Pipelineable On-Line Encryption with Tag (POET)

Farzaneh Abed[2] Scott Fluhrer[1] John Foley[1]
Christian Forler[2] Eik List[2] Stefan Lucks[2] David McGrew[1]
Jakob Wenzel[2]

[1] Cisco Systems, [2] Bauhaus-Universität Weimar

DIAC 2014

Santa Barbara, CA

Outline

1. Motivation
   - Case Study: OTN
   - Decryption Misuse
2. CAESAR Submission POET
3. Security of POET

Section 1

Motivation

Case Study: Optical Transport Network (OTN)

Task:

- *Secure* network traffic . . .

Case Study: Optical Transport Network (OTN)

Task:

- *Secure* network traffic . . .
- . . . of real-time applications . . .

Case Study: Optical Transport Network (OTN)

Task:

- *Secure* network traffic ...
- ...of real-time applications ...
- ...in an Optical Transport Network (OTN)

Case Study: Optical Transport Network (OTN)

Task:

- *Secure* network traffic . . .
- . . . of real-time applications . . .
- . . . in an Optical Transport Network (OTN)
  - High throughput (40 - 100 Gbit/s)
  - Low latency (few clock cycles)
  - Large message frames (64 KB)
    (usually consist of multiple TCP/IP or UDP/IP packages)

Requirements for OTNs

Security requirements:

- Data privacy (IND-CPA), and
- Data integrity (INT-CTXT)

Requirements for OTNs

Security requirements:

- Data privacy (IND-CPA), and
- Data integrity (INT-CTXT)

Functional requirements:

- On-line encryption/decryption

Problem and Workarounds

Problem: High Latency of Authenticated Decryption

1. Decryption of the *entire* message
2. Verification of the authentication tag

For 64-kB frames we have 4,096 ciphertext blocks (128 bits)

Problem and Workarounds

Problem: High Latency of Authenticated Decryption

1. Decryption of the *entire* message
2. Verification of the authentication tag

For 64-kB frames we have 4,096 ciphertext blocks (128 bits)

- Workarounds:
  - Decrypt-then-mask? [Fouque et al. 03] $\Rightarrow$ latency again
  - Pass plaintext beforehand and hope. . .

Problem and Workarounds

Problem: High Latency of Authenticated Decryption

1. Decryption of the *entire* message
2. Verification of the authentication tag

For 64-kB frames we have 4,096 ciphertext blocks (128 bits)

- Workarounds:
  - Decrypt-then-mask? [Fouque et al. 03] ⇒ latency again
  - Pass plaintext beforehand and hope. . .
- Drawbacks:
  - Plaintext information would leak if authentication tag invalid
  - Literature calls this setting *decryption-misuse* [Fleischmann, Forler, and Lucks 12]

How Severe is Decryption-Misuse?

- Puts security at high risk
- CCA-adversary may inject controlled manipulations
- Particularly, CTR-mode based AE schemes

$$C \oplus \Delta \rightarrow_{Dec} M \oplus \Delta$$

How Severe is Decryption-Misuse?

- Puts security at high risk
- CCA-adversary may inject controlled manipulations
- Particularly, CTR-mode based AE schemes

$$C \oplus \Delta \rightarrow_{Dec} M \oplus \Delta$$

Decryption-misuse is *not* covered by existing CCA3-security proofs

Decryption Misuse Resistance

- Best to wish for:
  - Manipulation of ciphertext block C_i
    $\Rightarrow$ completely random plaintext
  - Contradiction to on-line requirement

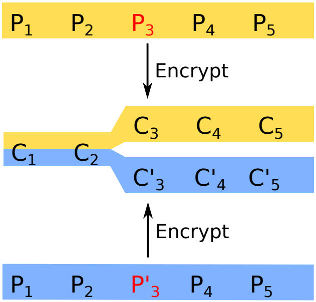
Decryption Misuse Resistance

- Best to wish for:
  - Manipulation of ciphertext block C_i
    $\Rightarrow$ completely random plaintext
  - Contradiction to on-line requirement
- What can we achive with an on-line encryption scheme?
  - Manipulation of $C_i \Rightarrow M_i, M_{i+1}, \ldots$ random garbage
  - Adversary sees at best common message prefixes

Decryption Misuse Resistance

- Best to wish for:
  - Manipulation of ciphertext block C_i
    $\Rightarrow$ completely random plaintext
  - Contradiction to on-line requirement
- What can we achive with an on-line encryption scheme?
  - Manipulation of $C_i \Rightarrow M_i, M_{i+1}, \ldots$ random garbage
  - Adversary sees at best common message prefixes
- The security notion of OPRP-CCA covers this behaviour [Bellare et al. 01]

On-Line Permutation



On-Line Pseudo Random Permutation (OPRP)

Like a PRP with the following property:
Plaintexts with common prefix $\rightarrow$ ciphertexts with common prefix

(Bellare et al.; “Online Ciphers and the Hash-CBC Construction”; CRYPTO’01)

OPRP-CCA

Definition (OPRP-CCA Advantage)

Let P be a random on-line permutation, $\Pi = (K, E, D)$ an on-line encryption scheme, $k \xleftarrow{\$} K()$, and $\mathcal{A}$ be an adversary. Then we have

$$\mathbf{Adv}_{\Pi}^{\text{OPRP-CCA}}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{E_k(.),D_k(.)} \implies 1\right] - \left[\mathcal{A}^{P(.),P^{-1}(.)} \implies 1\right]\right|$$

Intermediate (Authentication) Tags

Assume an OPRP-CCA secure encryption scheme

- Recap: Modifying $C_i \implies M_i, M_{i+1}, \ldots, M_M$ random garbage
- Redundancy in the plaintext (e.g., checksum)

Intermediate (Authentication) Tags

Assume an OPRP-CCA secure encryption scheme

- Recap: Modifying $C_i \implies M_i, M_{i+1}, \ldots, M_M$ random garbage
- Redundancy in the plaintext (e.g., checksum)

$\implies$ intermediate authentication tags

Intermediate (Authentication) Tags

Assume an OPRP-CCA secure encryption scheme

- Recap: Modifying $C_i \implies M_i, M_{i+1}, \ldots, M_M$ random garbage
- Redundancy in the plaintext (e.g., checksum)

$\implies$ intermediate authentication tags

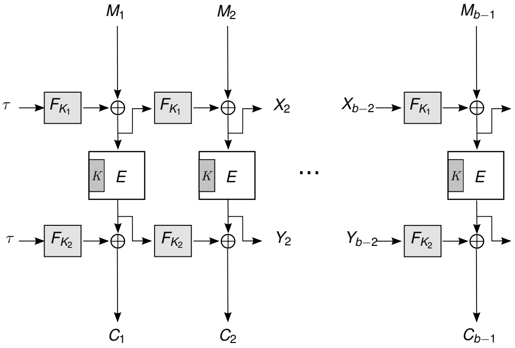
- Common network packets (TCP/IP, UDP/IP) have a checksum

$\implies$ OTN: *16-bit integrity* for free (per packet)

Section 2

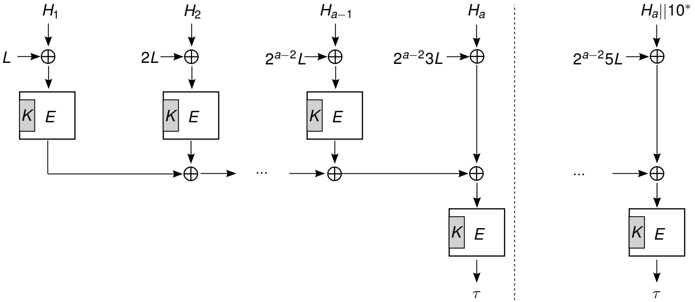
CAESAR Submission POET

Pipeline On-Line Encryption (POE)



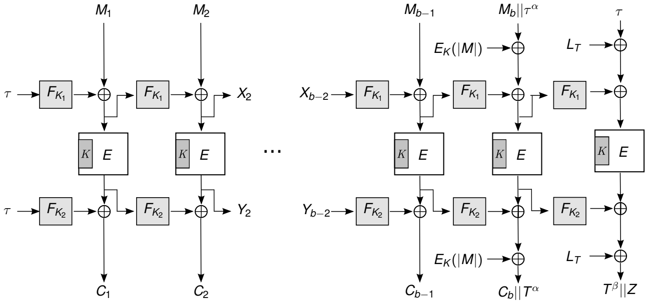
- POE is a OPRP-CCA secure enc scheme [Abed et al. 14]
- Actually, it provides birthday bound security
- POE is used to process a message or ciphertext

POET Header Processing



- We just borrowed the PMAC design [Black & Rogaway 02]
- Nonce is (part of) the header

POET



- Well pipelineable
- 1 BC + 2 AXU hash-function (F) calls per block
- Borrows tag-splitting procedure from McOE
- Robust against **nonce- and decryption-misuse**

Requirements for F

Basic Assumption (F is AXU)

$F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is ϵ-AXU

Requirements for F

Basic Assumption (F is AXU)

$$F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n \text{ is } \epsilon\text{-AXU}$$

Further Assumption (Cascade F^b is AXU)

$$F^b_\kappa(X) := F_\kappa(\ldots(F_\kappa(X_1) \oplus X_2), \ldots) \oplus X_b) \text{ is } b \cdot \epsilon\text{-AXU}$$

Requirements for F

Basic Assumption (F is AXU)

$$F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n \text{ is } \epsilon\text{-AXU}$$

***Further* Assumption (Cascade F^b is AXU)**

$$F^b_\kappa(X) := F_\kappa(\ldots(F_\kappa(X_1) \oplus X_2), \ldots) \oplus X_b) \text{ is } b \cdot \epsilon\text{-AXU}$$

Thanks to Mridul Nandi for pointing out this implicit assumption for F in our inital version

Requirements for F

Basic Assumption (F is AXU)

$$F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n \text{ is } \epsilon\text{-AXU}$$

***Further* Assumption (Cascade F^b is AXU)**

$$F^b_\kappa(X) := F_\kappa(\ldots(F_\kappa(X_1) \oplus X_2), \ldots) \oplus X_b) \text{ is } b \cdot \epsilon\text{-AXU}$$

Thanks to Mridul Nandi for pointing out this implicit assumption for F in our inital version

Nandi will give your more details about this in the next talk :-)

Recommended Instantiations of F

Primary Recommendation: **4-Round-AES**

- $10 + 4 + 4 = 18$ AES rounds/block
- ϵ-AXU with $\epsilon \approx 2^{-113}$ [Daemen et al. 09]

Recommended Instantiations of F

Primary Recommendation: **4-Round-AES**

- $10 + 4 + 4 = 18$ AES rounds/block
- ϵ-AXU with $\epsilon \approx 2^{-113}$ [Daemen et al. 09]

Secondary Recommendation: **10-Round-AES** (**Full-AES**)

- $3 \cdot 10 = 30$ AES rounds/block
- Full AES should be 2^{-128}-AXU

Recommended Instantiations of F

Primary Recommendation: **4-Round-AES**

- $10 + 4 + 4 = 18$ AES rounds/block
- ϵ-AXU with $\epsilon \approx 2^{-113}$ [Daemen et al. 09]

Secondary Recommendation: **10-Round-AES** (**Full-AES**)

- $3 \cdot 10 = 30$ AES rounds/block
- Full AES should be 2^{-128}-AXU

Withdrawn Recommendation: **GF-128 multiplication**

Reason: *Weak-Key Analysis of POET*

Abdelraheem, Bogdanov and Tischhauser applied the observations of Cid and Procter [CidP13] to POET

`https://eprint.iacr.org/2014/226`

Software Performance

- Software performance with Full-AES [Bogdanov et al. 14]
  - Single message scenario: 4.62 cpb
  - Multi message scenario: 2.75 cpb

Software Performance

- Software performance with Full-AES [Bogdanov et al. 14]
  - Single message scenario: 4.62 cpb
  - Multi message scenario: 2.75 cpb

- *Estimated* software performance with 4-AES
  - Single message scenario: $(18/30) \cdot 4.62$ cpb ≈ 2.77 cpb
  - Multi message scenario: $(18/30) \cdot 2.75$ cpb $= 1.65$ cpb

Software Performance

- Software performance with Full-AES [Bogdanov et al. 14]
  - Single message scenario: 4.62 cpb
  - Multi message scenario: 2.75 cpb

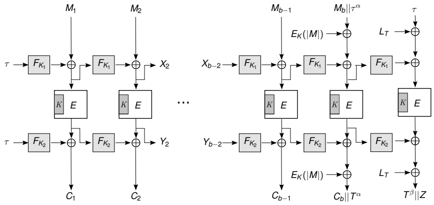
- *Estimated* software performance with 4-AES
  - Single message scenario: $(18/30) \cdot 4.62$ cpb ≈ 2.77 cpb
  - Multi message scenario: $(18/30) \cdot 2.75$ cpb $= 1.65$ cpb

We are looking for developers for high speed implementations
(`https://github.com/cforler/poet`)

Section 3

Security of POET

POET: Security

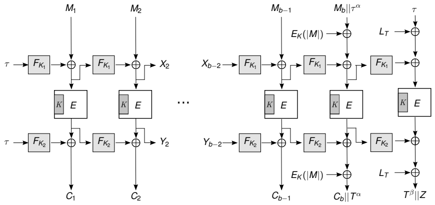


Birthday bound security

- POET is CCA3 secure against *nonce-respecting* adversaries

$$\mathbf{Adv}_{\Pi}^{\text{CCA3}}(q,\ell,t) \leq \mathbf{Adv}_{\Pi}^{\text{IND-CPA}}(q,\ell,t')+\mathbf{Adv}_{\Pi}^{\text{INT-CTXT}}(q,\ell,t'') \quad (*)$$

POET: Security

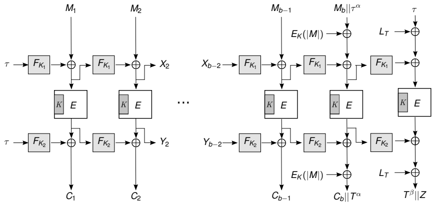


Birthday bound security

- POET is CCA3 secure against *nonce-respecting* adversaries

$$\mathbf{Adv}^{\text{CCA3}}_{\Pi}(q,\ell,t) \leq \mathbf{Adv}^{\text{IND-CCA}}_{\Pi}(q,\ell,t') + \mathbf{Adv}^{\text{INT-CTXT}}_{\Pi}(q,\ell,t'') \quad (*)$$

POET: Security



Birthday bound security

- POET is CCA3 secure against *nonce-respecting* adversaries

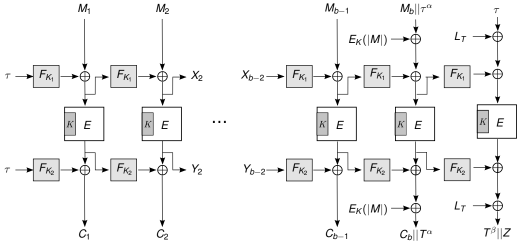
$$\mathbf{Adv}_{\Pi}^{\text{CCA3}}(q,\ell,t) \leq \mathbf{Adv}_{\Pi}^{\text{IND-CCA}}(q,\ell,t') + \mathbf{Adv}_{\Pi}^{\text{INT-CTXT}}(q,\ell,t'') \quad (*)$$

- POET is OCCA3 secure against *nonce-ignoring* adversaries

$$\mathbf{Adv}_{\Pi}^{\text{OCCA3}}(q,\ell,t) \leq \mathbf{Adv}_{\Pi}^{\text{OPRP-CCA}}(q,\ell,t') + \mathbf{Adv}_{\Pi}^{\text{INT-CTXT}}(q,\ell,t'') \quad (*)$$

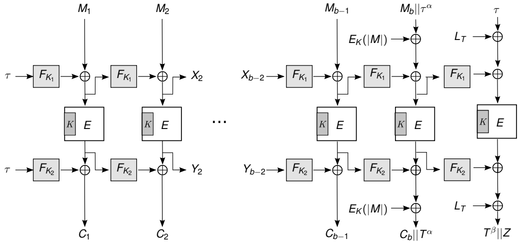
$(*) t', t'' \in O(t)$

POET: OPRP-CCA-Security



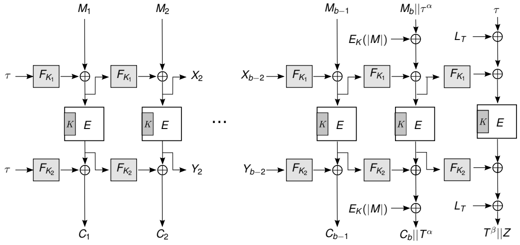
- $\mathcal{A}$ instantly wins if a **bad event** occurs

POET: OPRP-CCA-Security



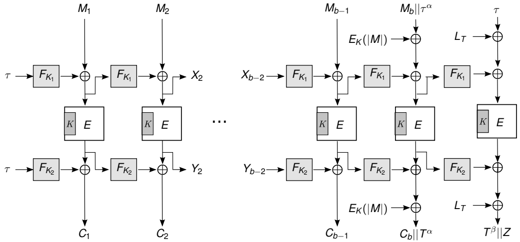
- $\mathcal{A}$ instantly wins if a **bad event** occurs
  1. $\mathcal{A}$ can distinguish E from random permutation

POET: OPRP-CCA-Security



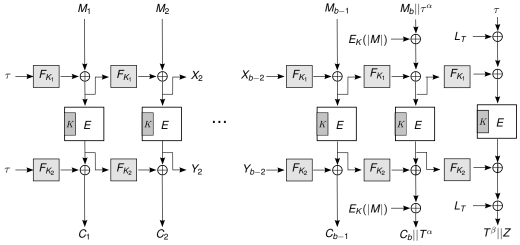
- $\mathcal{A}$ instantly wins if a **bad event** occurs
  1. $\mathcal{A}$ can distinguish E from random permutation
  2. Header collison (Pr[$COLL^{ad}$])

POET: OPRP-CCA-Security



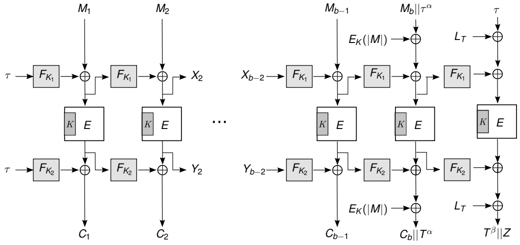
- $\mathcal{A}$ instantly wins if a **bad event** occurs
  1. $\mathcal{A}$ can distinguish E from random permutation
  2. Header collison (Pr[$COLL^{ad}$])
  3. Top row collison (Pr[$COLL^{top}$])

POET: OPRP-CCA-Security



- $\mathcal{A}$ instantly wins if a **bad event** occurs
  1. $\mathcal{A}$ can distinguish E from random permutation
  2. Header collison (Pr[$COLL^{ad}$])
  3. Top row collison (Pr[$COLL^{top}$])
  4. Bottom row collison (Pr[$COLL^{bot}$])

POET: OPRP-CCA-Security



- $\mathcal{A}$ instantly wins if a **bad event** occurs
  1. $\mathcal{A}$ can distinguish E from random permutation
  2. Header collison (Pr[$COLL^{ad}$])
  3. Top row collison (Pr[$COLL^{top}$])
  4. Bottom row collison (Pr[$COLL^{bot}$])
- $\mathcal{A}$ can distinguish POET without a collison (Pr[$NOCOLL$])

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}^{\text{IND-SPRP}}_{E,E^{-1}}(\ell + 2q, O(t))$

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}^{\text{IND-SPRP}}_{E,E^{-1}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either
     - Collision with a final message block: $\approx \ell^2 \epsilon$

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}^{\text{IND-SPRP}}_{E,E^{-1}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either
     - Collision with a final message block: $\approx \ell^2\epsilon$
     - Collision between non final mesage blocks: $\leq \ell^2\epsilon/2$

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either
     - Collision with a final message block: $\approx \ell^2\epsilon$
     - Collision between non final mesage blocks: $\leq \ell^2\epsilon/2$
  4. Collision in bottom row (see 3.)

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either
     - Collision with a final message block: $\approx \ell^2\epsilon$
     - Collision between non final mesage blocks: $\leq \ell^2\epsilon/2$
  4. Collision in bottom row (see 3.)
- Pr[*NOCOLL*] can be upper bound by $9 \cdot \ell^2/(2^n - 3\ell)$

POET: OPRP-CCA-Security

- Upper bounds for the four **bad events**
  1. Assume E is secure: $\mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$
  2. Upper bound for header collison: $\ell^2/2^n$
  3. Top row collision implies either
     - Collision with a final message block: $\approx \ell^2\epsilon$
     - Collision between non final mesage blocks: $\leq \ell^2\epsilon/2$
  4. Collision in bottom row (see 3.)
- $\Pr[NOCOLL]$ can be upper bound by $9 \cdot \ell^2/(2^n - 3\ell)$

$$\mathbf{Adv}_{\Pi}^{\text{OPRP-CCA}}(q, \ell, t) \leq 4\ell^2\epsilon + \frac{9\ell^2}{2^n - 3\ell} + \mathbf{Adv}_{E,E^{-1}}^{\text{IND-SPRP}}(\ell + 2q, O(t))$$

POET: INT-CTXT-Security

- INT-CTXT proof is game-based
- Combines the ideas from its OPRP-CCA proof and the INT-CTXT proof from McOE
- Details ($\rightarrow$ CAESAR submission)

INT-CTXT Advantage

$$\mathbf{Adv}_{\mathrm{POET}}^{\mathrm{INT\text{-}CTXT}}(q,\ell,t) \leq (\ell + 2q)^2/2^n + \frac{q}{2^n - q} + \mathbf{Adv}_{\Pi}^{\mathrm{OPRP\text{-}CCA}}(q,\ell,t)$$

Restated Security Claims

	Bits of Security
Confidentiality for the plaintext	$\log_2(2^{128} - c \cdot \epsilon \cdot \ell^2)$
Integrity for the plaintext	$\log_2(2^{128} - c \cdot \epsilon \cdot \ell^2)$
Integrity for the associated data	$\log_2(2^{128} - c \cdot \epsilon \cdot \ell^2)$
Integrity for the public message number	$\log_2(2^{128} - c \cdot \epsilon \cdot \ell^2)$
Security against key recovery	128
Security against tag guessing	128

Yu Sasaki pointed out that *our stated security claims had been confusing*

Conclusion

POET

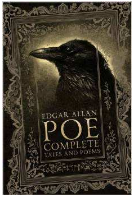
- is non-sequential and on-line
- support for intermediate tags
- is robust against nonce- and decryption-misuse (OCCA3-secure = OPRP-CCA + INT-CTXT)
- fulfills the demanding requirements of high-speed networks

Conclusion

POET

- is non-sequential and on-line
- support for intermediate tags
- is robust against nonce- and decryption-misuse (OCCA3-secure = OPRP-CCA + INT-CTXT)
- fulfills the demanding requirements of high-speed networks

Final Remark: Cryptanalysis, fruitful remarks and third party implementation etc. will be rewarded!



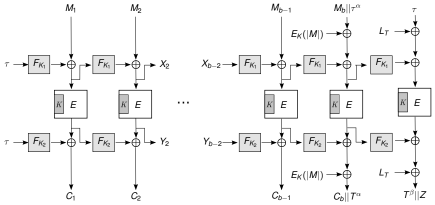
The End

Thank you for your attention!

POET Homepage

`http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/research/poet/`

Key Derivation



- POET needs five 128-bit keys: K, K_1, and K_2, L, and L_T
- They are derived from a 128 bit master key SK

$$K = E_{SK}(0), \qquad L = E_{SK}(1)$$
$$K_1 = E_{SK}(2) \qquad K_2 = E_{SK}(3),$$
$$L_T = E_{SK}(4)$$

(currently I am analysing the case: $K_1 = K_2$ and $L_T = 7L$)