New Results and Insighs on ForkAE

Elena Andreeva[1] Arne Deprez[2] Jowan Pittevils[2]
Arnab Roy[1] Amit Singh Bhati[2] Damian Vizár[5]

Alpen-Adria University Klagenfurt, Austria

imec-COSIC, KU Leuven, Belgium

CSEM, Switzerland

NIST LWC workshop 2020

0. ForkAE: recap
1. Cryptanalysis of ForkSkinny
2. Implementation results
3. SAEF: security update
4. Extending the use case

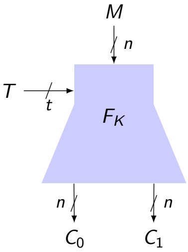
$+$ New forkcipher encryption modes

ForkAE: Forkcipher

$\approx$Two parallel TBC calls at lower cost

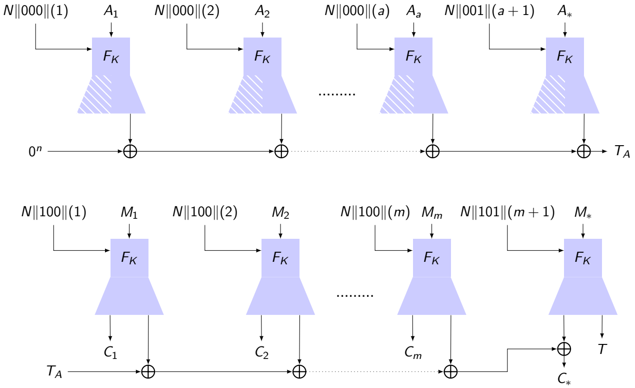
iterate-fork-iterate the well-cryptanalyzed SKINNY components

$\Rightarrow$ $(r_{\text{init}}, r_0, r_1)$ configuration with $r_0 = r_1$



Primitive F	n	t	$t + \|K\|$
ForkSkinny-64-192	64	64	192
ForkSkinny-128-192	128	64	192
ForkSkinny-128-256	128	128	256
ForkSkinny-128-288	128	128	288

ForkAE: PAEF

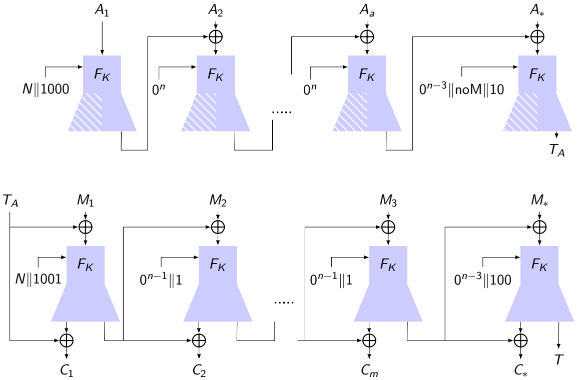


n-bit AE security

$$Adv_{PAEF}^{privacy}(\mathcal{A}) \leq Adv_{F}^{PRFP}(\mathcal{D})$$

$$Adv_{PAEF}^{auth}(\mathcal{A}) \leq Adv_{F}^{PRFP}(\mathcal{D}) + \frac{q_v \cdot 2^n}{(2^n - 1)^2}$$

ForkAE: SAEF



$n/2$-bit AE security

$$Adv_{SAEF}^{privacy}(\mathcal{A}) \leq Adv_F^{PRFP}(\mathcal{D}) + 2\frac{(\sigma - q)^2}{2^n}$$

$$Adv_{SAEF}^{auth}(\mathcal{A}) \leq Adv_F^{PRFP}(\mathcal{D}) + \frac{2(\sigma - q + 1)^2}{2^n} + \frac{\sigma(\sigma - q)}{2^n} + \frac{q_v(q+2)}{2^n}$$

Cryptanalysis

Status of ForkSkinny

- No weakness till date from publicly known cryptanalysis
- It continues to benefit from the security margin of SKINNY
- The best attack on SKINNY covers $\approx 50\%$ of the total nr of rounds

Cryptanalysis

Status of ForkSkinny

- No weakness till date from publicly known cryptanalysis
- It continues to benefit from the security margin of SKINNY
- The best attack on SKINNY covers $\approx 50\%$ of the total nr of rounds

ForkSkinny cryptanalysis (Bariant et al. ToSC 2020)

- ForkSkinny-128-256 (128-bit tweak, 128-bit key): 24 out of 48 rounds
- ForkSkinny-128-256 (no tweak): 26 rounds attacked
  - ✓ Not part of the ForkAE family

Cryptanalysis

General cryptanalysis (of forkcipher)

- ForkSkinny does not have the weaknesses of ForkAES
  - ✓ Reconstruction queries: a specific of forkciphers
  - ✓ ForkAES had a weakness wrt to these, cryptanalysis exploited it
  - ✓ ForkSkinny **does not have** such reconstruction query weakness

Cryptanalysis

General cryptanalysis (of forkcipher)

- ForkSkinny does not have the weaknesses of ForkAES
  - ✓ Reconstruction queries: a specific of forkciphers
  - ✓ ForkAES had a weakness wrt to these, cryptanalysis exploited it
  - ✓ ForkSkinny **does not have** such reconstruction query weakness

Remarks

- Reduced round instances should have $r_0 = r_1$
- ForkSkinny has comfortable secuirty margin
  - ✓ The nr of rounds can be reduced by ≥ 5, i.e. $r_0 = r_1 = 26$.
  - ✓ We are currently exploring further reduction

Portable SW implementations

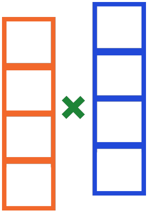
- We started with: constant-time implementations at https://github.com/rweather/lightweight-crypto

- Improved decryption with **preprocessed TKS**:
  - ✓ 38% less clock cycles
  - ✓ 1kB smaller ROM size
  - ✓ 252-696 bytes higher RAM usage

Table-based SW implementations

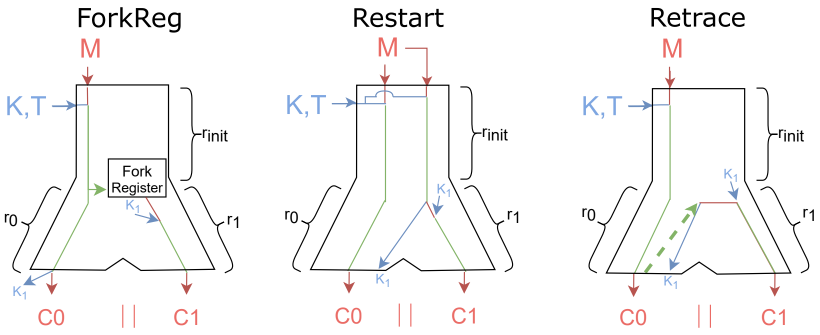
- Suitable for platforms without a cache, e.g. Cortex-M0
- Round function → 18 lookups + 19 XOR
- Performance on Cortex-M0 (wrt our portable implementations):
  - ✓ Enc / Dec up to 20% / 25% faster
  - ✓ Increased memory use: 4 tables of 1kB each
  - ✓ Memory overhead decrease: store 1 table with slight loss of performance

Neon SIMD SW implementations



- Implementation for Neon SIMD on Arm Cortex-A9
- 128-bit instances (S-box in parallel in a single branch):
  - ✓ 30% less clock cycles
  - ✓ 0.5 kB reduction in ROM size
  - ✓ RAM size equal
- 64-bit instance (S-box in both branches in parallel):
  - ✓ 29 % less clock cycles
  - ✓ ROM size approx. equal
  - ✓ RAM size increased

Low-area ForkSkinny HW architectures



Word-based architectures results

ForkReg

- Enc and Dec
- Best speedup
- 1.09-1.25 area of Skinny
- up to 129% throughput of Skinny

Restart

- Encryption only
- Best area
- 0.97-1.11 area of Skinny
- up to 79% throughput of Skinny

Retrace

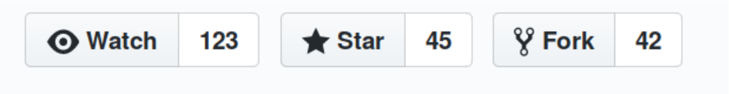
- Enc and Dec
- Goldilocks zone
- 0.93-1.04 area of Skinny
- up to 126% throughput of Skinny

results obtained w/ NanGate 45NM library, no clock gating or latches, datapath sizes of 1/16 block size

More about implementations

- SW implementations
  - ✓ A. Deprez Master Thesis 2020, "Optimized software implementations for ForkAE"
  - ✓ Check https://github.com/byt3bit/forkae
  - ✓ Updated results will be presented at CARDIS 2020
  - ✓ Implementations benchmarked at https://lwc.las3.de/
- HW implementations
  - ✓ J. Pittevils Master Thesis 2020, "Low-area Optimized Hardware Implementations for ForkAE"
- Questions to antoon.purnal@kuleuven.be

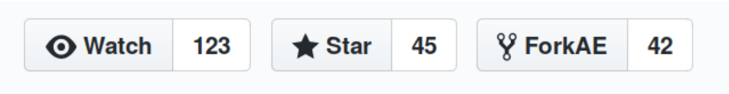
https://github.com/byt3bit/forkae/



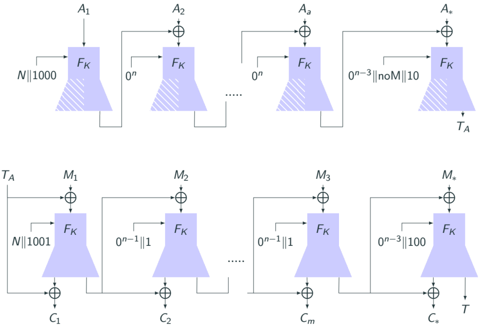
More about implementations

- SW implementations
  - ✓ A. Deprez Master Thesis 2020, "Optimized software implementations for ForkAE"
  - ✓ Check https://github.com/byt3bit/forkae
  - ✓ Updated results will be presented at CARDIS 2020
  - ✓ Implementations benchmarked at https://lwc.las3.de/
- HW implementations
  - ✓ J. Pittevils Master Thesis 2020, "Low-area Optimized Hardware Implementations for ForkAE"
- Questions to antoon.purnal@kuleuven.be

https://github.com/byt3bit/forkae/



SAEF: Security

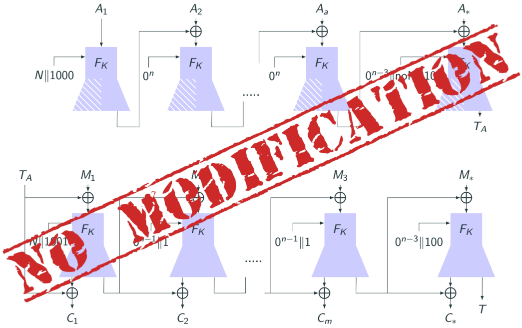


$n/2$-bit nonce-based AE security

$$Adv^{\text{priv}}_{SAEF}(\mathcal{A}) \leq Adv^{\text{PRFP}}_{F}(\mathcal{D}) + 2\frac{(\sigma - q)^2}{2^n}$$

$$Adv^{\text{auth}}_{SAEF}(\mathcal{A}) \leq Adv^{\text{PRFP}}_{F}(\mathcal{D}) + \frac{2(\sigma - q + 1)^2}{2^n} + \frac{\sigma(\sigma - q)}{2^n} + \frac{q_v(q+2)}{2^n}$$

SAEF: Security



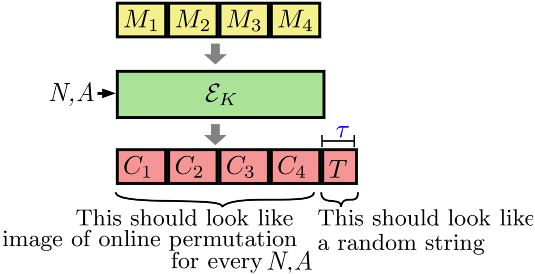
$n/2$-bit OAE security [ASV, SAC 2020]

$$Adv_{SAEF}^{\text{oprp}\|\text{prf}}(\mathcal{A}) \leq Adv_F^{\text{PRFP}}(\mathcal{D}) + \frac{3 \cdot \sigma^2}{2^{n+1}}$$

$$Adv_{SAEF}^{\text{mr-auth}}(\mathcal{A}) \leq Adv_F^{\text{PRFP}}(\mathcal{D}) + \frac{\sigma^2 + 4 \cdot q_v}{2^n}$$

OAE Security
[Fleischman,Forler,Lucks 12]

Against attacker **repeating (i.e, misusing) nonces**:



Privacy

OPerm[n] + random tag

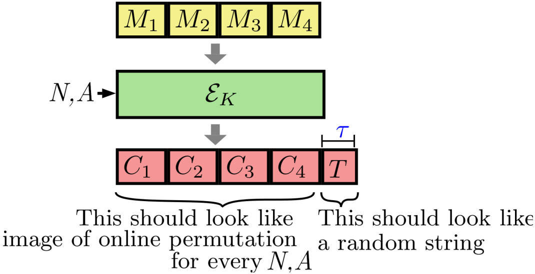
+

Authenticity

Unforgeability

OAE Security
[Fleischman,Forler,Lucks 12]

Against attacker **repeating (i.e, misusing) nonces**:



Privacy

OPerm[n] + random tag

+

Authenticity

Unforgeability

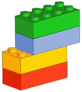
⇒ Leaks length of common n-aligned prefix of plaintexts if N, A repeat

⇒ Forging is as hard as with unique nonces

SAEF: Implications

Efficient OAE

✓ e.g. 0.8 complexity of COLM-SKINNY



Safe for blockwise (adaptive) processing [E**V**, FSE 2017]

✓ Constrained environment (latency, limited memory, ...)

Security under nonce misuse

✓ Integrity undamaged

✓ Well-defined privacy level

SAEF: Case studies

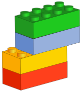
Nonce misuse in Lightweight applications

- ✓ Cheap HW platforms, forced resets, fault attacks etc
- ✓ Chosen Prefix, Secret Suffix attack on OAE (HTTPS) [HRR**V** 15]
- ✓ Possibly chosen prefix constant length $\Rightarrow$ CPSS shut down (MQTT)
- ✓ OAE-secure AE is a good, pragmatic solution

SAEF: Case studies

Nonce misuse in Lightweight applications

- ✓ Cheap HW platforms, forced resets, fault attacks etc
- ✓ Chosen Prefix, Secret Suffix attack on OAE (HTTPS) [HRR**V** 15]
- ✓ Possibly chosen prefix constant length ⇒ CPSS shut down (MQTT)
- ✓ OAE-secure AE is a good, pragmatic solution

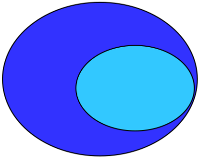


Blockwise encryption

- ✓ Large data (temp. firmware image, graphics assets, maps etc) often on ext. flash
- ✓ Blockwise encryption typically unavoidable
- ✓ OAE-secure AE is safe to use

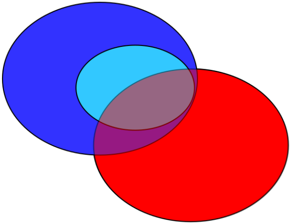
Extending the use case

ForkAE: an efficient candidate for lightweight applications, especially with predominantly short messages



Extending the use case

ForkAE: an efficient candidate for lightweight applications, especially with predominantly short messages



but also for defense in depth, offering the interesting combination of light weight and robustness.

Efficient encryption with Forkcipher

- **Generalized counter mode (GCTR)**
  - ✓ random IV AND/OR nonce
  - ✓ tweakable forkcipher
  - ✓ many ways to generate tweak/block input
  - ✓ direct use (encryption only)
  - ✓ as a component (such as in Deoxys II)
- **Systematic study of GCTR variants** [under submission]
  - ✓ high efficiency, up to BBB security
  - ✓ stay tuned!

Thank you!

damian.vizar@csem.ch