

# Using KECCAK technology for AE: KETJE, KEYAK and more

Guido BERTONI<sup>1</sup>   Joan DAEMEN<sup>1</sup>  
Michaël PEETERS<sup>2</sup>   Gilles VAN ASSCHE<sup>1</sup>   Ronny VAN KEER<sup>1</sup>

<sup>1</sup>STMicroelectronics

<sup>2</sup>NXP Semiconductors

SHA-3 2014 Workshop  
UC Santa Barbara, August 22, 2014

# Outline

- 1 Secure channel protocols
- 2 Preferred KECCAK-based AE: KEYAK
- 3 Scaling it down: KETJE
- 4 Nonceless AE: HADDOC
- 5 Robust AE: MR. MONSTER BURRITO

# Secure channels [Patterson, Crypto summer school Croatia 2014 ]

Secure channels widely deployed:

- SSL/TLS, DTLS, IPsec, SSH, OpenVPN
- WEP, WPA, WPA2
- GSM, UMTS, LTE
- Cryptocat, OTR, SilentCircle
- ISO 7816-4, Global Platform SCP 01, 02, 03, 11, ...
- Bluetooth, DECT, ...
- ...

# Desired security [Patterson, Crypto summer school Croatia 2014]

## Message confidentiality, integrity and origin authentication

- Channel and endpoint characteristics
  - stream-based or message-oriented
  - single-direction, half-duplex, full-duplex
  - with(-out) handling of errors: lost packets, noise, ...
  - side-channel leakage, faults
  - release of unverified plaintext, ...
- Against different types of attackers
  - passive: see ciphertexts and (some) plaintexts
  - active: can modify, delete, inject and re-order messages
- Additional protection:
  - semantic and multi-message: replay, message dropping, re-ordering
  - time-related: freshness
  - surveillance-related: traffic analysis
  - denial-of-service

# Desired security [Patterson, Crypto summer school Croatia 2014]

## Message confidentiality, integrity and origin authentication

### ■ Channel and endpoint characteristics

- stream-based or message-oriented
- single-direction, half-duplex, full-duplex
- with(-out) handling of errors: lost packets, noise, ...
- side-channel leakage, faults
- release of unverified plaintext, ...

### ■ Against different types of attackers

- passive: see ciphertexts and (some) plaintexts
- active: can modify, delete, inject and re-order messages

### ■ Additional protection:

- semantic and multi-message: replay, message dropping, re-ordering
- time-related: freshness
- surveillance-related: traffic analysis
- denial-of-service

# Desired security [Patterson, Crypto summer school Croatia 2014]

## Message confidentiality, integrity and origin authentication

### ■ Channel and endpoint characteristics

- stream-based or message-oriented
- single-direction, half-duplex, full-duplex
- with(-out) handling of errors: lost packets, noise, ...
- side-channel leakage, faults
- release of unverified plaintext, ...

### ■ Against different types of attackers

- passive: see ciphertexts and (some) plaintexts
- active: can modify, delete, inject and re-order messages

### ■ Additional protection:

- semantic and multi-message: replay, message dropping, re-ordering
- time-related: freshness
- surveillance-related: traffic analysis
- denial-of-service

# Desired security [Patterson, Crypto summer school Croatia 2014]

## Message confidentiality, integrity and origin authentication

- Channel and endpoint characteristics
  - stream-based or message-oriented
  - single-direction, half-duplex, full-duplex
  - with(-out) handling of errors: lost packets, noise, ...
  - side-channel leakage, faults
  - release of unverified plaintext, ...
- Against different types of attackers
  - passive: see ciphertexts and (some) plaintexts
  - active: can modify, delete, inject and re-order messages
- Additional protection:
  - semantic and multi-message: replay, message dropping, re-ordering
  - time-related: freshness
  - surveillance-related: traffic analysis
  - denial-of-service

# Authenticated encryption in a secure channel

- AE is just a building block, security also relies on:
  - key establishment, key management, user interface, ...
- Good engineering: **build in security from the getgo**
  - leads to efficient and effective solutions
  - in most cases nonces are required to protect against replay
- Unfortunately not all is based on good engineering
  - successful products often proof-of-concepts gotten out of hand
  - security has to be bolted on afterwards
  - sometimes there is no more room for a nonce

KECCAK supports both nonce-based and nonceless AE!



# Authenticated encryption in a secure channel

- AE is just a building block, security also relies on:
  - key establishment, key management, user interface, ...
- Good engineering: **build in security from the getgo**
  - leads to efficient and effective solutions
  - in most cases nonces are required to protect against replay
- Unfortunately not all is based on good engineering
  - successful products often proof-of-concepts gotten out of hand
  - security has to be bolted on afterwards
  - sometimes there is no more room for a nonce

**KECCAK supports both nonce-based and nonceless AE!**

# What do we mean by KECCAK-based AE?

- Underlying primitive:  $\text{KECCAK-}p[b, n_r]$  [FIPS 202 draft, 2014]
  - sharing of software/hardware with SHA-3
  - security assurance thanks to public scrutiny since 2008
  - efficient side-channel protection
- Range of parameters larger than in SHA-3 hashes and XOFs:
  - widths:  $b \in \{200, 400, 800, 1600\}$
  - default number of rounds :  $n_r = 12$
  - in MONKEYDUPLEX and DONKEYSPONGE:  $n_r \in \{1, 2, 6, 12\}$
- Target security strength: 128 bits
  - including multi-target attacks

# KECCAK-based AE modes: two approaches

## KEYAK:

- official duplex and sponge
- strong permutation
  - 12 rounds
  - block-oriented
  - *hermetic*
- cryptanalysis
  - permutation-level

## KETJE et al.:

- MONKEYDUPLEX and DONKEYSPONGE
- (thin) round function
  - #rounds in phases
  - stream-oriented
  - relying on e.g. max DP
- cryptanalysis
  - construction + round function

# Outline

- 1 Secure channel protocols
- 2 Preferred KECCAK-based AE: KEYAK**
- 3 Scaling it down: KETJE
- 4 Nonceless AE: HADDOC
- 5 Robust AE: MR. MONSTER BURRITO

# Our CAESAR candidate KEYAK

- Nonce-based
- Sequence of header-body pairs
  - keeps state during the session
- Optionally parallelizable
- Using
  - KECCAK- $p[1600, n_r = 12]$
  - KECCAK- $p[800, n_r = 12]$

# KEYAK instances and efficiency

Name	Width $b$	Parallelism $P$
OCEAN KEYAK	1600	4
SEA KEYAK	1600	2
LAKE KEYAK	1600	1
RIVER KEYAK	800	1

## ■ Processing

- long messages: about 50 % of SHAKE128
- short messages: 24 rounds

## ■ Working memory footprint

- reasonable on high- and middle-end platforms
- **not ideal on constrained platforms**

# Outline

- 1 Secure channel protocols
- 2 Preferred KECCAK-based AE: KEYAK
- 3 Scaling it down: KETJE**
- 4 Nonceless AE: HADDOC
- 5 Robust AE: MR. MONSTER BURRITO

# Our CAESAR candidate KETJE

- Nonce-based
- Sequence of header-body pairs
- Small footprint
- Reduce footprint while keeping high security strength:
  - use KECCAK- $p[200, n_r]$  (96-bit security) or KECCAK- $p[400, n_r]$
  - MONKEYDUPLEX instead of duplex
  - more ad-hoc security
  - strong reliance on nonce uniqueness
- Target niche: secure channel protocol on secure chips
  - banking card, ID, (U)SIM, secure element, FIDO, etc.
  - secure chip has strictly incrementing counter



## KETJE instances

feature		KETJE JR	KETJE SR
state size		25 bytes	50 bytes
block size		2 bytes	4 bytes
processing		computational cost	
initialization	per session	12 rounds	12 rounds
wrapping	per block	1 round	1 round
8-byte tag comp.	per message	9 rounds	7 rounds

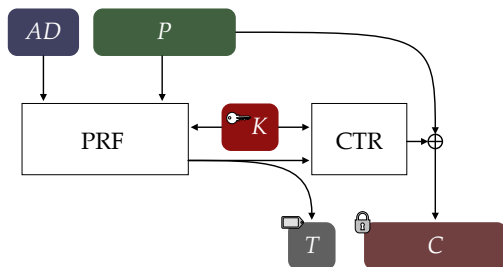
# KEYAK and KETJE in absence of nonces

- KEYAK may leak plaintext information
  - XOR of 1st differing plaintext blocks in otherwise identical sessions
- KETJE may break down completely
  - single-round differential attacks become possible

# Outline

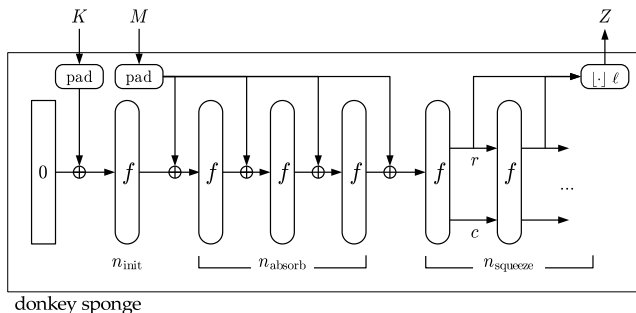
- 1 Secure channel protocols
- 2 Preferred KECCAK-based AE: KEYAK
- 3 Scaling it down: KETJE
- 4 Nonceless AE: HADDOC**
- 5 Robust AE: MR. MONSTER BURRITO

# HADDOC: the concept



- Permutation-based variant of SIV [Rogaway, Shrimpton 2006]
- Nonceless
- Leakage limited to:
  - length of messages
  - identical messages  $(AD, P)$  give identical ciphertexts  $(C, T)$

# Inside HADDOC: the DONKEYSPONGE PRF



- Absorbing phase exploits state secrecy [DR, Pelican, 2005]:
  - usage of full state width  $b$
  - $n_{init} = 2$ : make all state bits depend on the key
  - $n_{absorb} = 6$ : limit **max DP** to prevent state collisions
- Squeezing phase:  $n_{squeeze} = 12, c = 256$

# Building blocks of HADDOC

- PRF: DONKEYSPONGE with
  - input  $K$ : key
  - input  $M$ : injective coding of  $(AD, P)$
  - output  $T = \lfloor Z \rfloor_{256}$
- CTR: sponge in counter mode
  - single-block in- and outputs
  - $Z_i = \text{sponge}(K || T || i)$
  - $C_i = M_i \oplus Z_i$
  - $n_r = 12$
- Permutations
  - $\text{KECCAK-}p[1600, n_r]$
  - $\text{KECCAK-}p[800, n_r]$

# HADDOC features

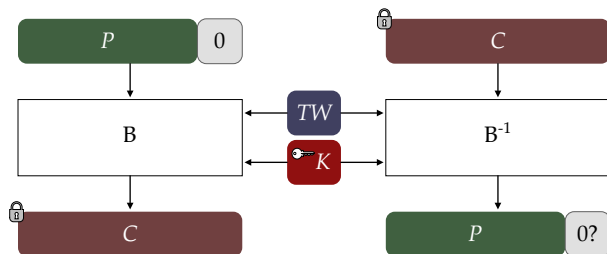
- Processing:
  - long messages: about 70 % of SHAKE128
  - short messages: 26 rounds
  - if  $P$  is absent we get a MAC function:
    - long messages: about 21 % of SHAKE128
    - short messages: 14 rounds
- Advantages
  - decryption: random access
  - encryption: PRF parallelizable
- Disadvantages
  - encryption strictly two-pass
  - message expansion by  $2n$ -bit tag for  $n$ -bit security

# Outline

- 1 Secure channel protocols
- 2 Preferred KECCAK-based AE: KEYAK
- 3 Scaling it down: KETJE
- 4 Nonceless AE: HADDOC
- 5 Robust AE: MR. MONSTER BURRITO

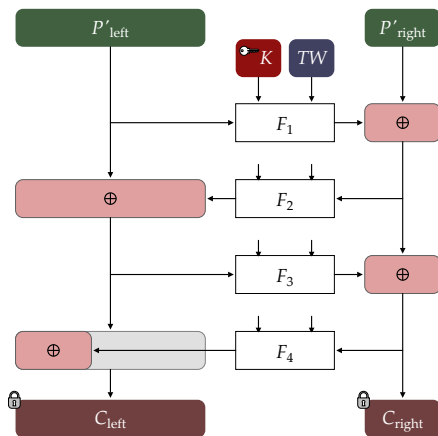


# MR. MONSTER BURRITO: the concept



- Robust AE [Rogaway, ACNS 2014]
  - inspired by AEZ [Hoang, Krovetz, Rogaway, Shrimpton 2014]
  - wide tweakable block cipher
  - variable key-, tweak- and blocksize:  $|K|$ ,  $|TW|$  and  $|B|$
- *Best possible forgery resistance for given message expansion*

# Inside MR. MONSTER BURRITO



■ Based on [Naor Goldreich 2006], thanks [DJB, Tenerife 2013]

- $F_2$  and  $F_3$ : PRF
- $F_1$  and  $F_4$ : constraint is  $\max \text{DP} < 2^{-256}$

# Building blocks of MR. MONSTER BURRITO

- Asymmetric Feistel: right part is single block
- $F_i$ : DONKEYSPONGE instances as in HADDOC PRF
- $F_i$  input:
  - $K$ : key
  - $M$ : injective coding of  $(|B|, TW, S_{\text{left/right}}, i)$
- $F_i$  output length:
  - $F_1, F_3$  and  $F_4$ : single-block
  - $F_2$ : same length as left part
- Permutations
  - KECCAK- $p[1600, n_r]$
  - KECCAK- $p[800, n_r]$

# MR. MONSTER BURRITO features

- Processing
  - block length above rate: close to 100% of SHAKE128
  - short block length: 56 rounds
- Advantages
  - minimum data expansion for given anti-forgery level
  - can even exploit redundancy in plaintext
- Disadvantages: *heavyweight crypto*
  - four-pass
  - inefficient for small block lengths

# Conclusions

KECCAK-based AE:

- our preferred: CAESAR-candidate **KEYAK**
- constrained platforms: CAESAR-candidate **KETJE**
- nonceless: **HADDOC** and **MR. MONSTER BURRITO**
  - will not win beauty contest
  - but performance degradation is not dramatic

Inspired by [Rogaway on AE, ACNS 2014]



Thanks for your attention!