Introduction to the sponge and duplex constructions

Gilles Van Assche[1]

[1]STMicroelectronics

Indocrypt, New Delhi, December 2018

Based on joint work with Elena Andreeva, Guido Bertoni, Joan Daemen, Bart Mennink, Michaël Peeters, Ronny Van Keer

Outline

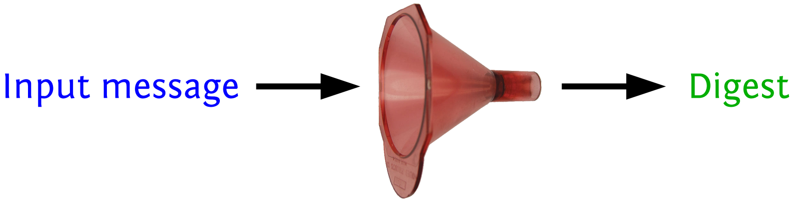
1. Unkeyed applications
   - Hashing requirements
   - Traditional constructions
   - Modern generic security
   - The sponge construction
   - The duplex construction
2. Intermezzo: why permutations?
3. Keyed applications
   - The outer keyed sponge and duplex constructions
   - Generic security, the beginning
   - Generic security, progressing
   - The full-state keyed duplex construction

Outline

Outline

Cryptographic hash functions

$$h : \{0,1\}^* \rightarrow \{0,1\}^n$$



- Applications
  - *Signatures*: $\mathrm{sign}_{\mathrm{RSA}}(h(M))$ instead of $\mathrm{sign}_{\mathrm{RSA}}(M)$
  - *Key derivation*: master key K to derived keys $(K_i = h(K\|i))$
  - *Bit commitment, predictions*: $h(\text{what I know})$
  - *Message authentication*: $h(K\|M)$
  - ...

Generalized: extendable output function (XOF)

$$h : \{0,1\}^* \to \{0,1\}^*$$

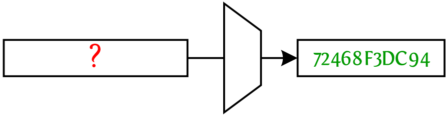
"XOF: a function in which the output can be extended to any length."

[Ray Perlner, SHA-3 workshop 2014]

- Applications
  - *Signatures*: full-domain hashing, mask generating function
  - *Key derivation*: as many/long derived keys as needed
  - *Stream cipher*: $C = P \oplus h(K \| \text{nonce})$

Preimage resistance

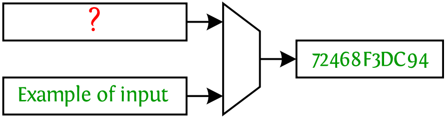
- Given $y \in \mathbf{Z}_2^n$, find $x \in \mathbf{Z}_2^*$ such that $h(x) = y$



- If h is a random function, about 2^n attempts are needed
- **Example**: given derived key $K_1 = h(K \| 1)$, find master key K

Second preimage resistance

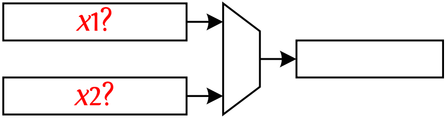
- Given $x \in \mathbf{Z}_2^*$, find $x' \neq x$ such that $h(x') = h(x)$



- If h is a random function, about 2^n attempts are needed
- **Example**: signature forging
  - Given M and $\text{sign}(h(M))$, find $M' \neq M$ with equal signature

Collision resistance

- Find $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$



- If h is a random function, about $2^{n/2}$ attempts are needed
  - Birthday paradox: among 23 people, probably two have same birthday
  - Scales as $\sqrt{|\text{range}|} = 2^{n/2}$

Other requirements

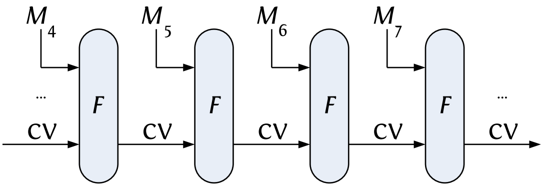
- Security claims by listing desired properties
  - Collision resistant
  - (Second-) preimage resistant
    - Multi-target preimage resistance
    - Chosen-target forced-prefix preimage resistance
  - Correlation-free
  - Resistant against length-extension attacks
  - ...
- But ever-growing list of desired properties
- A good hash function should behave like a **random mapping**...

Security requirements summarized

- Hash or XOF h with n-bit output
- Modern security requirements
  - h behaves like a random mapping
  - ... up to security strength s
- Classical security requirements, derived from it

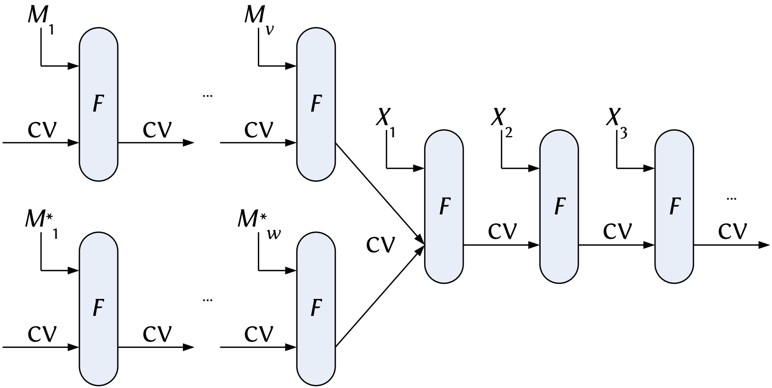
Preimage resistance	$2^{\min(n,s)}$
Second-preimage resistance	$2^{\min(n,s)}$
Collision resistance	$2^{\min(n/2,s)}$

Iterated functions



- All practical hash functions are iterated
  - Message M cut into blocks $M_1, \ldots, M_l$
  - q-bit chaining value
- Output is function of final chaining value

Internal collisions!

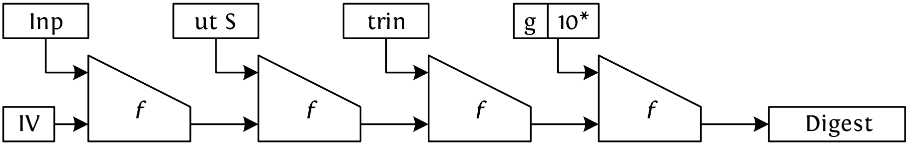


- Different inputs M and M^* giving the same chaining value
- Messages $M \| X$ and $M^* \| X$ always collide for any string X

Does not occur in a random mapping!

Outline

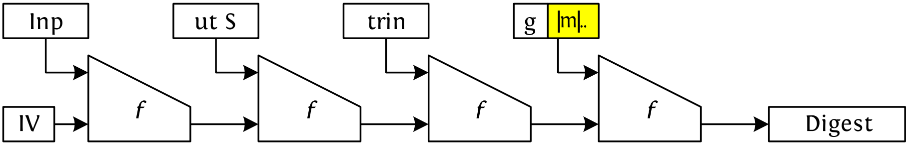
Merkle-Damgård



- Uses a compression function from $n + m$ bits to n bits
- Instances: MD5, SHA-1, SHA-2 ...
- Merkle-Damgård strengthening

[Merkle, CRYPTO'89], [Damgård, CRYPTO'89]

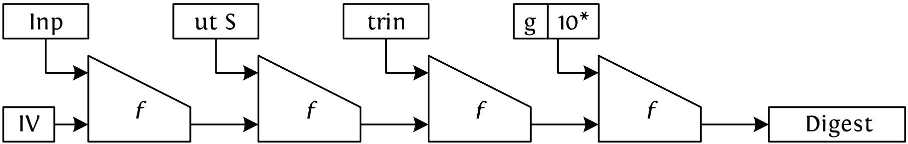
Merkle-Damgård

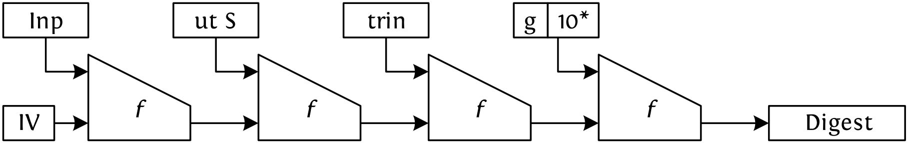


- Uses a compression function from $n + m$ bits to n bits
- Instances: MD5, SHA-1, SHA-2 ...
- Merkle-Damgård strengthening

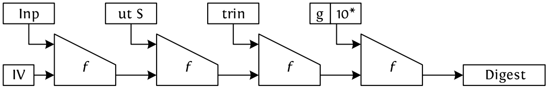
[Merkle, CRYPTO'89], [Damgård, CRYPTO'89]

Merkle-Damgård: preserving collision resistance





Merkle-Damgård: length extension



Recurrence (modulo the padding):

- $h(M_1) = f(\mathrm{IV}, M_1) = \mathrm{CV}_1$
- $h(M_1 \| \dots \| M_i) = f(\mathrm{CV}_{i-1}, M_i) = \mathrm{CV}_i$

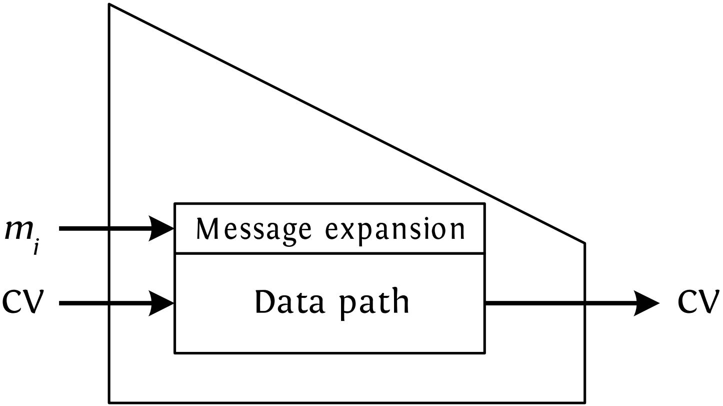
Forgery on naïve message authentication code (MAC):

- $\mathrm{MAC}(M) = h(\mathrm{Key} \| M) = \mathrm{CV}$
- $\mathrm{MAC}(M \| \mathrm{suffix}) = f(\mathrm{CV} \| \mathrm{suffix})$

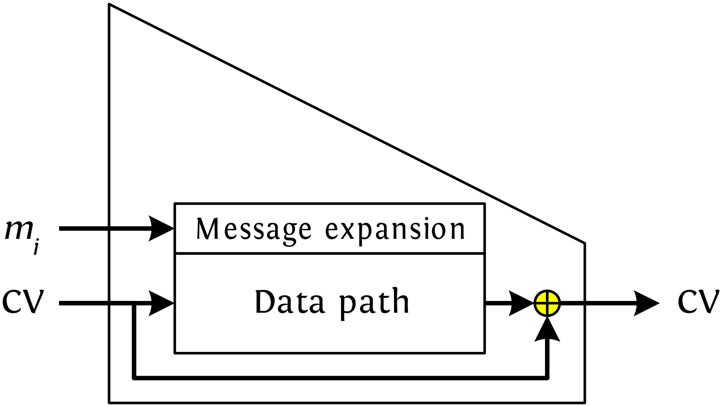
Solution: **HMAC**

$$\mathrm{HMAC}(M) = h(\mathrm{Key}_{\mathrm{out}} \| h(\mathrm{Key}_{\mathrm{in}} \| M))$$

Davies-Meyer

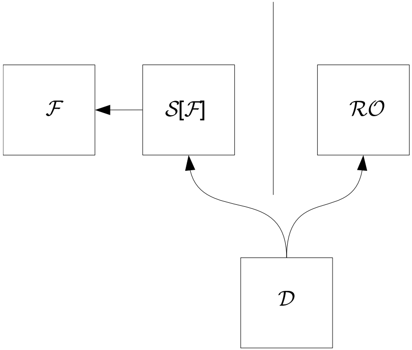


Davies-Meyer



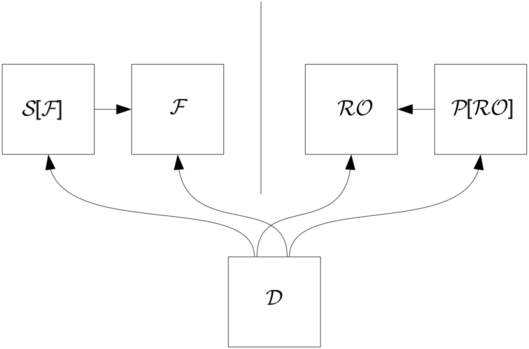
Outline

Generic security: indistinguishability



- Adversary $\mathcal{D}$ must tell apart
  - the ideal function: a monolithic random oracle $\mathcal{RO}$
  - construction $\mathcal{S}[\mathcal{F}]$ calling an ideal primitive $\mathcal{F}$
- Express $\Pr(\text{success}|\mathcal{D})$ as a function of total cost of queries N
- Problem: in real world, $\mathcal{F}$ is available to adversary

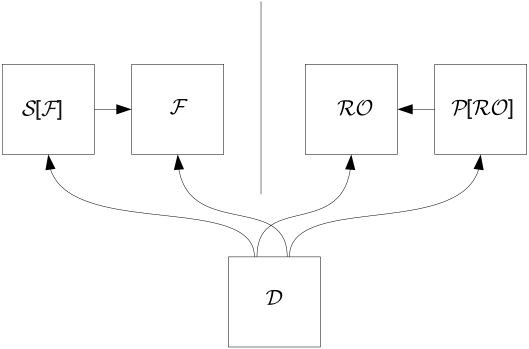
Generic security: indifferentiability [Maurer et al. (2004)]



Applied to hash functions in [Coron et al. (2005)]

- distinguishing mode-of-use from ideal function ($\mathcal{RO}$)
- covers adversary with access to primitive $\mathcal{F}$ at left
- additional interface, covered by a *simulator* at right

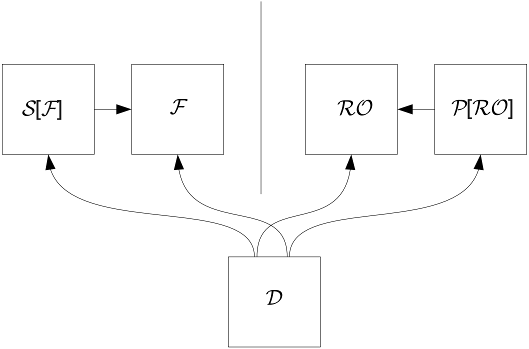
Generic security: indifferentiability [Maurer et al. (2004)]



Methodology:

- build $\mathcal{P}$ that makes left/right distinguishing difficult
- prove bound for advantage given this simulator $\mathcal{P}$
- $\mathcal{P}$ may query $\mathcal{RO}$ for acting $\mathcal{S}$-consistently: $\mathcal{P}[\mathcal{RO}]$

Generic security: indifferentiability [Maurer et al. (2004)]



$$\mathsf{Adv}(q) = \left| \Pr\left(\mathcal{D}^{\mathcal{S}[\mathcal{F}],\mathcal{F}}\right) - \Pr\left(\mathcal{D}^{\mathcal{RO},\mathcal{P}[\mathcal{RO}]}\right) \right| \leq \epsilon(q)$$

Consequences of indifferentiability

- Let $\mathcal{D}$: n-bit output pre-image attack. Success probability:
  - for random oracle: $P_{\text{pre}}(\mathcal{D}|\mathcal{RO}) = q2^{-n}$
  - for our construction: $P_{\text{pre}}(\mathcal{D}|\mathcal{S}[\mathcal{F}]) = ?$
- A distinguisher $\mathcal{D}$ with $\text{Adv}(q) = P_{\text{pre}}(\mathcal{D}|\mathcal{S}[\mathcal{F}]) - P_{\text{pre}}(\mathcal{D}|\mathcal{RO})$
  - do pre-image attack
  - if success, conclude random sponge and $\mathcal{RO}$ otherwise
- But we have a proven bound $\text{Adv}(q) \leq \epsilon(q)$, so

$$P_{\text{pre}}(\mathcal{D}|\mathcal{S}[\mathcal{F}]) \leq P_{\text{pre}}(\mathcal{D}|\mathcal{RO}) + \epsilon(q)$$

- Can be generalized to any attack

Consequences of indifferentiability

Theorem 2. *Let $\mathcal{H}$ be a hash function, built on underlying primitive π, and RO be a random oracle, where $\mathcal{H}$ and RO have the same domain and range space. Denote by $\boldsymbol{Adv}_{\mathcal{H}}^{\mathrm{pro}}(q)$ the advantage of distinguishing $(\mathcal{H}, \pi)$ from (RO, S), for some simulator S, maximized over all distinguishers $\mathcal{D}$ making at most q queries. Let* atk *be a security property of $\mathcal{H}$. Denote by $\boldsymbol{Adv}_{\mathcal{H}}^{\mathrm{atk}}(q)$ the advantage of breaking $\mathcal{H}$ under* atk*, maximized over all adversaries $\mathcal{A}$ making at most q queries. Then:*

$$\boldsymbol{Adv}_{\mathcal{H}}^{\mathrm{atk}}(q) \leq \boldsymbol{Pr}_{RO}^{\mathrm{atk}}(q) + \boldsymbol{Adv}_{\mathcal{H}}^{\mathrm{pro}}(q), \tag{1}$$

where $\boldsymbol{Pr}_{RO}^{\mathrm{atk}}(q)$ denotes the success probability of a generic attack against $\mathcal{H}$ under atk*, after at most q queries.*

[Andreeva, Mennink, Preneel, ISC 2010]

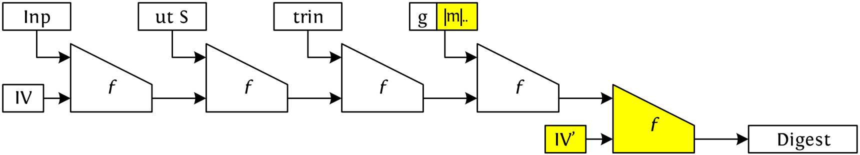
Limitations of indifferentiability

- Only about the mode
  - No security proof with a concrete primitive
- Only about single-stage games [Ristenpart et al., Eurocrypt 2011]
  - Example: hash-based storage auditing

$$Z = h(\mathsf{File} \| C)$$

Making Merkle-Damgård indifferentiable

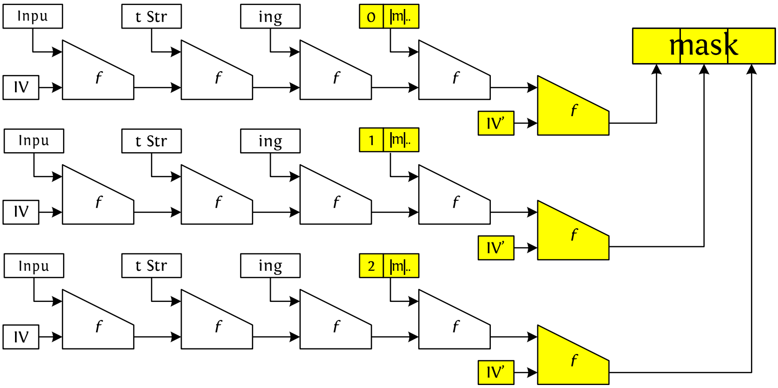
Enveloped Merkle-Damgård



[Bellare and Ristenpart, Asiacrypt 2006]

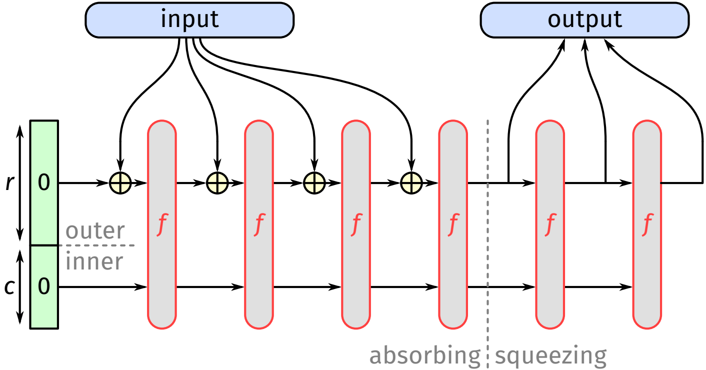
Making Merkle-Damgård suitable for XOFs

Mask generating function construction "MGF1"



Outline

The sponge construction



- Calls a b-bit permutation f, with $b = r + c$
  - r bits of *rate*
  - c bits of *capacity* (security parameter)
- Natively implements a XOF

Generic security of the sponge construction

Theorem (Bound on the $\mathcal{RO}$-differentiating advantage of sponge)

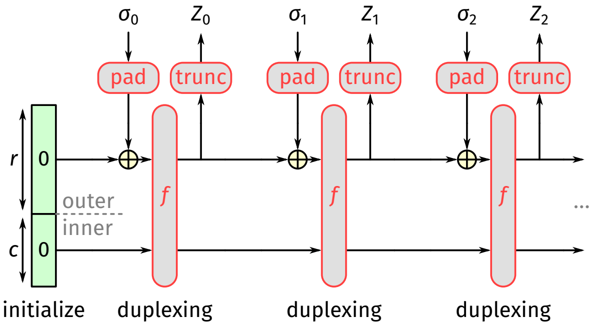
$$A \leq \frac{N^2}{2^{c+1}}$$

A: differentiating advantage of random sponge from random oracle
N: total data complexity *c: capacity* [KECCAK Team, Eurocrypt 2008]

Preimage resistance	$2^{\min(n,c/2)}$
Second-preimage resistance	$2^{\min(n,c/2)}$
Collision resistance	$2^{\min(n/2,c/2)}$
Any other attack	$2^{\min(\mathcal{RO},c/2)}$

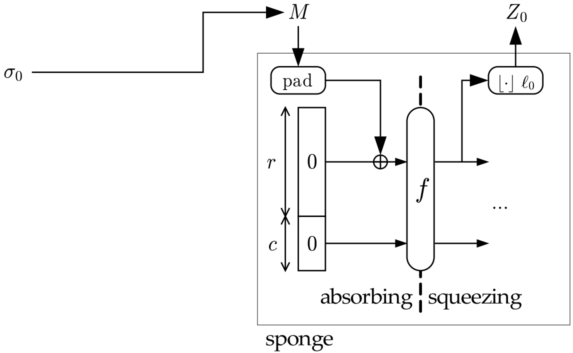
Outline

The duplex construction



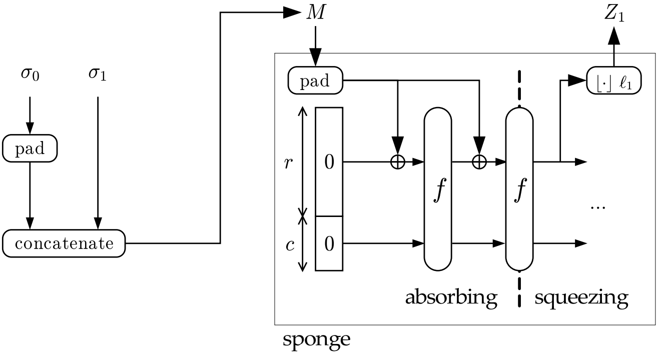
- Object: $D = \textsc{duplex}[f, \text{pad}, r]$
- Requesting ℓ-bit output $Z = D.\text{duplexing}(\sigma, \ell)$
  - input σ and output Z limited in length
  - Z depends on all previous inputs

Generating duplex responses with a sponge



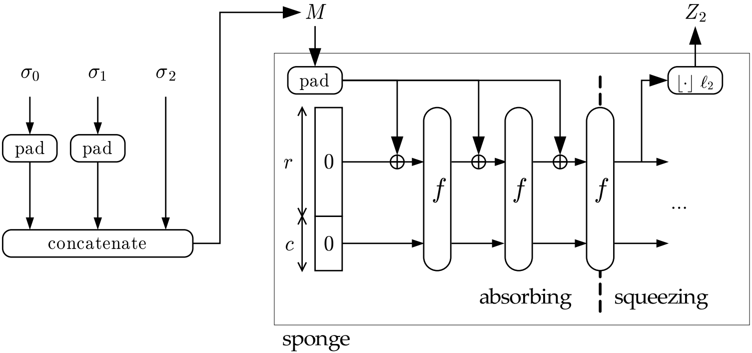
$$Z_0 = \mathsf{sponge}(\sigma_0, \ell_0)$$

Generating duplex responses with a sponge



$$Z_1 = \mathsf{sponge}(\mathsf{pad}(\sigma_0)||\sigma_1, \ell_1)$$

Generating duplex responses with a sponge



$$Z_2 = \mathsf{sponge}(\mathsf{pad}(\sigma_0)||\mathsf{pad}(\sigma_1)||\sigma_2, \ell_2)$$

Security of the duplex construction

Duplexing-sponge lemma

Every output block of a duplex object $\textsc{duplex}[f, \text{pad}, r]$ is a valid output of $\textsc{sponge}[f, \text{pad}, r]$

Proof is trivial

Corollary

The security of $\textsc{duplex}[f, \text{pad}, r]$ can be reduced to that of $\textsc{sponge}[f, \text{pad}, r]$

Outline

Symmetric crypto: what textbooks and intro's say

Symmetric cryptography primitives:

- Block ciphers
- Key stream generators
- Hash functions

And their modes-of-use

Picture by GlasgowAmateur

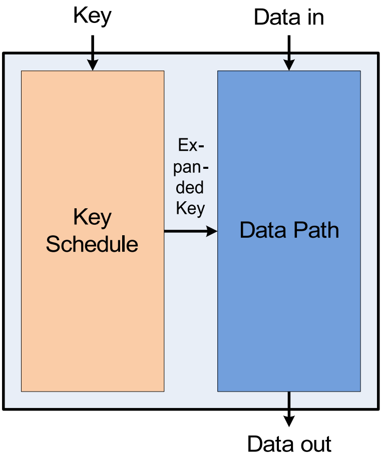
The truth about symmetric crypto today

Block ciphers:

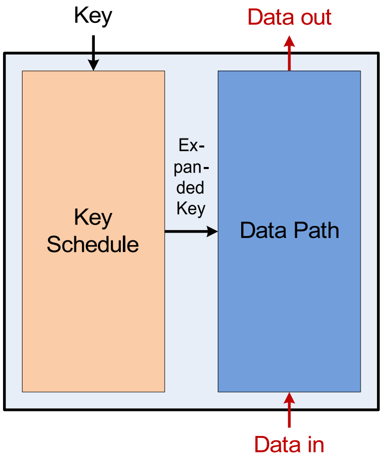
What block cipher are used for

- Hashing (Davies-Meyer) and its modes HMAC, MGF1, ...
- Block encryption: ECB, CBC, ...
- Stream encryption:
  - synchronous: counter mode, OFB, ...
  - self-synchronizing: CFB
- MAC computation: CBC-MAC, C-MAC, ...
- Authenticated encryption: OCB, GCM, CCM ...

Block cipher operation



Block cipher operation: the inverse



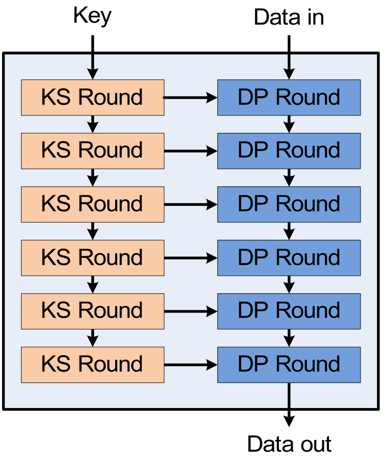
When do you need the inverse?

Indicated in red:

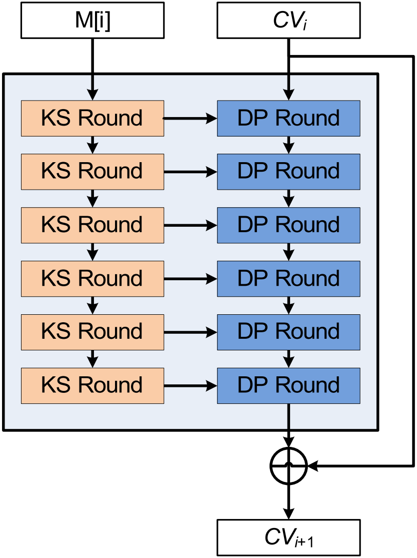
- Hashing and its modes HMAC, MGF1, ...
- Block encryption: ECB, CBC, ...
- Stream encryption:
  - synchronous: counter mode, OFB, ...
  - self-synchronizing: CFB
- MAC computation: CBC-MAC, C-MAC, ...
- Authenticated encryption: OCB, GCM, CCM ...
  - Most schemes with misuse-resistant claims

So for most uses you don't need the inverse!

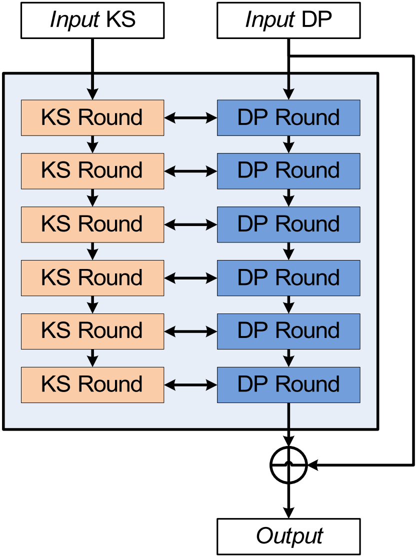
Block cipher internals



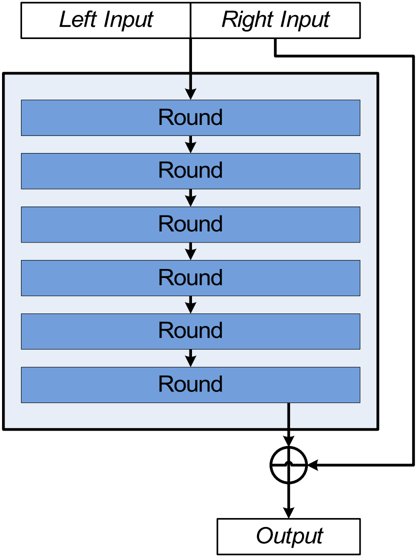
Davies-Meyer compression function



Removing restrictions not required in hashing



Simplifying the view: iterated permutation



Designing a permutation

- Remaining problem: design of iterated permutation
  - round function: good approaches known
  - asymmetry: round constants
- Advantages with respect to block ciphers:
  - less barriers $\Rightarrow$ more diffusion
  - no more need for efficient inverse
  - no more worries about key schedule

Examples of permutations

- In Salsa, Chacha, Grindahl...
- In SHA-3 candidates: CubeHash, Grøstl, JH, MD6, ...
- In CAESAR candidates: Ascon, Icepole, Norx, π-cipher, Primates, Stribob, ...
- In recent proposals: Gimli, Xoodoo

And of course in KECCAK

What textbooks and intro's should say

Symmetric cryptography primitives:

- Block ciphers
- Key stream generators
- **Permutations**

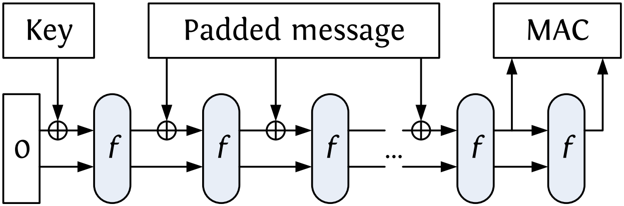
And their modes-of-use

Picture by Sébastien Wiertz

Outline

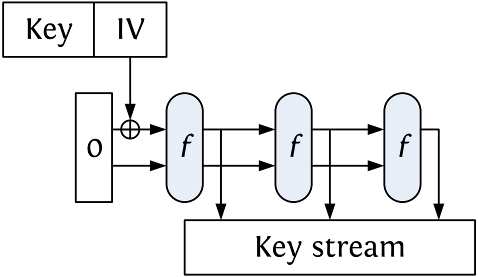
Outline

Message authentication codes



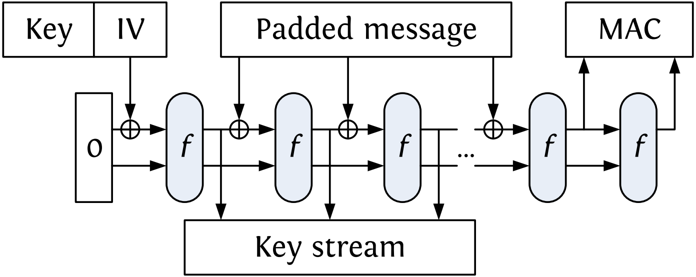
- Using sponge
- See also **KMAC** [NIST SP 800-185]

Stream encryption



- Using sponge
- Long output stream per IV: similar to OFB mode
- Short output stream per IV: similar to counter mode

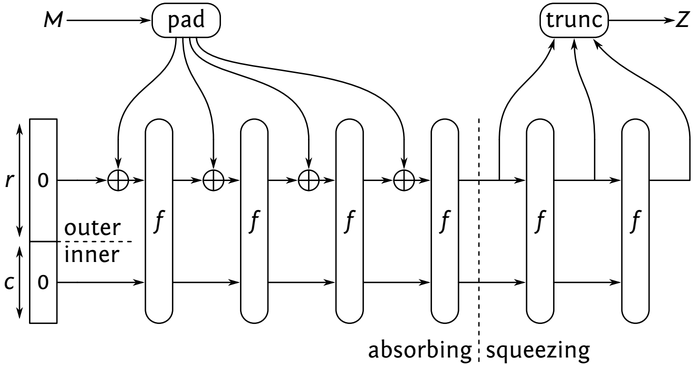
Authenticated encryption: spongeWrap



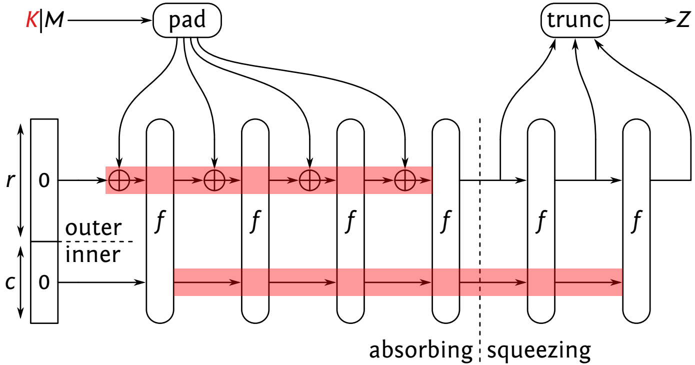
- Using duplex
- Adopted by several CAESAR candidates

[KECCAK Team, SAC 2011]

Outer keyed sponge

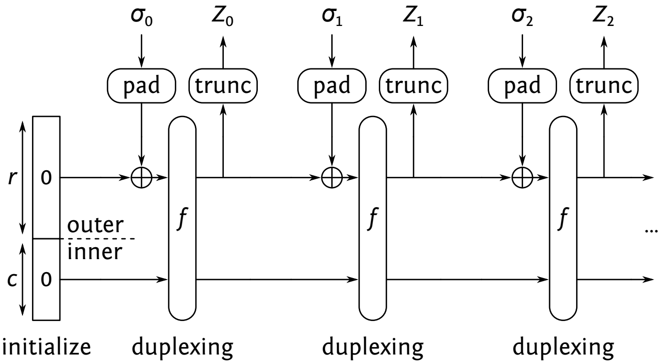


Outer keyed sponge



$$\mathrm{OKS}_K^f(M) = \textsc{sponge}^f(K||M)$$

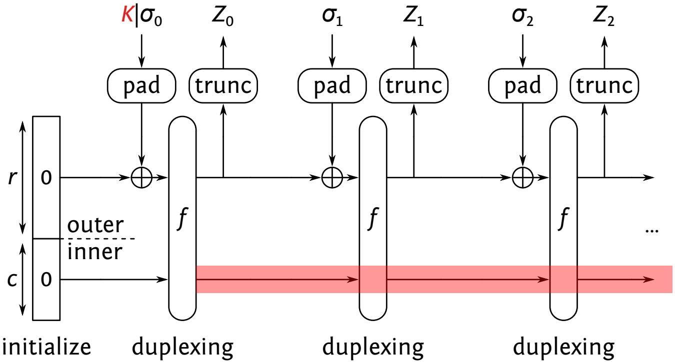
Outer keyed duplex



Duplexing-sponge lemma

$Z_i = \text{SPONGE}(\sigma_0||\text{pad}||\ldots||\sigma_i)$

Outer keyed duplex

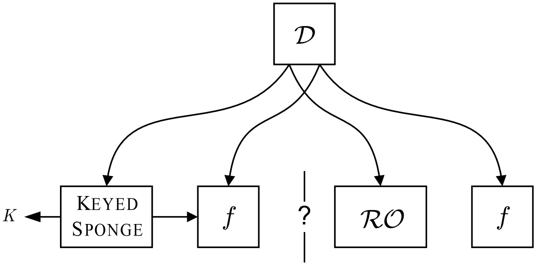


Duplexing-sponge lemma

$Z_i = \text{SPONGE}(K||\sigma_0||\text{pad}||\ldots||\sigma_i) \Rightarrow$ equivalent to OKS_K

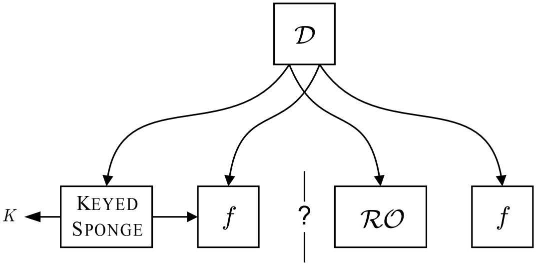
Outline

Keyed sponge: distinguishing setting



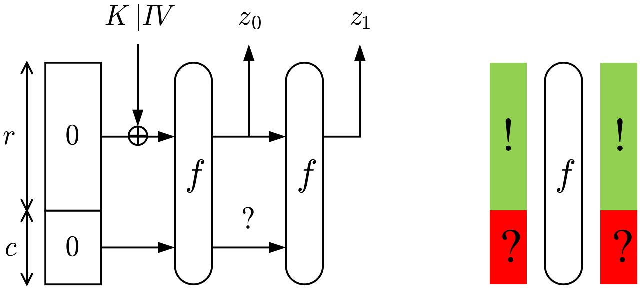
- Straightforward bound: $M^2/2^{c+1} + M/2^k$
- Security strength s: expected complexity of succesful attack
  - strength s means attack complexity 2^s
  - bounds can be converted to security strength statements
- Here: $s \leq \min(c/2, k)$
  - e.g., $s = 128$ requires $c = 256$ and $k = 128$
  - $c/2$: birthday bound

More fine-grained attack complexity



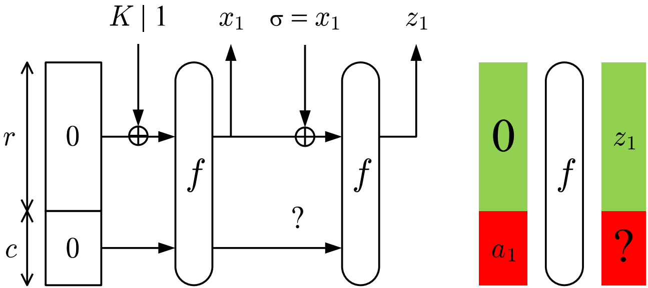
- Splitting attack complexity:
  - queries to construction: data complexity M
  - queries to f or f^{-1}: computational complexity N
- Our ambition around 2010: $M^2/2^{c+1} + NM/2^c + N/2^k$
- If we limit data complexity $M \leq 2^a \lll 2^{c/2}$:
  - $s \leq \min(c - a, k)$
  - e.g., $s = 128$ and $a = 64$ require $c = 192$ and $k = 128$

Intuition behind $NM/2^c$



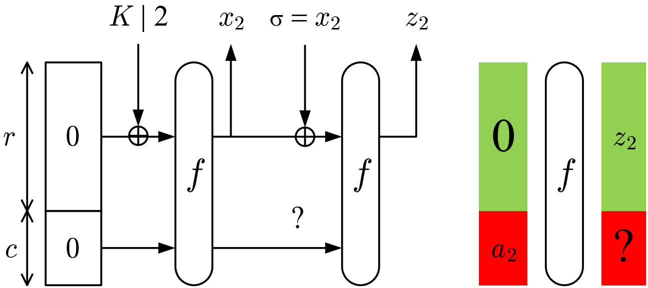
- Typically just one instance with the same partial r-bit input
- Success probability per guess: $1/2^c$

Intuition behind $NM/2^c$



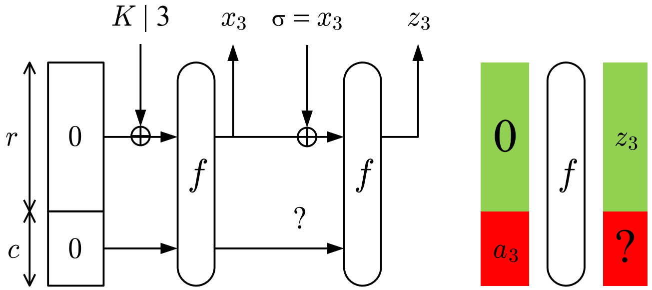
- Multiple instances ($\mu \leq M$) with same partial r-bit input
- Success probability per guess: $\mu/2^c$

Intuition behind $NM/2^c$



- Multiple instances ($\mu \leq M$) with same partial r-bit input
- Success probability per guess: $\mu/2^c$

Intuition behind $NM/2^c$



- Multiple instances ($\mu \leq M$) with same partial r-bit input
- Success probability per guess: $\mu/2^c$

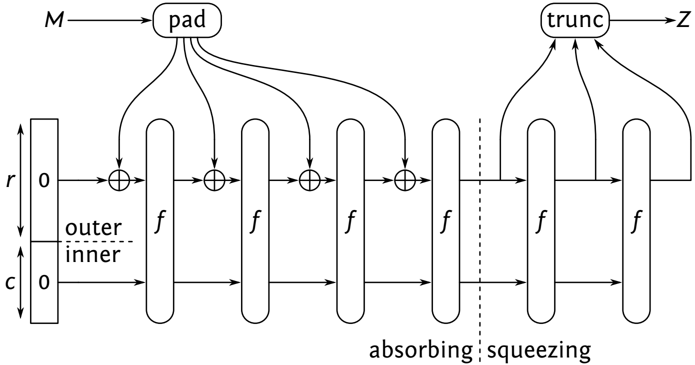
An initial attempt

- Proof of bound $M^2/2^{c+1} + NM/2^{c-1} + N/2^k$
- Problems and limitations
  - does not cover multi-target attacks
  - did not convince reviewers
  - does not support new variants, e.g., inner-keyed sponge

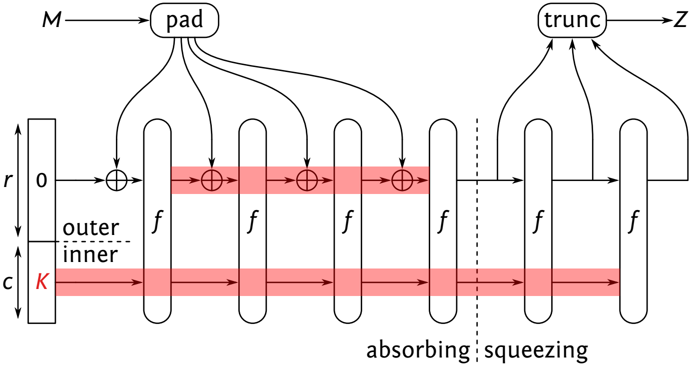
[Keccak Team, SKEW 2011]

Outline

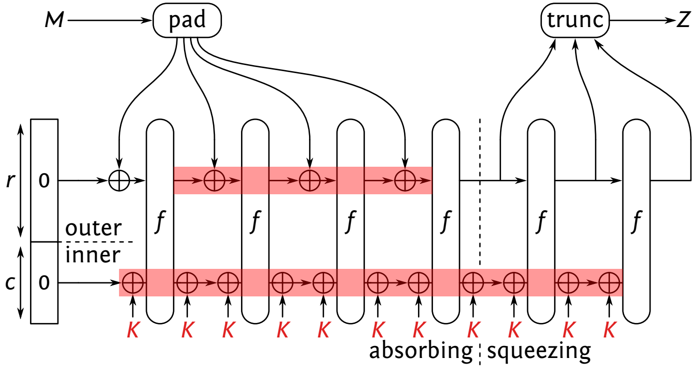
The inner keyed sponge



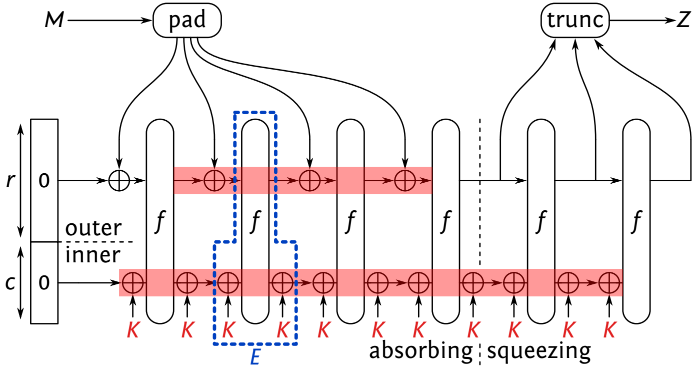
The inner keyed sponge



The inner keyed sponge



The inner keyed sponge



$\mathrm{IKS}_K^f(M) = \textsc{sponge}^{E_K^f}(M)$ [Chang, Dworkin, Hong, Kelsey, Nandi, 2012]

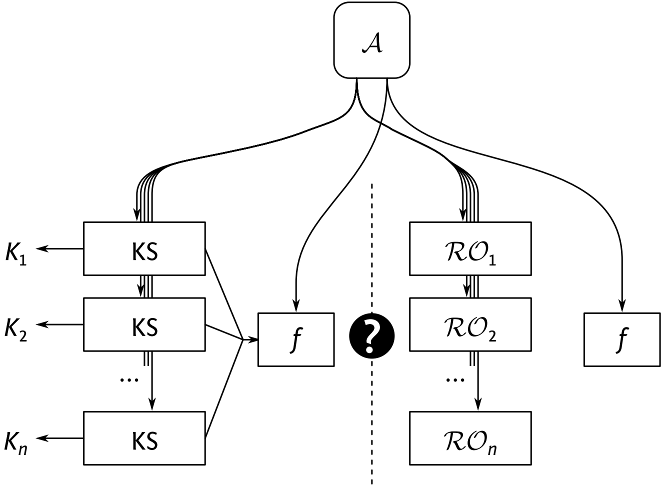
A modular proof approach

- Inner/outer-keyed, multi-target (n), multiplicity μ
- Modular proof using Patarin's H-coefficient technique
- Bound: $M^2/2^{c+1} + \mu N/2^{c-1} + nN/2^k + \ldots$

[Andreeva, Daemen, Mennink, Van Assche, FSE 2015]

Multi-target

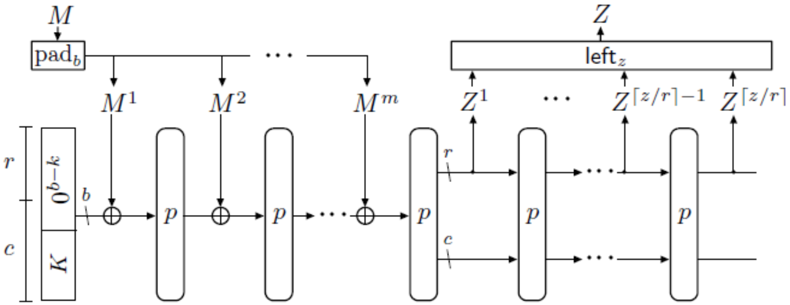
System with n independent keys, damage if any instance is broken



Other proofs

- Security beyond $2^{c/2}$
  [Jovanovic, Luykx, Mennink, Asiacrypt 2014]
- Partially full-state sponge-based AE
  [Sasaki, Yasuda, CT-RSA 2015]
- Full-state keyed sponge (but fixed output size)
  [Gaži, Pietrzak, Tessaro, Crypto 2015]
- Full-state keyed sponge and duplex
  [Mennink, Reyhanitabar, Vizár, Asiacrypt 2015]
- Improved security of the outer keyed sponge
  [Naito, Yasuda, FSE 2016]

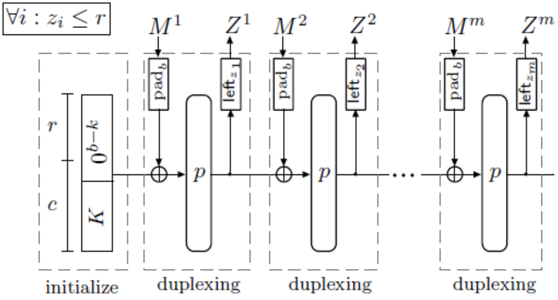
Full-state absorbing!



Absorbing on full permutation width does not degrade bounds

[Mennink, Reyhanitabar, Vizár, Asiacrypt 2015]

Full-state absorbing!



Absorbing on full permutation width does not degrade bounds

[Mennink, Reyhanitabar, Vizár, Asiacrypt 2015]

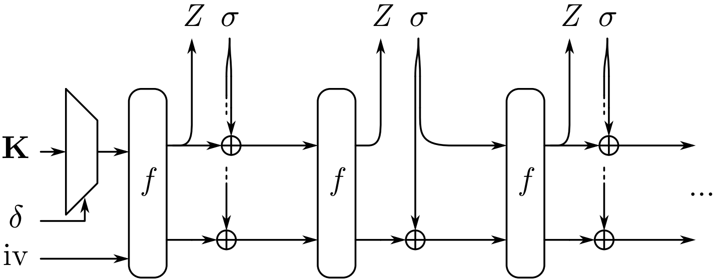
Limitations

But proven bounds had some limitations and problems:

- term $\mu N/2^k$ rather than $\mu N/2^c$
- no multi-key security
- multiplicity μ only known a posteriori

Outline

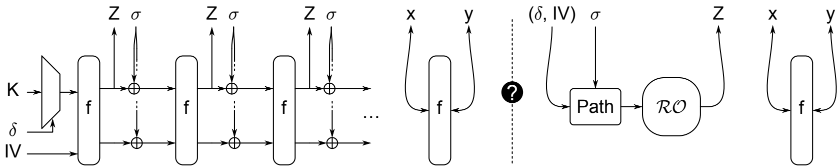
Keyed duplex



- Initial state: concatenation of key $k = \mathbf{K}[\delta]$ and IV
- Full-state absorbing, no padding: $|\sigma| = b$
- Re-phased: f, Z, σ instead of σ, f, Z

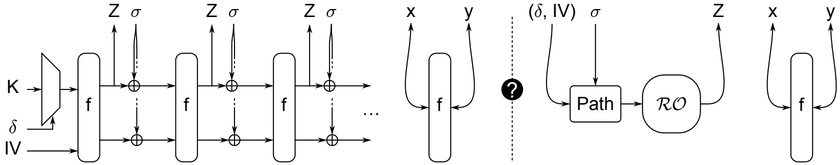
$\approx$ all keyed sponge functions are modes of this

Generic security of keyed duplex: the setup



- Ideal function: Ideal eXtendable Input Function (IXIF)
  - $\mathcal{RO}$-based object with duplex interface
  - Independent outputs Z for different paths
- Further refine adversary's capability
  - L: $\#$ queries to keyed duplex/$\mathcal{RO}$ with repeated path
  - q_{IV} : $\max_{IV} \#$ init queries with different keys

Generic security of keyed duplex: the bound



$$L^2/2^{c+1} + (L + 2\nu)N/2^c + q_{IV}N/2^k + \ldots$$

with ν: chosen such that probability of ν-wise multi-collision in set of M r-bit values is negligible

[Daemen, Mennink, VA, Asiacrypt 2017]

Application: counter-like stream cipher

- Only init calls, each taking Z as keystream block
- IV is nonce, so $L = 0$
- Assume $M \lll 2^{r/2}$: $\nu = 1$

Bound:

$$(2\nu)N/2^c + q_{\mathrm{IV}}N/2^k + \dots$$

Strength:

$$s \leq \min(c - 1, k - \log_2(q_{\mathrm{IV}}))$$

Application: lightweight MAC

- Message padded and fed via IV and σ blocks
- t-bit tag, squeezed in chunks of r bits: $c = b - r$
- adversary chooses IV so $L \approx M = 2^a$
- q_{IV} is total number of keys n

Bound:

$$M^2/2^{c+1} + MN/2^{c-1} + nN/2^k + \ldots$$

Strength:

$$s \leq \min(c - a - 1, k - \log_2(n))$$

Imposes a minimum width of the permutation:

$$b > c > s + a$$

Application: authenticated encryption

Scheme		Parameters			Respecting	Misuse
		b	c	r	Strength	Strength
Ketje	Jr.	200	184	16	$\min\{196-a, 177\}$	$189-a$
	Sr.	400	368	32	$\min\{396-a, 360\}$	$374-a$
Ascon	128	320	256	64	$\min\{317-a, 248\}$	$263-a$
	128a	320	192	128	$\min\{318-a, 184\}$	$200-a$
NORX	32	512	128	384	127	$137-a$
	64	1024	256	768	255	$266-a$
Keyak	River	800	256	544	255	$266-a$
	Lake	1600	256	1344	255	$267-a$

Any questions?

Thanks for your attention!

`https://keccak.team/`

life.augmented