Deck-Based Wide Block Cipher Modes*

Aldo Gungsin, 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)
In order to encrypt variablesized messages, we need a mode of operation. These modes require an nonce.

Block cipher

- Plaintext $P$ encrypted to ciphertext $C$ with secret key $K$
- Fixed block size
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
A tweakable wide block cipher additionally has a tweak

- Twist \( W \) 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

Generalization of Farfalle-WBC by Bertoniet al. (2017)

Feistel-like structure

Two keyed hash functions $H_{K}$ on the outside, two deck functions $F_{K}$ on the inside

Outer lanes of fixed size

Inner lanes of variable size
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

\[ T \quad U \quad V \]

\[ X \quad Y \quad Z \]

\[ H_K \]

\[ F_{K_1} \]

\[ F_{K_2} \]

\[ H_K \]

\[ \text{Variant of double-decker} \]

\[ \text{Onelaneless} \]

\[ \text{Outerlanes of fixed size} \]

\[ \text{Inner lane of variable size} \]

\[ \text{Deck functions} \]

\[ F \text{ gets fixed sized input, so they conceptually become stream ciphers} \]
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 $\varepsilon$-XOR-universal if for all $x \neq x'$ and $y$

$$\mathbb{P}[H_K(x) \oplus H_K(x') = y] \leq \varepsilon$$
XOR-universality

- A keyed hash $H$ is $\varepsilon$-XOR-universal if for all $x \neq x'$ and $y$
  \[ \mathbb{P}[H_K(x) \oplus H_K(x') = y] \leq \varepsilon \]

- This conventional property only considers the XOR-difference between a single query pair
A keyed hash $H$ is $\varepsilon$-XOR-universal if for all $x \neq x'$ and $y$

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

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

For $q$ queries the bound becomes $\left(\frac{q}{2}\right)\varepsilon$.
A keyed hash $H$ is $\varepsilon$-XOR-universal if for all $x \neq x'$ and $y$

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

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

For $q$ queries the bound becomes $\left(\frac{q}{2}\right) \varepsilon$.

However:

- $\varepsilon$ is the worst-case bound on all possible $x \neq x'$.
- For some functions not all query pairs have similar probabilities.
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.

\[
\begin{align*}
X &\xrightarrow{\Delta} H_K \xrightarrow{\Delta} RO_1 \\
&\quad \Downarrow \\
&\quad \Downarrow \\
X &\xrightarrow{\Delta} RO_2
\end{align*}
\]
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 $\varepsilon$-XOR-universal keyed hash function $H$
- Consider $q$ queries and $q_W$ queries with tweak $W$

<table>
<thead>
<tr>
<th>loss on $H$</th>
<th>naive</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>general bound</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td></td>
</tr>
<tr>
<td>one tweak</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td></td>
</tr>
<tr>
<td>no tweak repetitions</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td></td>
</tr>
</tbody>
</table>
Power of tweak-separation

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

<table>
<thead>
<tr>
<th>loss on $H$</th>
<th>naive</th>
<th>actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>general bound</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td>$\sum_W \left(\frac{q_W}{2}\right)\varepsilon$</td>
</tr>
<tr>
<td>one tweak</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
</tr>
<tr>
<td>no tweak repetitions</td>
<td>$\left(\frac{q}{2}\right)\varepsilon$</td>
<td>0</td>
</tr>
</tbody>
</table>
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 $\approx 500$ 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 \( \approx 500 \) 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.
We introduced (docked-)double-decker, two tweakable wide block ciphers based on deck functions and keyed hash functions.

We also introduced the security model \textit{bkh} for keyed hashes as a generalization of XOR-universality.

Using this model we were able to \textbf{prove better bounds}.

Our usage of the tweak \textbf{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!