Efficient and Provable White-Box Primitives

Pierre-Alain Fouque[✧] **Pierre Karpman**[🙙] Paul Kirchner[∾]
Brice Minaud[✿]

[✧]Université de Rennes 1 and Institut universitaire de France
[🙙]Inria, École polytechnique, NTU and CWI
[∾]École normale supérieure
[✿]Université de Rennes 1 and Royal Holloway University of London

ASIACRYPT, Hanoi
2016–12–05

Context

Provably secure white-box primitives

Implementation aspects

Motivation: incompressibility

Informally:

- Let $\mathcal{E} : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ be a block cipher
- Let $\mathbb{E} \hookleftarrow \mathcal{E}$ be an incompressible implementation of $\mathcal{E}$
- Given only $\mathbb{E}$, it must be hard to find $\mathbb{E}'$ s.t.
  1. $\forall k \in \mathcal{K}, \forall m \in \mathcal{P}, \mathbb{E}'(k, m) = \mathbb{E}(k, m)$
  2. $\#(\mathbb{E}') \ll \#(\mathbb{E})$

Explicit (relaxed) targets:

- $\mathbb{E}(k, m) = \mathbb{E}'(k, m)$ for a proportion α of inputs
- $\#(\mathbb{E}') < c \cdot \#(\mathbb{E})$

White-box encryption schemes

White-box encryption scheme

A pair of two encryption schemes

$$\mathcal{E} : \mathcal{K} \times \mathcal{K}' \times \mathcal{R} \times \mathcal{P} \to \mathcal{C}$$
$$\mathbb{E} : \mathcal{T} \times \mathcal{K}' \times \mathcal{R} \times \mathcal{P} \to \mathcal{C}$$

with a *white-box compiler* $\mathsf{C} : \mathcal{K} \to \mathcal{T}$ s.t.:

$$\forall k \in \mathcal{K}, \mathcal{E}(k, \cdot, \cdot, \cdot) \equiv \mathbb{E}(\mathsf{C}(k), \cdot, \cdot, \cdot)$$

- Take $\#\mathcal{K} \ll \#\mathcal{T}$
- $T \in \mathcal{T} \approx$ "pseudorandom tables" generated from $k \in \mathcal{K}$
  - ASASA, SPACE, SPNbox, this work

Black and white adversaries

Black-box attacks:

- Pick k, attack $\mathcal{E}(k, \cdot, \cdot, \cdot)$ as a symmetric cryptosystem

White-box attacks:

- Given $\mathbb{E}(\mathsf{C}(k), \cdot, \cdot, \cdot)$, find equivalent smaller $\mathbb{E}'$
  - *Compiler adversary:* extract k from $\mathsf{C}(k)$
  - *Implementation adversary:* use less of $\mathsf{C}(k)$ while maintaining functionality

White-box security

Protecting against compiler adversaries:

- Build the tables as secure small block ciphers
  - ASASA (Biryukov et al., 2014), broken (Minaud et al., 2015), (Dinur et al., 2015)
  - SPNbox (Bogdanov et al., 2016)
- Build on a normal-sized strong cipher (e.g. the AES)
  - SPACE (Bogdanov and Isobe, 2015)
  - Also this work

Protecting against implementation adversaries:

- Force many unpredictable table accesses when running $\mathbb{E}$

Context

Provably secure white-box primitives

Implementation aspects

The objective

Design white-box encryption schemes:

- With provable arguments v. all black and white-box adversaries
- With easily tunable parameters (implementation size, security)

Focus on the necessary primitives:

- White-box block ciphers $\Rightarrow$ the PuppyCipher family
- White-box key generators $\Rightarrow$ the CoureurDesBois family

Global strategy

1. Rely on the AES to defeat black-box adversaries
2. " to defeat compiler adversaries
3. Define a security model w.r.t. implementation adversaries
4. Use it to prove security bounds for the constructions

Getting rid of the easy adversaries

Black-box adversaries:

- Use "black-box calls" to the AES as part of the scheme
  - Example: $\hat{\mathbb{E}} = \mathsf{AES}_{k''} \circ \mathbb{E} \circ \mathsf{AES}_{k'}$
  - Happens naturally for our constructions, e.g. PuppyCipher

Compiler adversaries:

- Define $\mathsf{C}(k)$ from the AES with key k
  - Example: $\mathsf{C}(k) = [\mathsf{AES}_k(0^{112}||i)]$, $0 \leq i < 2^{16}$

A model for (weak) incompressibility

For a table-based scheme $\mathbb{E} : \mathcal{T} \times \mathcal{K}' \times \mathcal{R} \times \mathcal{P} \to \mathcal{C}$:

ENC-TCOM (weak incompressibility)

Security parameters: s, λ, δ

B picks T from $\mathcal{T}$ uniformly at random
A adaptively queries $T[q_i]$, $0 \leq i < s$
B picks (K', R, P) from $\mathcal{K}' \times \mathcal{R} \times \mathcal{P}$ uniformly at random
A wins by providing $C = \mathbb{E}(T, K', R, P)$

$\mathbb{E}$ is (s, λ, δ)-secure if with $\Pr = 1 - 2^{-\lambda}$ over the choice of T, **A** wins with $\Pr < \delta$

Remarks on ENC-TCOM

Source of "weakness":

- Assumption on the adversarial strategy

Strong variant (sketch):

- **A** chooses a leak function f guaranteeing

$$\text{min-entropy}(x|f(x)) > \mu$$

- **B** picks T, sends $f(T)$ to **A**
- **A** tries to encrypt a random message

The CoureurDesBois family

Objective: A family $\mathsf{CDB}-t : \mathcal{R} \to \mathcal{K}$

- Can be used for key generation in a hybrid system
- Tunable implementation size parameter t
- Provably secure w.r.t. ENC-TCOM

A simple structure

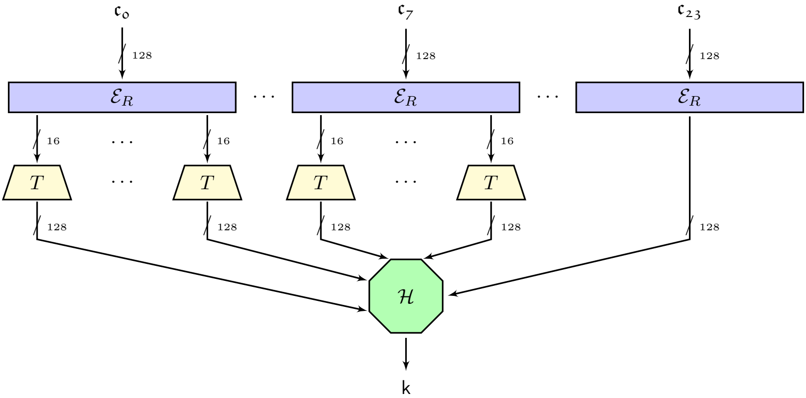
Compilation phase:

- $T = \mathrm{C}(k) = [\mathrm{AES}_k(0^{128-t}||i)]$, $0 \leq i < 2^t$
- T has size 2^{t+4} bytes

Use the random input r to CDB $-t$ to:

1. Generate a pseudorandom sequence (S_i) of n t-bit values (use AES-CTR)
2. Access T at indices $S_0, \ldots, S_{n-1}$
3. Arrange the outputs in a matrix $Q \in \mathcal{M}_d(\mathbb{F}_{2^{128}})$, $d = \lceil \sqrt{n} \rceil$
4. Generate $a, b \in \mathbb{F}_{2^{128}}^d$ (use AES-CTR)
5. The result is $\mathbf{k} = \sum_{i,j} Q_{i,j} \cdot a_i \cdot b_j$ (extractor from Coron et al., 2011)

CoureurDesBois-16 in a picture



How many table accesses are necessary?

Idea: **A** cannot predict **k** if it doesn't know $T[x]$ for some x

- Let **A** keep s table outputs (ratio $\alpha := s/\#T$)
- What should be n for **A** to miss at least one T input w.h.p.?

Security target: $\delta = 128 - \log(s) \approx 128 - t$ bits

- **A** could store s random values **k** instead

A generic lower bound:

- We need at least r rounds with $\alpha^r \leq 2^{-\delta}$

The result

- One more round than the generic lower bound is enough
- See the paper for details

Example: $\alpha = 2^{-2}$

- CDB -16: 57 table accesses ($\delta = 112$)
- CDB -20: 55 table accesses ($\delta = 108$)
- CDB -24: 53 table accesses ($\delta = 104$)

There is more

CoureurDesBois can also be proven secure in the strong model

- Exploit similarity of incompressibility and bounded-storage models
- Use results from Vadhan on local extractors (2004)

The PuppyCipher family

Objective: A family $\mathsf{PC}-t : \mathcal{K}' \times \mathcal{P} \to \mathcal{C}$

- Take $\mathcal{K}' = \mathcal{P} = \mathcal{C} = \{0,1\}^{128}$ (typical block cipher sizes)
- Tunable implementation size parameter t
- Provably secure w.r.t. ENC-TCOM
- Can be seen as a sequential variant of CoureurDesBois

A simple structure (again)

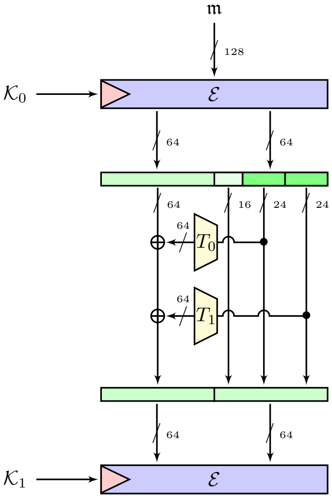
Compilation phase:

- $T_{u=0,\ldots,64/t-1} = \mathsf{C}(k) = [\lfloor \mathsf{AES}_k(K_u||i) \rfloor_{64}],\ 0 \leq i < 2^t$
- $\{T_u\}$ has size $(64/t - 1) \times 2^{t+3}$ bytes

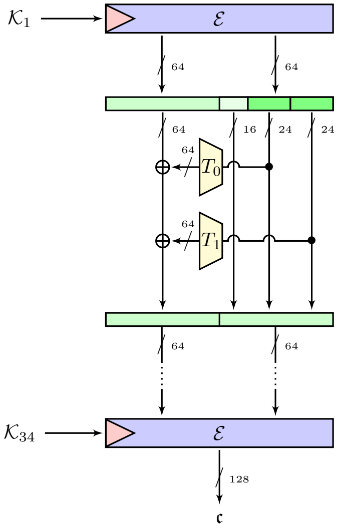
Encryption phase:

- Round function: one Feistel step + one AES call
- $(m_L||(m_{R1}||m_{R2})) \mapsto \mathsf{AES}((m_L \oplus T_0(m_{R1}) \oplus T_1(m_{R2})||m_R))$

PuppyCipher-24 in a picture (top)



PuppyCipher-24 in a picture (bottom)



How many table accesses are necessary?

- Proof idea similar to CoureurDesBois (weak model)
- More intricate because of non-independence of inputs
- See the paper for details

Example: $\alpha = 2^{-2}$

- PC -16: 18 rounds / 72 table accesses ($\delta = 112$)
- PC -20: 23 rounds / 69 table accesses ($\delta = 108$)
- PC -24: 34 rounds / 68 table accesses ($\delta = 104$)

Context

Provably secure white-box primitives

Implementation aspects

Features of CDB and PC

- All individual components are efficient
- # Table access is near-minimal for a given security
- CoureurDesBois is highly parallelizable
- Some table accesses also parallel in PuppyCipher

- More aggressive variant of PuppyCipher: Hound
  - Use only 5-round AES after each Feistel step

Selected implementation figures: PC

Execution time in cycles / one block / Xeon E5-1603v3

	Size (bytes)	Avg.	Std. Dev.
PC-16 (white-box)	2^{21}	2800	70
PC-16 (secret)	negl.	3940	10
PC-24 (white-box)	2^{28}	23390	1340
PC-24 (secret)	negl.	6600	60
HD-24 (white-box)	2^{28}	21740	1230
HD-24 (secret)	negl.	5360	60

- 175 to 1460 cycles/byte

Selected implementation figures: CDB

Execution time in cycles / one call / Xeon E5-1603v3

	Size (bytes)	Avg.	Std. Dev.
CDB-16 (white-box)	2^{20}	2020	20
CDB-16 (secret)	negl.	2150	30
CDB-20 (white-box)	2^{24}	4700	600
CDB-20 (secret)	negl.	2900	20
CDB-24 (white-box)	2^{28}	11900	610
CDB-24 (secret)	negl.	3050	30

- $\approx 1.4 - 2.4\times$ faster than PuppyCipher/Hound

Performance as $\# \equiv$ sequential table calls

A single table access for PC-24 costs 490 cycles in our tests (beware of the variance!)

- PC-24: $\equiv$ 48 sequential accesses (v. 68 real)
- CDB-24: $\equiv$ 25 sequential accesses (v. 53 real)

A single table access for PC-16 costs 59 cycles in our tests

- PC-16: $\equiv$ 47 sequential accesses (v. 72 real)
- CDB-16: $\equiv$ 35 sequential accesses (v. 57 real)

(Lack of) comparison with SPACE & SPNbox

PuppyCipher v. Hound v. CoureurDesBois v. SPACE v. SPNbox

- Meaningful comparison from existing data is hard
  - Unequal security level, different message sizes, different systems
- $\Rightarrow$ No attempts to summarize a comparison here

Fin!