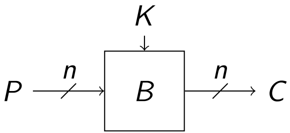
Deck-Based Wide Block Cipher Modes*

Aldo Gunsing, Joan Daemen and Bart Mennink

The Third NIST Workshop on Block Cipher Modes of Operation 2023

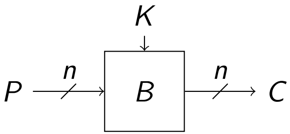
* Contribution is based on the publication *Deck-Based Wide Block Cipher Modes and an Exposition of the Blinded Keyed Hashing Model* at ToSC 2019(4)

Block cipher



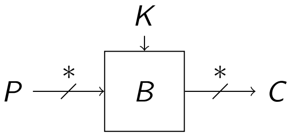
- Plaintext P encrypted to ciphertext C with secret key K
- Fixed block size

Block cipher



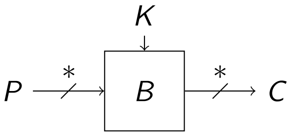
- Plaintext P encrypted to ciphertext C with secret key K
- Fixed block size
- In order to encrypt variable sized messages, we need a mode of operation
  - These modes require a nonce

Wide block cipher



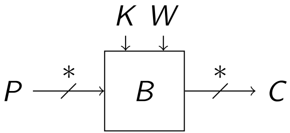
- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size

Wide block cipher



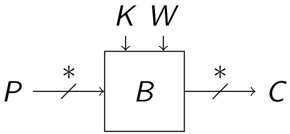
- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size
- Every part of the output (ideally) depends on every part of the input

Tweakable wide block cipher



- A tweakable wide block cipher additionally has a tweak
- Tweak W public, ciphertext completely changes with a different tweak

Tweakable wide block cipher



- A tweakable wide block cipher additionally has a tweak
- Tweak W public, ciphertext completely changes with a different tweak
- Useful for e.g. disk encryption, where every sector gets its own tweak

Our contribution

We build two tweakable wide block ciphers based on three primitives:

Our contribution

We build two tweakable wide block ciphers based on three primitives:

- Doubly-extendable cryptographic keyed (deck) functions:
  - Input: any size
  - Output: arbitrarily long

Our contribution

We build two tweakable wide block ciphers based on three primitives:

- Doubly-extendable cryptographic keyed (deck) functions:
  - Input: any size
  - Output: arbitrarily long
- Stream ciphers:
  - Input: fixed size
  - Output: arbitrarily long

Our contribution

We build two tweakable wide block ciphers based on three primitives:

- Doubly-extendable cryptographic keyed (deck) functions:
  - Input: any size
  - Output: arbitrarily long
- Stream ciphers:
  - Input: fixed size
  - Output: arbitrarily long
- Keyed hashes:
  - Input: any size
  - Output: fixed size

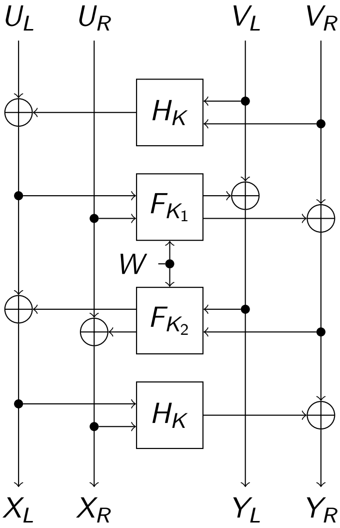
Our contribution

We build two tweakable wide block ciphers based on three primitives:

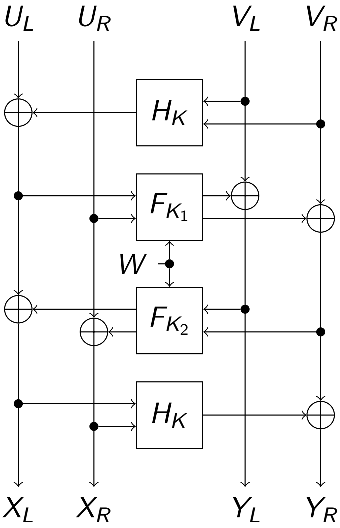
- Doubly-extendable cryptographic keyed (deck) functions:
  - Input: any size
  - Output: arbitrarily long
- Stream ciphers:
  - Input: fixed size
  - Output: arbitrarily long
- Keyed hashes:
  - Input: any size
  - Output: fixed size

In contrast to block ciphers, these primitives are not invertible and do not need to be, which allows for a more flexible design

Double-decker

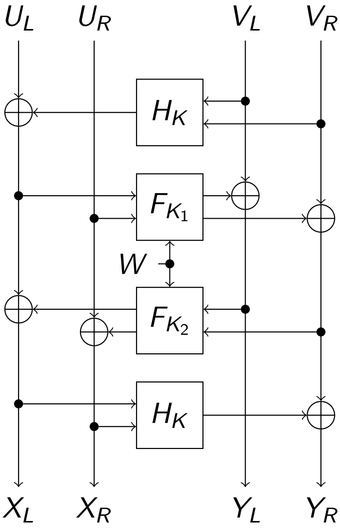


Double-decker



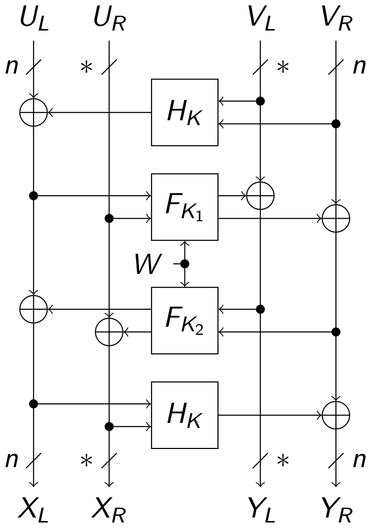
- Generalization of Farfalle-WBC by Bertoni et al. (2017)
- Feistel-like structure

Double-decker



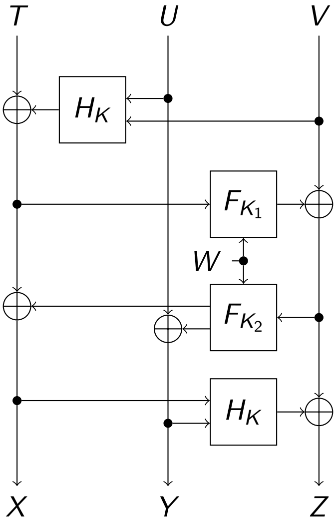
- Generalization of Farfalle-WBC by Bertoni et al. (2017)
- Feistel-like structure
- Two keyed hashes H on the outside, two deck functions F on the inside

Double-decker

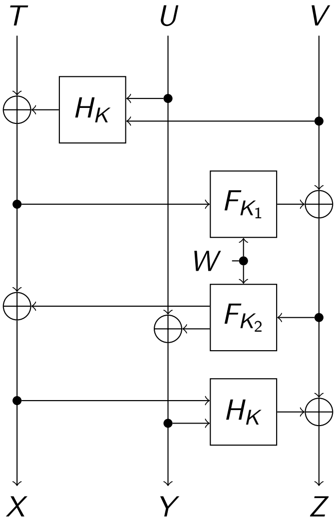


- Generalization of Farfalle-WBC by Bertoni et al. (2017)
- Feistel-like structure
- Two keyed hashes *H* on the outside, two deck functions *F* on the inside
- Outer lanes of fixed size
- Inner lanes of variable size

Docked-double-decker

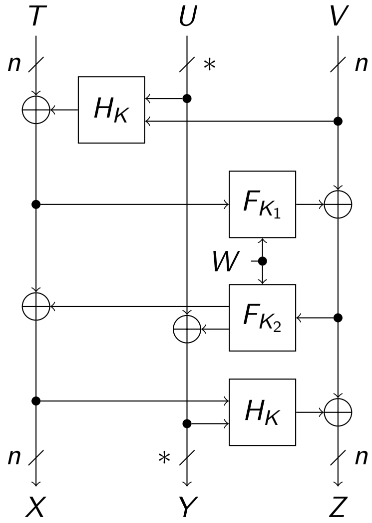


Docked-double-decker



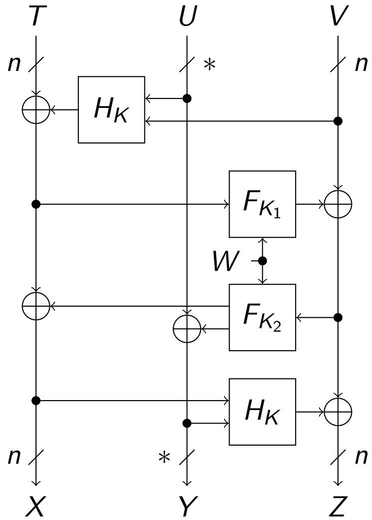
- Variant of double-decker
- One lane less

Docked-double-decker



- Variant of double-decker
- One lane less
- Outer lanes of fixed size
- Inner lane of variable size

Docked-double-decker



- Variant of double-decker
- One lane less
- Outer lanes of fixed size
- Inner lane of variable size
- Deck functions F get fixed sized input, so they conceptually become stream ciphers

XOR-universality

- A keyed hash H is ε-XOR-universal if for all $x \neq x'$ and y

$$\mathbb{P}[H_K(x) \oplus H_K(x') = y] \leqslant \varepsilon$$

XOR-universality

- A keyed hash H is ε-XOR-universal if for all $x \neq x'$ and y

$$\mathbb{P}[H_K(x) \oplus H_K(x') = y] \leqslant \varepsilon$$

- This conventional property only considers the XOR-difference between a single query pair

XOR-universality

- A keyed hash H is ε-XOR-universal if for all $x \neq x'$ and y

$$\mathbb{P}[H_K(x) \oplus H_K(x') = y] \leqslant \varepsilon$$

- This conventional property only considers the XOR-difference between a single query pair

- For q queries the bound becomes $\binom{q}{2}\varepsilon$

XOR-universality

- A keyed hash H is ε-XOR-universal if for all $x \neq x'$ and y

$$\mathbb{P}[H_K(x) \oplus H_K(x') = y] \leqslant \varepsilon$$

- This conventional property only considers the XOR-difference between a single query pair

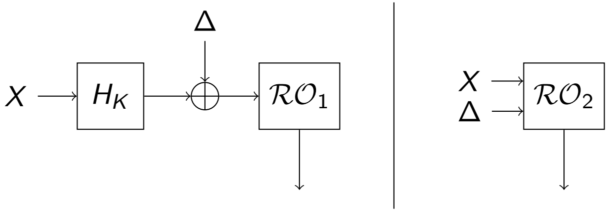
- For q queries the bound becomes $\binom{q}{2}\varepsilon$
- However:
  - ε is the worst-case bound on all possible $x \neq x'$
  - For some functions not all query pairs have similar probabilities

Blinded keyed hash

- We consider blinded keyed hash (bkh) security to achieve a more accurate estimate when multiple queries are taken into account

Blinded keyed hash

- We consider blinded keyed hash (bkh) security to achieve a more accurate estimate when multiple queries are taken into account
- The keyed hash function H is bkh secure if it is indistinguishable in the following setup



Security results

- We cannot apply the bkh model directly to our construction
  - The real difficulty is to reduce to the bkh model
  - For XOR-universality this was trivial

Security results

- We cannot apply the bkh model directly to our construction
  - The real difficulty is to reduce to the bkh model
  - For XOR-universality this was trivial
- We show that the two double-deckers are secure when:
  - The keyed hash H is bkh secure
  - The deck function F is prf secure

Security results

- We cannot apply the bkh model directly to our construction
  - The real difficulty is to reduce to the bkh model
  - For XOR-universality this was trivial
- We show that the two double-deckers are secure when:
  - The keyed hash H is bkh secure
  - The deck function F is prf secure
- Furthermore, by applying the tweak to the deck functions the bound of H becomes tweak-separated
  - Deck functions behave independently for different tweaks
  - Significantly improves security bound for certain settings

Power of tweak-separation

- Consider a ε-XOR-universal keyed hash function H
- Consider q queries and q_W queries with tweak W

loss on H	naive	actual
general bound	$\binom{q}{2}\varepsilon$	
one tweak	$\binom{q}{2}\varepsilon$	
no tweak repetitions	$\binom{q}{2}\varepsilon$	

Power of tweak-separation

- Consider a ε-XOR-universal keyed hash function H
- Consider q queries and q_W queries with tweak W

loss on H	naive	actual
general bound	$\binom{q}{2}\varepsilon$	$\sum_W \binom{q_W}{2}\varepsilon$
one tweak	$\binom{q}{2}\varepsilon$	$\binom{q}{2}\varepsilon$
no tweak repetitions	$\binom{q}{2}\varepsilon$	0

Application to disk encryption on SSDs

- Double-decker is very suitable for disk encryption
  - Disks are separated in sectors
  - Block size is equal to the sector size
  - Physical sector number used as tweak

Application to disk encryption on SSDs

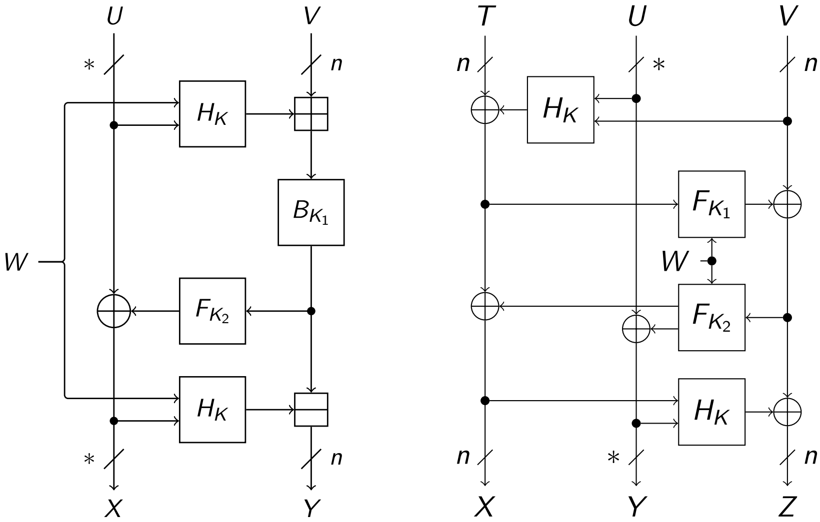
- Double-decker is very suitable for disk encryption
  - Disks are separated in sectors
  - Block size is equal to the sector size
  - Physical sector number used as tweak
- The sectors in SSDs have a limited lifetime as they get damaged every time data is written
- The Kingston UV500 960 GB has $N = 2^{28}$ sectors, where every sector can be written at most ≈ 500 times

Application to disk encryption on SSDs

- Double-decker is very suitable for disk encryption
  - Disks are separated in sectors
  - Block size is equal to the sector size
  - Physical sector number used as tweak
- The sectors in SSDs have a limited lifetime as they get damaged every time data is written
- The Kingston UV500 960 GB has $N = 2^{28}$ sectors, where every sector can be written at most ≈ 500 times

- Without tweak-separation secure when $2\binom{500N}{2}\varepsilon \approx 2^{74}\varepsilon \ll 1$
- With tweak-separation this improves to $2N\binom{500}{2}\varepsilon \approx 2^{46}\varepsilon \ll 1$

Comparison with Adiantum



Adiantum (FSE 2019) Docked-double-decker

Conclusion

- We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions

Conclusion

- We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions
- We also introduced the security model *bkh* for keyed hashes as a generalization of XOR-universality

Conclusion

- We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions
- We also introduced the security model *bkh* for keyed hashes as a generalization of XOR-universality
- Using this model we were able to prove better bounds

Conclusion

- We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions
- We also introduced the security model *bkh* for keyed hashes as a generalization of XOR-universality
- Using this model we were able to prove better bounds
- Our usage of the tweak improves security when tweaks reuse is limited

Conclusion

- We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions
- We also introduced the security model *bkh* for keyed hashes as a generalization of XOR-universality
- Using this model we were able to prove better bounds
- Our usage of the tweak improves security when tweaks reuse is limited

Thank you for your attention!