Introduction to symmetric cryptography

Joan Daemen
Institute for Computing and Information Sciences
Radboud University
Šibenik summer school 2016

Outline

Currently we are here...

Security services

Stream encryption

Authentication and authenticated encryption

Building schemes with modes

Building the primitives

Example: Noekeon

Confidentiality

- To protect:
  - people's privacy
  - company assets
  - enforcing business: no pay, no content
  - meta: PIN, password, cryptographic keys
- Data confidentiality
  - only authorised entities get access to the data
  - cryptographic operation: encryption
- Protection against traffic analysis
  - existence of communication between parties
  - frequency and statistics of communication
  - called *metadata*
  - no direct link with a basic cryptographic operation

Data integrity and authentication

- Basic concepts:
  - data integrity: was not modified without proper authorization
  - entity authentication: entity is what it claims to be
  - data origin authentication: data received as it was sent
  - symmetric crypto operation: message authentication codes
- Freshness:
  - entity is there now
  - received message was written recently
  - mechanism: unpredictable challenge
- Protection against replay:
  - authenticated message was not just a copy of an earlier one
  - mechanism: nonce

Secure channel

- cryptographically secured link between two entities
- data confidentiality and data origin authentication
- session-level authentication, protection against
  - insertion of messages
  - removal of messages
  - shuffling of messages
- can be one-directional or full-duplex
- can be online or store-and-forward
- can require freshness or just protection against replay
- examples: SSH, TLS, GP SCP03, ...

Symmetric cryptography operations

- Core business
  - encryption
  - MAC computation
  - authenticated encryption (including sessions)
- Requires secret key shared between sender and receiver
  - key generation requires qualitative random generator
  - key transfer between entities may require other keys
  - a lot can go wrong here!
- On the side
  - cryptographic hashing
  - deterministic random bit generation (DRBG), ...

Currently we are here...

Security services

Stream encryption

Authentication and authenticated encryption

Building schemes with modes

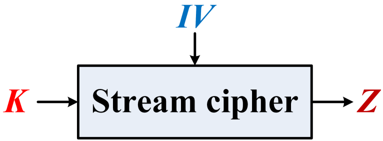
Building the primitives

Example: Noekeon

Encryption: one-time pad

- Let P be a plaintext of n bits: P_1 to P_n
- Assume Z is a shared secret of n bits: Z_1 to Z_n
- Encryption to n-bit cryptogram C
  - $\forall i : C_i = P_i + Z_i$
- Decryption back to P
  - $\forall i : P_i = C_i + Z_i$
- Advantages
  - no expansion
  - very efficient
  - provably secure in information-theoretical sense!
- Disadvantage: requires 1 fresh secret bit per message bit encrypted

Stream cipher

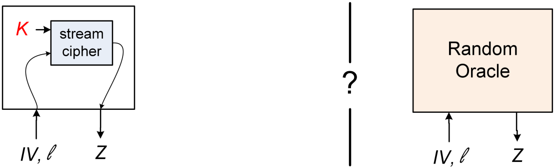


- Generates arbitrary-length keystream Z from
  - K: short secret key, typically 128 or 256 bits
  - IV: initial value, for generating multiple keystreams per K
- Desired properties
  - knowing K: computing $Z = \mathrm{SC}[K](IV)$ shall be efficient
  - not knowing K: predicting Z shall be infeasible for any IV

Random oracle $\mathcal{RO}$ [Bellare-Rogaway 1993]

- A random oracle $\mathcal{RO}$ maps:
  - input of arbitrary length P
  - to an infinite output string Z
- $\mathcal{RO}$ supports queries of following type: (P, ℓ)
  - P: input
  - ℓ: requested number of output bits
- Response Z
  - string of ℓ bits
  - independently and uniformly distributed bits
  - self-consistent: equal inputs P give matching outputs

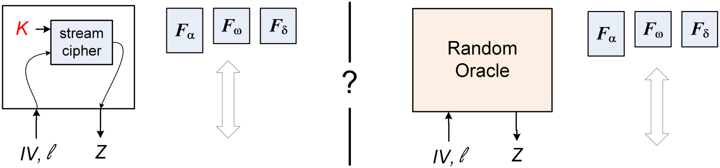
Security notion: Pseudorandom function (PRF)



Distinguishing game (black box version)

- Adversary sends queries Q to system that is either:
  - stream cipher with unknown key K
  - $\mathcal{RO}$
- Then based on responses Z must guess what system is
  - $\Pr(\text{success}) \leq F(|Q|)$: some bound on success probability
  - Advantage: $\text{Adv} = 2\Pr(\text{success}) - 1$

Security notion: PRF (cont'd)



- Black box fails to model public concrete stream cipher
- We give additional query access to internal functions
- We model query complexity in two parts:
  - M: online complexity, represents data
  - N: offline complexity, represents computation and storage
- We express Advantage as $\mathrm{Adv}(M, N)$

Implications of PRF property

- Informally: a function is a PRF if the advantage is negligible
- What really matters is the concrete bound
- A bound $\mathrm{Adv}(M, N)$ for stream cipher implies:
  - any adversary with resources M and N
  - will not learn anything about plaintext from ciphertext
  - with probability $1 - \mathrm{Adv}(M, N)$.
- but for concrete schemes we cannot prove such bounds!

Security claim

- Lack of proof leaves following questions on a concrete scheme:
  - what kind of security does it offer?
  - when does a *demonstrated property* break it?
- Addressed by a security claim
  - statement on expected security of a cryptographic scheme
  - bound on distinguishing advantage from ideal scheme
- For cryptanalysts: challenge
  - break: attack performing better than the claim
- For users: security specification
  - …as long as it is not broken
- Often claims are missing but implied by size parameters

How concrete schemes gain assurance

- The (open) cryptologic activity (70s - today):
  - cryptographic schemes are published
  - ...and (academically) attacked by cryptanalysts
  - ...and corrected/improved,
  - ...and attacked again, etc.
  - by researchers for prestige/career
- This leads to
  - better understanding
  - ever improving cryptographic schemes
- Trust in cryptographic scheme depends on
  - perceived simplicity
  - perceived amount of analytic effort invested in it

Security strength

- Security strength of a cryptographic scheme
  - expected effort required to *break* it
  - expressed in *bits*
  - s bits means best attack has expected complexity 2^s
- Link with bound on distinguishing advantage
  - amount of data and/or computation such that Adv becomes significant
  - kind of coarse
- Current view on computational complexity
  - 80 bits: lightweight
  - 96 bits: solid
  - 128 bits: secure for the foreseeable future
  - 256 bits: for the clueless

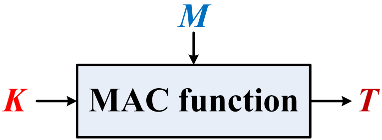
See `www.keylength.com`

Limit to security strength: exhaustive key search

- Single-target: attacker gets couple $(IV, Z = \mathrm{SC}[K](IV))$
  - attacker tries guesses K' until $\mathrm{SC}[K'](IV) = Z$
  - expected effort 2^{k-1}, so strength $k-1$ bits
  - Implicit security claim: no attack better than this
- Multi-target: attacker gets m couples $(IV, Z_i = \mathrm{SC}[K_i](IV))$
  - attacker tries guesses K' until $\exists K_i, \mathrm{SC}[K'](IV) = Z_i$
  - every key guess has success probability $m/2^k$
  - expected effort $2^k/(m+1)$, so strength $\approx k - \log_2(m)$
- key length does not equal security strength!
  - security erosion in case of multi-target
  - can be prevented by making IV global nonce

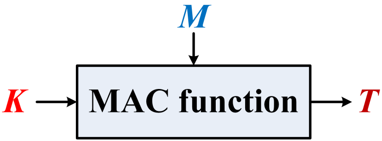
Currently we are here...

Message authentication code (MAC) functions



- Generates short tag T from
  - K: short secret key, typically 128 or 256 bits
  - M: arbitrary-length message
- Desired properties (informally)
  - knowing K: computing $T = \mathrm{MF}[K](M)$ shall be efficient
  - not knowing K: predicting T for any M shall be infeasible

MAC function security



- Forgery: generating pair (M, T) without querying MF[K](M)
- Limit to forgery security strength: random tag guessing
  - single attempt: success probability $\geq 2^{-t}$
  - expected data complexity: 2^t attempts
  - ... if T is unpredictable: PRF!
- MAC function security strength bound by sum of two terms
  - $2^{-t}q$ with $q = \#$ forgery attempts
  - distinguishing advantage of MF

Authenticated encryption (AE) with PRFs only

- Wrapping: $(C, T) = \text{wrap}[K](IV, P)$
  - compute $C = P + \text{PRF}_0[K](IV)$
  - compute $T = \text{PRF}_1[K](C)$
  - return (C, T)
- Unwrapping $P = \text{unwrap}[K](IV, C, T)$ or $\perp$
  - If $T \neq \text{PRF}_1[K](C)$ return $\perp$
  - Else return $P = C + \text{PRF}_0[K](IV)$
- Attacker model:
  - M: wrap and unwrap queries
  - N: computation without access to key
- Security strength:
  - 2 aspects: forgery and secrecy
  - strength for either: min. of t bits (in data) and the PRF strength

Domain separation

- We need one PRF for encryption and one for tag computation
- Reduce to one with domain separation
  - $\mathrm{PRF}[K](P|\mathbf{0})$ and $\mathrm{PRF}[K](P|\mathbf{1})$ are *independent*
  - ... unless PRF is distinguishable from a $\mathcal{RO}$
- So we can take
  - $\mathrm{PRF}_0[K](\cdot) = \mathrm{PRF}[K](\cdot|\mathbf{0})$
  - $\mathrm{PRF}_1[K](\cdot) = \mathrm{PRF}[K](\cdot|\mathbf{1})$
- Generalization: multi-input PRF $\mathrm{PRF}'[K](P_0, P_1, P_2, \ldots)$

(1) Compute $P = \mathrm{encode}(P_0, P_1, P_2, \ldots)$ with injective encoding
(2) Compute $Z = \mathrm{PRF}[K](P)$

iCIS | Digital Security
Radboud University

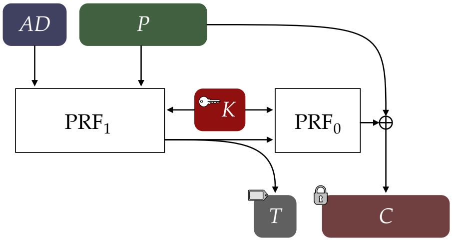
AE with associated data

- Wrapping: $(C, T) = \text{wrap}[K](IV, P, AD)$
  - compute $C = P + \text{PRF}[K](IV, 0)$
  - compute $T = \text{PRF}[K](C, AD, 1)$
  - return (C, T)
- Unwrapping $P = \text{unwrap}[K](IV, C, AD, T)$ or $\perp$
  - If $T \neq \text{PRF}[K](C, AD, 1)$ return $\perp$
  - Else return $P = C + \text{PRF}[K](IV, 0)$
- All you need is one PRF

The problem with the IV

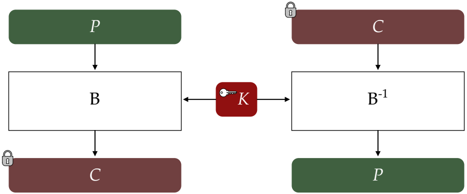
- PRF is deterministic
  - repeating IV leads to same keystream Z
  - for every encryption (or decryption) IV shall be different
  - IV shall be a nonce
- Stream encryption requires nonce management
  - can be done but requires good system architecture
  - not robust against attackers that can manipulate the IV
- Wish for nonce-abuse resilience

AEAD: Synthetic IV mode [Rogaway, Shrimpton 2006]



- Tag on plaintext and AD used as IV for encryption
- $T' \neq T$ lead to independent keystreams Z and Z'
- $(AD, P) \neq (AD', P')$ give independent T and T'
  - colliding tag lead to secrecy violation $P' = P + C + C'$
  - probability if n messages: $2^{-(t+1)} n^2$
  - tag must be twice as long as security strength

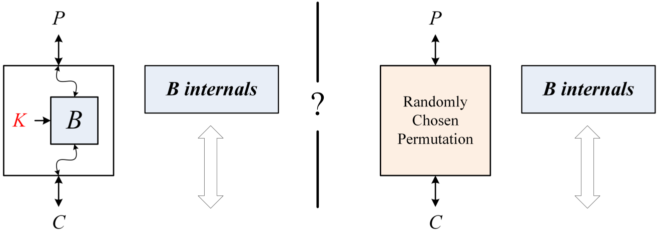
Encryption: wide block encryption



- b-bit message P is subject to permutation
- permutation depends on secret key K: we write $B[K, b]$
- decryption: inverse permutation $B[K, b]^{-1}$
- B: wide block cipher
- Limitation: information leakage if repeated messages
  - short messages
  - low-entropy messages

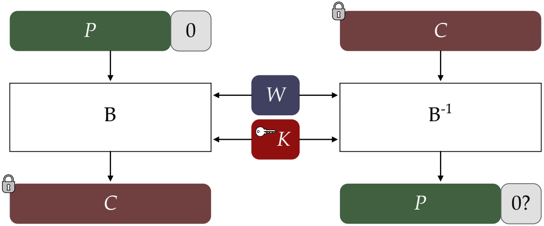
iCIS | Digital Security
Radboud University

Pseudorandom Permutation (PRP) security



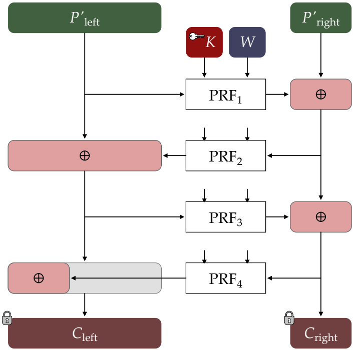
- Advantage in distinguishing $B[K, b]$ from b-bit random permutation
- With b chosen by adversary for each query
- $\mathrm{Adv}(M, N)$
  - N queries Q_c to B internals
  - PRP: M queries Q_s to $B[K, b]$ or RCP
  - SPRP: M includes queries Q_i to $B[K, b]^{-1}$ or RCP^{-1}

AEAD: wide tweakable block cipher [Rogaway, 2014]



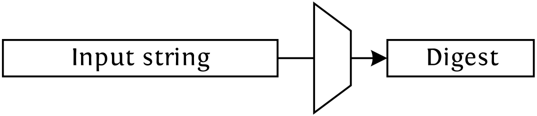
- additional parameter tweak W can take AD or nonce
- no separate tag, reduncancy in plaintext
- forgery strength equal to redundancy in plaintext

Wide tweakable block cipher with a PRF



- e.g., Mr Monster Burrito [Keccak team, 2014]
- Based on [Naor Reingold 1997], thanks [DJB, Tenerife 2013]

On the side: cryptographic hashing



- Hash function: maps arbitrary input strings to n-bit digest
- Variant: eXtendable Output Function (XOF) [FIPS 202]
- Desired property: should behave like an $\mathcal{RO}$
  - distinguishing setup problematic due to absence of secret input
- Implications for security strength
  - collision: $n/2$
  - (first or second) pre-image: n

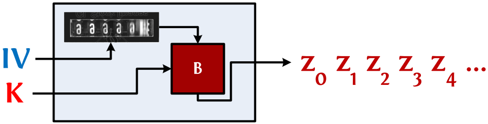
More on hashing by Bart Preneel, this Thursday 2PM

Currently we are here...

Building PRFs

- PRF can have arbitrary input length and/or arbitrary output length
- Two approaches
  - design from scratch: tricky (see e.g. Panama [Daemen Clapp 1998])
  - as modes of use of fixed-length primitives
- Primitives we think we can build from scratch
  - permutation
  - block cipher, including tweakable (maybe)
- Modes can be applied in multiple layers
  - block cipher based on permutation [Even Mansour 1991]
  - tweakable block cipher based on block cipher
- some examples follow

Block cipher based stream cipher: counter mode



Advantage in distinguishing from $\mathcal{RO}$: sum of two terms

- $2^{-(b+1)}M^2$
  - birthday bound: collision in M random values of b bits
  - proven part
- PRP bound of underlying block cipher
  - assumed or claimed part

Block cipher based AEAD: OCB

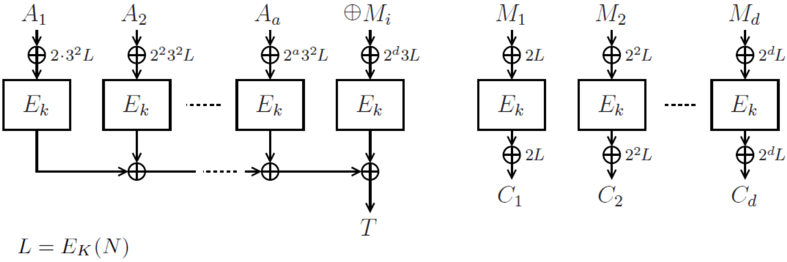
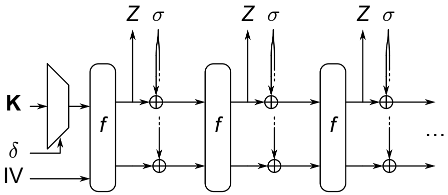


figure: thanks Bart Mennink

- Offset CodeBook [Rogaway et al. 2001]
- Adversary secrecy and forgery advantages: sum of two terms
  - proven term: birthday bound plus $2^{-t}q$
  - PRP bound of underlying block cipher
- Parallelizable, requires nonce, block encryption (but not wide)

Permutation-based PRF: keyed duplex

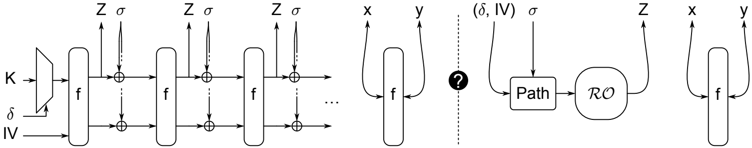


Work in progress

- Based on sponge/duplex with $|Z| = r = b - c$ but
  - full-state absorbing $|\sigma| = b$ [Mennink et al. 2015]
  - caller must provide input σ before getting output Z
  - multi-key built into model
- More than a PRF
  - $\forall i$: mapping of $(IV, \sigma_1, \sigma_2, \ldots \sigma_i)$ to Z_i is a PRF
- Can be used as stream cipher, MAC function, AE scheme, PRNG

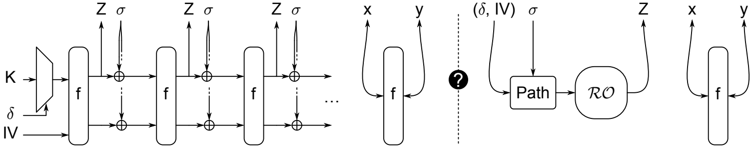
iCIS | Digital Security
Radboud University

Generic security of keyed duplex: the setup



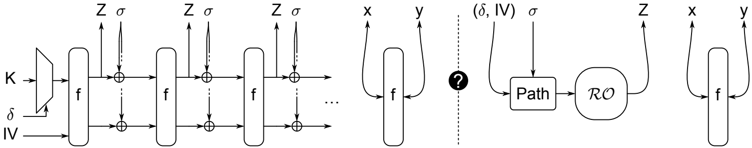
- Advantage of distinguishing from ideal function
  - $\mathcal{RO}$-based object with the same interface
  - additional query access to underlying permutation f
- but f cannot be a PRP

Security of keyed duplex: requirements for f



- f cannot be a PRP
  - as there is no dedicated key input
  - similar to defining distinguishing setup to hash functions
- Some requirements for f
  - given any set of N couples $\{(x_i, y_i)\}$, getting fresh (x, y) with $\Pr(\text{succ.}) > 1/(2^b - N)$ has about same cost as $f^{(-1)}(x)$
  - let κ be a string with last c bits unknown. Given M chosen values s_i, $t_i = \kappa + f(s_i + \kappa)$, finding κ in N queries has $\Pr(\text{succ.}) < NM2^{-c}$
  - very similar to those for block cipher with PRP ambition

Generic security of keyed duplex: the bound



$$\frac{\mu N}{2^k} + \frac{(L + 2\nu)N}{2^c} + \frac{L^2}{2^{c+1}} + \frac{M^2}{2^b}$$

- with
  - N: # queries to f or f^{-1}
  - M: # queries to keyed duplex or $\mathcal{RO}$-equivalent
  - L: # queries to keyed duplex or $\mathcal{RO}$ with repeated path
  - $\mu = \max_{IV}$ # init queries with different keys
  - ν: chosen such that probability of ν-wise multi-collision in set of M r-bit values is negligible

Counter-like stream cipher with keyed duplex

- Only init calls with Z keystream block
- IV is nonce, so $L = 0$. We get:

$$\frac{\mu N}{2^k} + \frac{2\nu N}{2^c} + \frac{M^2}{2^b}$$

- If global nonce or single key $\mu = 1$
- ν: if $r > c$ this reduces to 2
- For s bits of security we can take $k = s + \epsilon_1$ and $c = s + \epsilon_2$

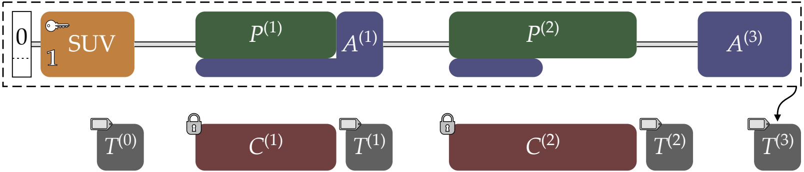
MAC with keyed duplex

- Message padded and fed via IV and σ blocks
- t-bit tag, capacity is de facto $b - t$
- adversary chooses IV so
  - L can be as large as $M/2$
  - μ are total number of keys m

$$\frac{mN}{2^k} + \frac{MN}{2^{b-t+2}} + \frac{M^2}{2^{b-t+3}}$$

- Suggests a minimum width of the permutation: $b > s + t + \log_2(M)$
- E.g. $s = 128, t = 64, M \leq 2^{64}$ suggests $b \geq 256$

AE secure channel with keyed duplex: Motorist



[Keyak team 2015]

- Session: tag authenticates all message history
- Plaintext absorbed in outer part, AD in inner part also
- SUV = Secret and Unique Value $\rightarrow L = 0$
- Used in Keyak with $c = 256$ and $b = 1600$ or $b = 800$:

$$\frac{\mu N}{2^k} + \frac{N}{2^{255}}$$

Currently we are here...

How to build a cryptographic permutation?

- Two principles:
  - for f and g permutations, $g \circ f$ is a permutation
  - in general $g \circ f$ is more *complex* than f and g
- Iterated permutation: apply a simple round function repeatedly
- Let A_i be the addition of a constant, then:

$$f = A_r \circ R \circ A_{r-1} \circ R \ldots A_1 \circ R \circ A_0$$

- Choose round function R and # rounds such that:
  - $f(a) + f(a + \Delta_a)$ hard to predict from Δ_a
  - low input-output correlation $C(u^{\mathrm{T}} f(a), v^{\mathrm{T}} a)$
  - f has high algebraic degree
  - f has no symmetry properties, ...

How to build a block cipher

- Key-alternating: apply a simple round function repeatedly
- Let K_i be the addition of a round key, then:

$$B[K] = K_r \circ R \circ K_{r-1} \circ R \ldots K_1 \circ R \circ K_0$$

- Round keys K_i derived from K
  - mapping from K to array of K_i: *key expansion*
- Additional constraint: R shall have an efficient inverse
- Simpler method: Even-Mansour

$$B[K] = K \circ f \circ K$$

- Better: $K_i = K + A_i$ with A_i round constant

Building a round function: wide trail strategy

- Three layers, sharing the following desired properties
  - cheap to implement and secure against side channel attacks
  - simple, and with high amount of symmetry
- Strongly based on differential (DC) and linear cryptanalysis (LC)
- Non-linear layer
  - DC: max probability decrease with HW of input difference
  - LC: correlation decreases with HW of output parity
- Mixing layer (linear)
  - DC: difference propagation with low HW input AND output are rare
  - LC: correlations between low HW input AND output are rare
- Transposition (AKA dispersion) layer
  - moves *nearby* bits away from each other
  - nearness determined by other layers

Currently we are here...

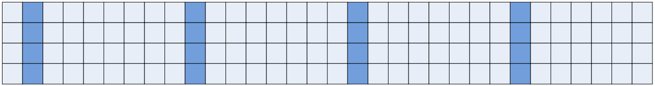
Noekeon [Daemen, Peeters, Rijmen and Van Assche, 2000]

- Block cipher
  - 128-bit blocks
  - 128-bit keys
  - security claim: PRP $2^{-128}\mu N$
- Pedigree
  - bit-slice cipher, similar to Serpent [Biham, Knudsen, Anderson, 1997]
  - descendent of 3-Way [Daemen 1993] and BaseKing [Daemen 1993]
- Design goals:
  - simplicity: interesting object for (crypt)analysis
  - lightweight: hardened low-cost implementations in HW and SW
  - LC/DC: proof no 12-round trails exist with ELP/EDP $> 2^{-144}$

See http://gro.noekeon.org/

iCIS | Digital Security
Radboud University

The Noekeon state



- Two-dimensional $4 \times \ell$ array
  - 4 *rows*
  - ℓ *columns*
- Additional partitioning of the state: *slices*
  - $\ell/4$ slices
- $\ell = 32$

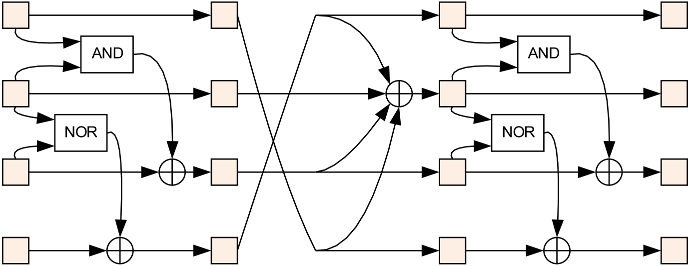
Round transformation

- γ: nonlinear layer
  - 4-bit S-box operating on columns
  - Involution
- θ: combines mixing layer and round key addition
  - Linear 16-bit mixing layer operating on slices
  - Involution
- π: dispersion between slices
  - Rotation of bits within ℓ-bit rows
  - Two instances that are each others inverse
- ι: round constant addition for asymmetry

The round and its inverse

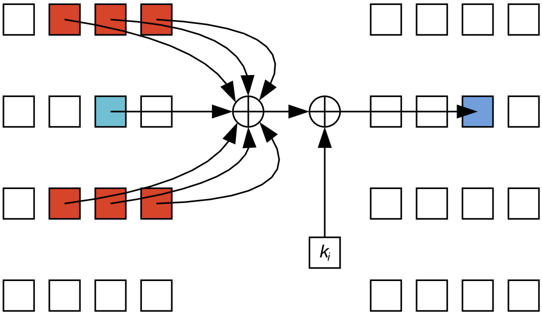
- Round: $\pi_2 \circ \gamma \circ \pi_1 \circ \theta[k]$
- Inverse round:
  - $\theta[k]^{-1} \circ \pi_1^{-1} \circ \gamma^{-1} \circ \pi_2^{-1}$
  - $\theta[k] \circ \pi_2 \circ \gamma \circ \pi_1$
- $\theta[k]$ as final transformation:
  - Regrouping: round of inverse cipher = cipher round
  - round constants prevent involution
- Noekeon: 16 rounds and a final transformation
  - Inverse cipher equal to cipher itself
  - Asymmetry provided by round constants only

Nonlinear layer γ



- Two identical nonlinear steps with a linear step in between
- Simple algebraic expression

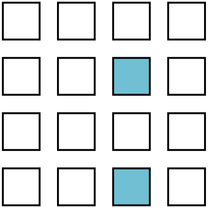
Mixing layer θ



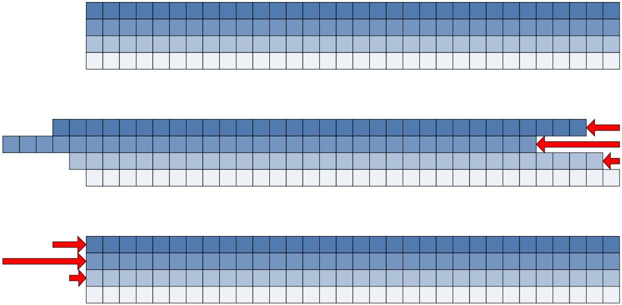
- High average diffusion
- Small number of operations thanks to symmetry

Mixing layer θ cont'd

- Branch number $\mathcal{B}$ only 4 due to symmetry
- Invariant sparse states in kernel, e.g.:



Transposition steps π



- π_1 and π_2 are each others inverses

Lightweight aspect

- Hardware
  - # gates: [640 – 1050] XOR, 64 AND, 64 NOR, 128 MUX
  - Gate delay: 7 XOR, 1 AND, 1 MUX
  - Coprocessor architecture: speed/area trade-off
- Software: e.g. numbers for ARM7:
  - code size 332 bytes, 44.5 cycles/byte
  - code size 3688 bytes, 30 cycles/byte
  - RAM usage: everything in registers
- Cipher and inverse are equal: re-use of circuit and code

Currently we are here...

Security services

Stream encryption

Authentication and authenticated encryption

Building schemes with modes

Building the primitives

Example: Noekeon

Conclusions

- Modern symmetric cryptographic schemes
  - are built in a modular way
  - from (keyed) permutations as primitives
  - and modes making use of them
- Modes have certain provable security properties
- Primitives cannot be proven secure but there is hope
  - insight grows thanks to cryptologic activity
  - better and better designs

Thanks for your attention!