Our BC Based MACs



- Dubbed the HENK (*Hash-then-Encrypt with Nonce as Key*) and HEHK (*Hash-then-Encrypt with Hash as Key*) constructions.
- Based on a BC E and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the BC and 1 call (2 for HEHK) to H.
- Provably secure in the Ideal Cipher Model.

Our BC Based MACs



- Dubbed the HENK (*Hash-then-Encrypt with Nonce as Key*) and HEHK (*Hash-then-Encrypt with Hash as Key*) constructions.
- Based on a BC E and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the BC and 1 call (2 for HEHK) to H.
- Provably secure in the Ideal Cipher Model.

The HENK construction and its randomized variant HERK

- probability of forgery for a $(\mu, q_e, q_m, q_v)-$adversary is lower than

$$(5\mu + n - 2)\varepsilon q + \frac{q}{2^n - \mu - q} + \left(\frac{q}{2^n - q}\right)^{n+1},$$

where $q = \max(q_e, q_m, q_v)$ (more accurate bound in the paper).

- Randomized variant is dubbed the HERK construction (*Hash-then-Encrypt with Random Key*).
- probability of forgery for a $(q_e, q_m, q_v)-$adversary is then lower than

$$(6n - 2)\varepsilon q + \frac{q}{2^n - n - q} + \left(\frac{q}{2^n - q}\right)^{n+1} + \left(\frac{q}{|\mathcal{K}|}\right)^n \frac{q}{2^n}.$$

The HENK construction and its randomized variant HERK

- probability of forgery for a $(\mu, q_e, q_m, q_v)-$adversary is lower than

$$(5\mu + n - 2)\varepsilon q + \frac{q}{2^n - \mu - q} + \left(\frac{q}{2^n - q}\right)^{n+1},$$

  where $q = \max(q_e, q_m, q_v)$ (more accurate bound in the paper).
- Randomized variant is dubbed the HERK construction (*Hash-then-Encrypt with Random Key*).
- probability of forgery for a $(q_e, q_m, q_v)-$adversary is then lower than

$$(6n - 2)\varepsilon q + \frac{q}{2^n - n - q} + \left(\frac{q}{2^n - q}\right)^{n+1} + \left(\frac{q}{|\mathcal{K}|}\right)^{n} \frac{q}{2^n}.$$

The HENK construction and its randomized variant HERK

- probability of forgery for a $(\mu, q_e, q_m, q_v)-$adversary is lower than

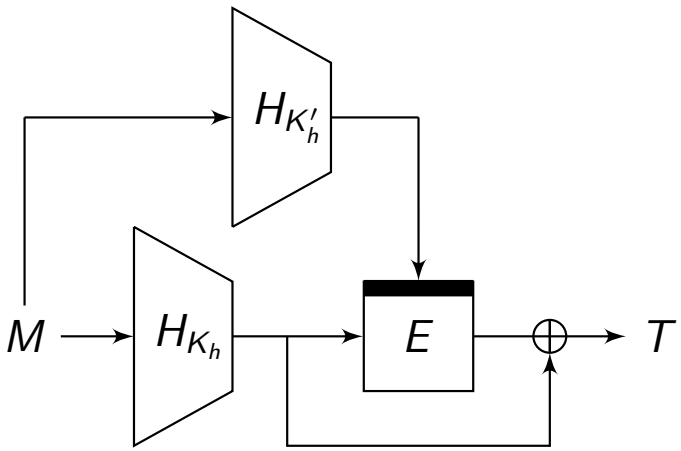
$$(5\mu + n - 2)\varepsilon q + \frac{q}{2^n - \mu - q} + \left(\frac{q}{2^n - q}\right)^{n+1},$$

where $q = \max(q_e, q_m, q_v)$ (more accurate bound in the paper).

- Randomized variant is dubbed the HERK construction (*Hash-then-Encrypt with Random Key*).
- probability of forgery for a $(q_e, q_m, q_v)-$adversary is then lower than

$$(6n - 2)\varepsilon q + \frac{q}{2^n - n - q} + \left(\frac{q}{2^n - q}\right)^{n+1} + \left(\frac{q}{|\mathcal{K}|}\right)^n \frac{q}{2^n}.$$

The HEHK construction



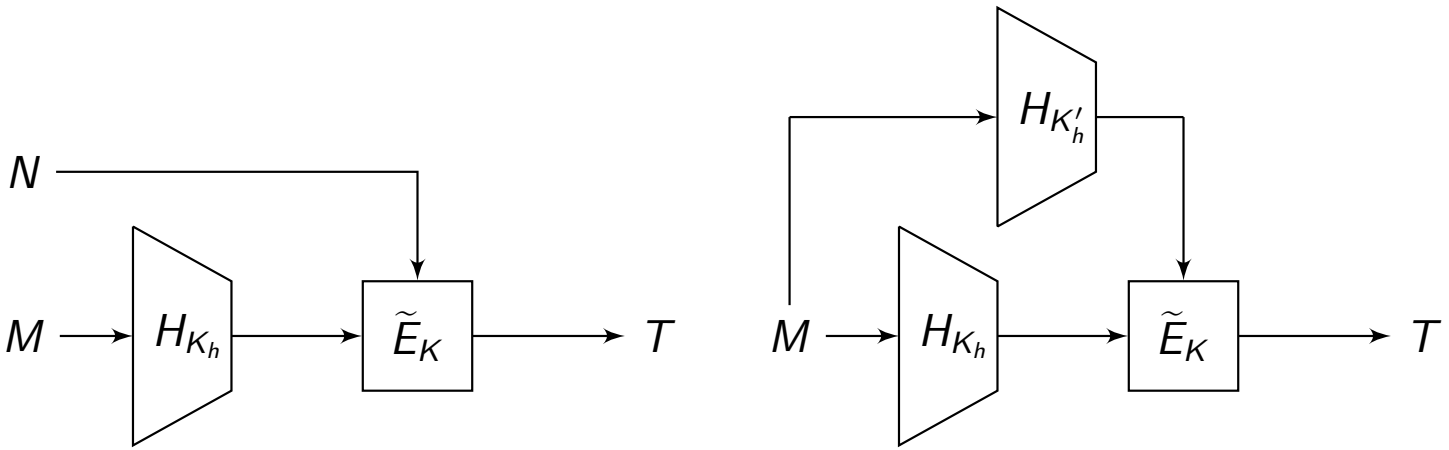
Probability of forgery for a $(q_e, q_m, q_v)-$adversary is lower than

$$6\varepsilon^2 q^2 + \frac{q}{2^n - 2q},$$

where $q = \max(q_e, q_m, q_v)$.

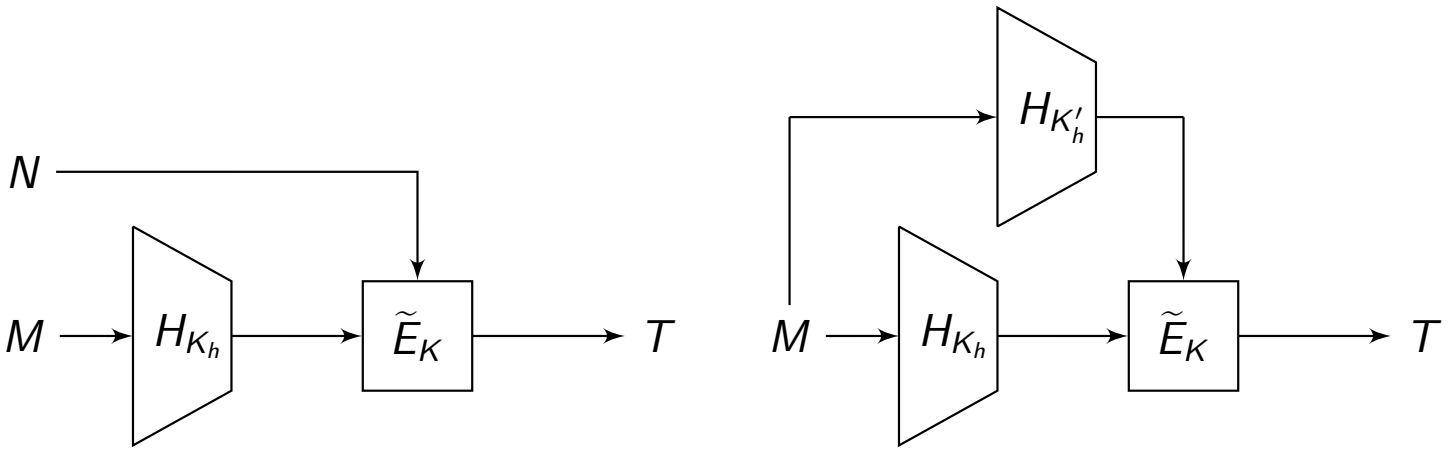
Outline

Our TBC-based MAC constructions



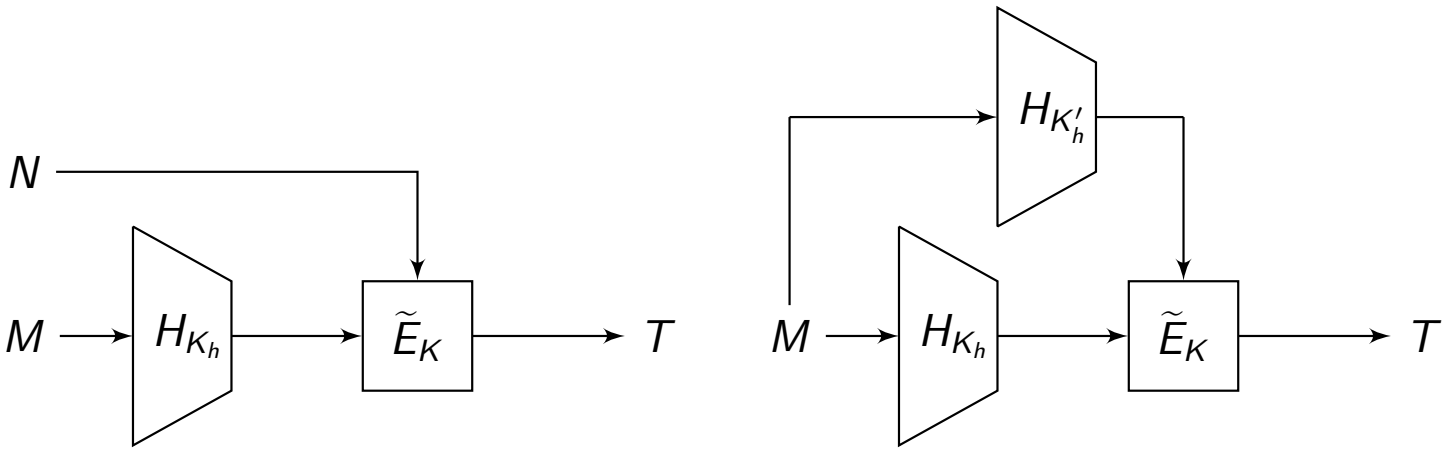
- Dubbed the HENT (*Hash-then-Encrypt with Nonce as Tweak*) and HEHT (*Hash-then-Encrypt with Hash as Tweak*) construction.
- Based on a TBC $\widetilde{E}$ and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the TBC and 1 call (2 calls for HEHT) to H.
- Provably secure in the Standard Model!

Our TBC-based MAC constructions



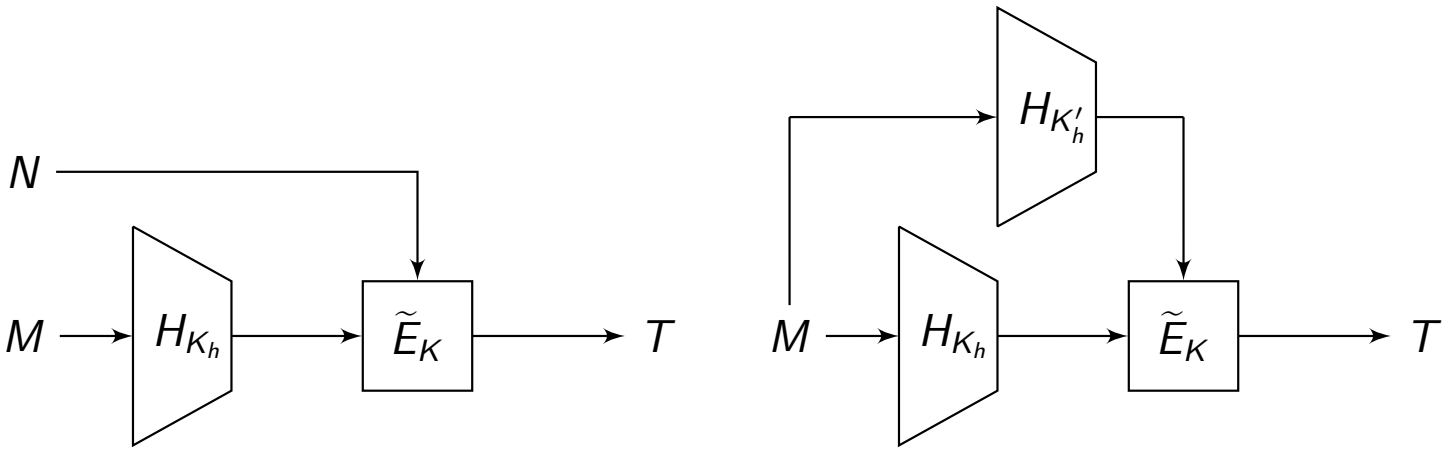
- Dubbed the HENT (*Hash-then-Encrypt with Nonce as Tweak*) and HEHT (*Hash-then-Encrypt with Hash as Tweak*) construction.
- Based on a TBC $\widetilde{E}$ and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the TBC and 1 call (2 calls for HEHT) to H.
- Provably secure in the Standard Model!

Our TBC-based MAC constructions



- Dubbed the HENT (*Hash-then-Encrypt with Nonce as Tweak*) and HEHT (*Hash-then-Encrypt with Hash as Tweak*) construction.
- Based on a TBC $\widetilde{E}$ and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the TBC and 1 call (2 calls for HEHT) to H.
- Provably secure in the Standard Model!

Our TBC-based MAC constructions



- Dubbed the HENT (*Hash-then-Encrypt with Nonce as Tweak*) and HEHT (*Hash-then-Encrypt with Hash as Tweak*) construction.
- Based on a TBC $\widetilde{E}$ and a ε-AXU and uniform hash function H.
- Efficient: 1 call to the TBC and 1 call (2 calls for HEHT) to H.
- Provably secure in the Standard Model!

The HENT construction and its randomized variant HERT

- Probability of forgery for a $(\mu, q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(2q) + (3\mu - 2)q\varepsilon + \frac{q}{2^n - \mu}$$

  where $q = \max(q_e, q, q_v)$ (batter bound in the paper).
- Randomized variant is dubbed the HERT construction (*Hash-then-Encrypt with Random Tweak*).
- probability of forgery for a $(q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(2q) + (3n - 2)q\varepsilon + \frac{q}{2^n - n} + \frac{q^{n+1}}{|\mathcal{T}|^n \cdot 2^n}.$$

The HENT construction and its randomized variant HERT

- Probability of forgery for a $(\mu, q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(2q) + (3\mu - 2)q\varepsilon + \frac{q}{2^n - \mu}$$

  where $q = \max(q_e, q, q_v)$ (batter bound in the paper).
- Randomized variant is dubbed the HERT construction (*Hash-then-Encrypt with Random Tweak*).
- probability of forgery for a $(q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(2q) + (3n - 2)q\varepsilon + \frac{q}{2^n - n} + \frac{q^{n+1}}{|\mathcal{T}|^n \cdot 2^n}.$$

The HENT construction and its randomized variant HERT

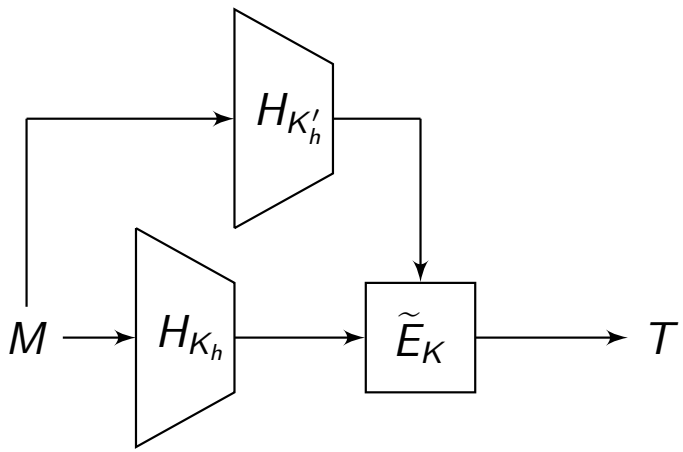
- Probability of forgery for a $(\mu, q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}}(2q) + (3\mu - 2)q\varepsilon + \frac{q}{2^n - \mu}$$

  where $q = \max(q_e, q, q_v)$ (batter bound in the paper).
- Randomized variant is dubbed the HERT construction (*Hash-then-Encrypt with Random Tweak*).
- probability of forgery for a $(q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}}(2q) + (3n - 2)q\varepsilon + \frac{q}{2^n - n} + \frac{q^{n+1}}{|\mathcal{T}|^n \cdot 2^n}.$$

The HEHT construction



Probability of forgery for a $(q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}}(q_m + q_v) + 2\varepsilon^2 q_m^2 + \varepsilon^2 q_m q_v + \frac{q_v}{2^n - q_m}.$$

Outline

What about truncated variations of our constructions ?

Our constructions compose well with truncation.

What about truncated variations of our constructions ?

Our constructions compose well with truncation.

E.g., if one takes the first s bits of the outputs of the HEHT construction, the probability of forgery of a $(q_m, q_v)-$adversary is lower than

$$\mathbf{Adv}_{\widetilde{E}}^{\mathrm{TPRP}}(\mathrm{A}') + 2\varepsilon^2 q_m^2 + 2^{n-s}\varepsilon^2 q_m q_v + \frac{2^{n-s} q_v}{2^n - q_m}.$$

Outline

Proof strategy

- H-coefficients technique [Pat08, CS14] (good/bad transcripts).
- No "Ideal world" in the MAC security notions.
- "Artificial" ideal world where the MAC oracle is a random oracle and the verification oracle always rejects.
- As usual, at the end of the interaction, the hashing key is revealed to the adversary.
- Bad transcripts: when the ideal world does things that are impossible in the real world.

Proof strategy

- H-coefficients technique [Pat08, CS14] (good/bad transcripts).
- No "Ideal world" in the MAC security notions.
- "Artificial" ideal world where the MAC oracle is a random oracle and the verification oracle always rejects.
- As usual, at the end of the interaction, the hashing key is revealed to the adversary.
- Bad transcripts: when the ideal world does things that are impossible in the real world.

Proof strategy

- H-coefficients technique [Pat08, CS14] (good/bad transcripts).
- No "Ideal world" in the MAC security notions.
- "Artificial" ideal world where the MAC oracle is a random oracle and the verification oracle always rejects.
- As usual, at the end of the interaction, the hashing key is revealed to the adversary.
- Bad transcripts: when the ideal world does things that are impossible in the real world.

Proof strategy

- H-coefficients technique [Pat08, CS14] (good/bad transcripts).
- No "Ideal world" in the MAC security notions.
- "Artificial" ideal world where the MAC oracle is a random oracle and the verification oracle always rejects.
- As usual, at the end of the interaction, the hashing key is revealed to the adversary.
- Bad transcripts: when the ideal world does things that are impossible in the real world.

Proof strategy

- H-coefficients technique [Pat08, CS14] (good/bad transcripts).
- No "Ideal world" in the MAC security notions.
- "Artificial" ideal world where the MAC oracle is a random oracle and the verification oracle always rejects.
- As usual, at the end of the interaction, the hashing key is revealed to the adversary.
- Bad transcripts: when the ideal world does things that are impossible in the real world.

Bad transcripts for HENT (1/2)

- In the ideal world, the verification oracle always rejects $\Rightarrow$ avoid situations where an adversary would have won in the real world:
  - happens if there exists a MAC query $(N_i, M_i, T_i) \in \tau_m$ and a verification query $(N'_j, M'_j, T'_j, b_j) \in \tau_v$ such that

$$\begin{cases} N_i = N'_j \\ T_i = T'_j \\ H_{K_h}(M_i) = H_{K_h}(M'_j), \end{cases}$$

  - BUT only with probability lower than $\mu q_v \varepsilon$.

Bad transcripts for HENT (1/2)

- In the ideal world, the verification oracle always rejects $\Rightarrow$ avoid situations where an adversary would have won in the real world:
  - happens if there exists a MAC query $(N_i, M_i, T_i) \in \tau_m$ and a verification query $(N'_j, M'_j, T'_j, b_j) \in \tau_v$ such that

$$\begin{cases} N_i = N'_j \\ T_i = T'_j \\ H_{K_h}(M_i) = H_{K_h}(M'_j), \end{cases}$$

  - BUT only with probability lower than $\mu q_v \varepsilon$.

Bad transcripts for HENT (1/2)

- In the ideal world, the verification oracle always rejects $\Rightarrow$ avoid situations where an adversary would have won in the real world:
  - happens if there exists a MAC query $(N_i, M_i, T_i) \in \tau_m$ and a verification query $(N'_j, M'_j, T'_j, b_j) \in \tau_v$ such that

$$\begin{cases} N_i = N'_j \\ T_i = T'_j \\ H_{K_h}(M_i) = H_{K_h}(M'_j), \end{cases}$$

  - BUT only with probability lower than $\mu q_v \varepsilon$.

Bad transcripts for HENT (2/2)

- Avoid collisions which make the transcript incompatible with a permutation:
  - can happen if there exists two distinct MAC queries (N, M, T) and (N', M', T') such that $N = N'$ and either $H_{K_h}(M) = H_{K_h}(M')$ or $T = T'$.
  - happens with probability lower than $2(\mu - 1)q_m\varepsilon$.

Bad transcripts for HENT (2/2)

- Avoid collisions which make the transcript incompatible with a permutation:
  - can happen if there exists two distinct MAC queries (N, M, T) and (N', M', T') such that $N = N'$ and either $H_{K_h}(M) = H_{K_h}(M')$ or $T = T'$.
  - happens with probability lower than $2(\mu - 1)q_m\varepsilon$.

Bad transcripts for HENT (2/2)

- Avoid collisions which make the transcript incompatible with a permutation:
  - can happen if there exists two distinct MAC queries (N, M, T) and (N', M', T') such that $N = N'$ and either $H_{K_h}(M) = H_{K_h}(M')$ or $T = T'$.
  - happens with probability lower than $2(\mu - 1)q_m\varepsilon$.

Good transcripts for HENT

Fix any good transcript $\tau = (\tau_m, \tau_v, K_h)$

- r is the number of distinct tweaks used in τ_m, and for $i = 1, \ldots, r$, q_i is the number of occurences of the $i-$th tweak in τ_m;
- then

$$\Pr[X_{\mathrm{id}} = \tau] = \frac{1}{|\mathcal{K}_h| \cdot 2^{nq_m}},$$

$$\Pr[X_{\mathrm{re}} = \tau] \geq \frac{1}{\prod_{i=1}^{r} (2^n)_{q_i}} \left(1 - \frac{q_v}{2^n - \mu}\right),$$

- and

$$\Pr[X_{\mathrm{re}} = \tau] / \Pr[X_{\mathrm{id}} = \tau] \geq 1 - \frac{q_v}{2^n - \mu}.$$

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Conclusion

- We propose efficient and provably secure MAC constructions based on a BC or a TBC and $\varepsilon-$AXU and uniform hash function:
- two Nonce-based MACs (HENK and HENT) whose security degrades linearly as the number of nonce reuses grows,
- two Randomized MACs (HERK and HERT) based on the previous ones,
- and two Deterministic MACs with security up to roughly ε^{-1}.
- Paper will soon be on ePrint.

Thanks for your attention !

References I

Shan Chen and John Steinberger. Tight Security Bounds for Key-Alternating Ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, 2014. Full version available at http://eprint.iacr.org/2013/222.

Edgar N. Gilbert, F. Jessie MacWilliams, and Neil J. A. Sloane. Codes which detect deception. *Bell System Technical Journal*, 53(3):405–424, 1974.

Helena Handschuh and Bart Preneel. Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 144–161. Springer, 2008.

Antoine Joux. Authentication Failures in NIST Version of GCM. Comments submitted to NIST Modes of Operation Process, 2006. Available at http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/800-38_Series-Drafts/GCM/Joux_comments.pdf.

References II

Jacques Patarin. The "Coefficients H" Technique. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography - SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, 2008.

Mark N. Wegman and Larry Carter. New Hash Functions and Their Use in Authentication and Set Equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.