Notes on (failed) attempts to instantiate TLR3

Alexander Maximov

Ericsson Research, Lund, Sweden
alexander.maximov@ericsson.com

Abstract. In this short paper we share our experience on instantiating the width-extension construct TLR3, based on a variety of tweakable block cipher constructs. As many of our attempts failed, we highlight the complexity of getting a practical tweakable block cipher and the gap between theory and practice.

Keywords: tweakable block cipher · TLR3 · LRW

1 Introduction

As part of our research we studied a possibility to use an n-bit width block cipher E with n- or $2n$-bit key (e.g., AES-128 or AES-256, where $n = 128$), to create a $2n$-bit width block cipher with a $2n$-bit key. We assume the attacker has 2^{2n} of compute time at his disposal and can make up to $q \approx 2^n$ queries – as it is naturally expected from a $2n$-bit secret permutation. We also wanted to have as little calls to the base E as possible.

We denote a block cipher by $E(k, m)$ or $E(k_0||k_1, m)$ where the key is of size either n or $2n$ bits (depending on the context), and the message m is an n-bit block. Respectively, a tweakable block cipher (TBC) to be denoted by $\tilde{E}(k, t, m)$ or $\tilde{E}(k_0||k_1, t, m)$ where the tweak t is also n-bit long. For the wanted $2n$-bit extended width block cipher we denote the input message by $(L||R)$, the output by $(S||T)$, and the $2n$-bit key by $K = (k_0||k_1)$ – all halves are n-bit long.

At one point, we found the *tweakable Left-Right with 3 rounds* (TLR3) construct [CDMS10] by Coron et.al., which is a Feistel-based design and defined through three independent tweakable block ciphers as follows:

$$X = \tilde{E}1(k, R, L)$$
$$S = \tilde{E}2(k, X, R)$$
$$T = \tilde{E}3(k, S, X)$$

where the key k is only n bits long. For the TLR3 instance, there is a need for ideal TBCs, where the authors say: "*To get an ideal cipher, it suffices to prepend a key k to the 3 ideal ciphers $E1$, $E2$ and $E3$; one then gets a family of independent random permutation, parametrised by k, i.e. an ideal cipher.*" Then the proposed instantiation of TBCs with n-bit secret key $\tilde{E} : \mathcal{K}^n \times \mathcal{T}^n \times \mathcal{M}^n$ is to utilise a base block cipher $E : \mathcal{K}^{2n} \times \mathcal{M}^n$ in a straightforward manner by simply mapping 1-to-1 the set of $\mathcal{K}^n \times \mathcal{T}^n$ in $\tilde{E}$ to $\mathcal{K}^{2n}$ in E as $\tilde{E}(k, t, m) = E(k||t, m)$, and such a construct is a secure TBC up to $\approx 2^n$ queries. The same idea has been used in e.g. the construct TAES in [BGIM19].

The original paper [CDMS10] only has a security proof for TLR3 up to $q \approx 2^{n/2}$, but that bound was significantly improved by Bhaumik et. al. [BNR21], where the security of TLR3 was improved to $(n - 2\log n)$ bits of indifferentiability. In the proof it is considered that ideal tweakable ciphers are utilised. The question about choosing the plurality of tweakable block ciphers $\tilde{E}*$ based on the secret key was not considered in the formal proof for indifferentiability, and it is left out of scope.

However, for a concrete instantiation of a construct that would be keyed, one has to consider the key space that is at least of size of the security level in a formal proof. That is to say, any TLR3 instance with a secret key at least $(n - 2\log n)$ bits would satisfy the [BNR21] security proof of indifferentiability, as long as the underlying TBCs are secure up to at least $2^{(n-2\log n)}$ queries. Otherwise, a simple key exhaustive loop would break the construct faster.

Introduction of a secret key for a concrete instantiation of TLR3 also brings another aspect to the security – the attacker's time that is available. For example, the attacker may have time 2^{2n} but can only make up to 2^n queries, for which the [BNR21] proof then applies. The main obstacle in these two targets is that besides the need for the attacker to spend time for collecting 2^n queries, he also has additional time 2^n to guess parts of the TBC construct and make it "insecure". For example, consider a TBC with $2n$-bit key: $\tilde{E}(k_0||k_1, t, m) = E(k_0 \oplus t||k_1, m)$ – here the attacker guesses k_1 in time 2^n, then he has a TBC with a simple XOR of the n-bit remaining secret half k_0 and the tweak – that has security only up to $2^{n/2}$ queries, which contradicts the [BNR21] proof.

This way, we want to find an n-bit width TBC instance with a $2n$-bit key and an n-bit tweak, where the attacker would have time 2^{2n} (where 2^n is the extra time) with TBC being secure up to (or close to) 2^n queries. Then we could instantiate TLR3 with wanted security levels. A possible way forward to construct a tweakable block cipher $\tilde{E} : \mathcal{K}^{2n} \times \mathcal{T}^n \times \mathcal{M}^n$ is to use another $(3n, n)$ block cipher E' with the key-interface being $3n$-bits long, and similarly to Coron's [CDMS10] and TAES [BGIM19] approach, as follows:

$$\tilde{E}(k_0||k_1, t, m) = E'(k_0||k_1||t, m)$$

But we only have e.g. E =AES-256 which is a $(2n, n)$ block cipher, and here is where the problem starts with.

2 Some failed attempts on TBCs for TLR3

Examples of trivial constructs

A TBC construct may be as simple as $\tilde{E}(k, t, m) = E(k \oplus t, m)$ (e.g., E =AES-128). However, it is only secure for up to $2^{n/2}$ queries in a single-key model, and only 2 queries are needed in the related-key model since $\tilde{E}(k, t, m) = \tilde{E}(k \oplus \delta, t \oplus \delta, m)$.

Another trivial example is given by Simpira authors in Eq.(2) of [GM16] where

$$c = \tilde{E}(k, t, m) := \pi(m \oplus k \cdot t) \oplus k \cdot t.$$

One can ignore t, or simply set $t = 1$. Then consider two related pairs (m, k) and (m', k') such that $(m \oplus k) = (m' \oplus k')$, then the outputs are heavily related $c \oplus c' = k \oplus k'$. In an ideal keyed block cipher we want that for a distinct k we get a pseudo-random permutation/mapping on all 2^{2n} input values (t, m), which is not the case here in this construct as k and m are heavily correlated.

LRW1 construct as given in Theorem 1

Liskov et. al. propose two TBC constructs secure up to $q \approx 2^{n/2}$ [LRW02]. We call these two constructs as LRW1 and LRW2 followed by the names of the authors. Let us try to build a TLR3 instance with these secure TBCs, and we would even use three independent $n/2n$-bit subkeys k_1, k_2, k_3, e.g. derived from the master $2n$-bit key in some way. The analysis that we sketch below seems to show a general weakness of a TBC construct where the tweak t is not well mixed with the key k nor an independent part of the key to the base building block.

Theorem 1 of [LRW02] defines the following TBC: $\tilde{E}(k,t,m) = E(k, t \oplus E(k,m))$, which has the property that $\tilde{E}(k,t,m) \neq \tilde{E}(k, t \oplus \delta, m)$ and this can be distinguished with $q \approx 2^{n/2}$ queries. When using the above construct as the building block for TLR3, we get the following instance:

$$\begin{aligned} X &= E(k_1, R \oplus E(k_1, L)) \\ T &= E(k_2, X \oplus E(k_2, R)) \\ S &= E(k_3, T \oplus E(k_3, X)) \end{aligned}$$

Note that here all three instances $E(k_i, m)$ may have distinct keys each of which is of length n or $2n$ bits. For an attack, collect $q \approx 2^{n/2}$ queries (L^i, R^i) such that $R^i = R^j = c$ and $L^i \neq L^j$, which implies:

$$\begin{aligned} X^i = E(k_1, c \oplus E(k_1, L^i)) \quad &\Rightarrow \quad X^i \neq X^j \\ T^i = E(k_2, X^i \oplus E(k_2, c)) \quad &\Rightarrow \quad T^i \neq T^j \end{aligned}$$

Due to the birthday paradox we would expect a pair of queries i and j where $T^i = T^j$, but this will never happen as seen in above derivations, thus we have a distinguishing attack of complexity $q \approx 2^{n/2}$.

LRW2 construct as given in Theorem 2

A seemingly more secure TBC is given as $\tilde{E}(k,t,m) = E(k, m \oplus h(t)) \oplus h(t)$, where $h()$ is a hash function. Applying Theorem 2 of [LRW02] for TLR3 we get the following instance:

$$\begin{aligned} X &= E(k_1, L \oplus h(R)) \oplus h(R) \\ T &= E(k_2, R \oplus h(X)) \oplus h(X) \\ S &= E(k_3, X \oplus h(T)) \oplus h(T) \end{aligned}$$

An attacker collects $2^{n/2}$ queries (L^i, R^i) such that $R^i \neq R^j$ and $L^i = h(R^i)$, which implies:

$$\begin{aligned} X^i &= c \oplus h(R^i), \quad \text{where the constant } c = E(k_1, 0) \\ S^i &= E(k_3, c \oplus h(R^i) \oplus h(T^i)) \oplus h(T^i) \end{aligned}$$

Due to the birthday paradox we should find a pair of queries i and j where $h(R^i) \oplus h(T^i) = h(R^j) \oplus h(T^j)$. Then we take that pair and check that $S^i \oplus h(T^i) = S^j \oplus h(T^j)$ with probability 1, while in a random construction that match should happen with probability 2^{-n}. This results in a distinguishing attack of complexity $q \approx 2^{n/2}$.

Yet another example when the tweak is not mixed with the key

Let us have a $2n$-bit key $(k_0||k_1)$ and we construct TBC as $E(k_0, (k_1 \oplus t) \oplus E(k_0, m))$. The TLR3 instance is then:

$$\begin{aligned} X &= E(k_0, k_1 \oplus R \oplus E(k_0, L)) \\ T &= E(k_0, k_1 \oplus X \oplus E(k_0, R)) \\ S &= E(k_0, k_1 \oplus T \oplus E(k_0, X)) \end{aligned}$$

For an attack we collect $q \approx 2^{n/2}$ queries such that $R^i = R^j = c$ and $L^i \neq L^j$ which implies:

$$\begin{aligned} X^i = E(k_0, k_1 \oplus c \oplus E(k_0, L^i)) \quad &\Rightarrow \quad X^i \neq X^j \\ T^i = E(k_0, k_1 \oplus X^i \oplus E(k_0, c)) \quad &\Rightarrow \quad T^i \neq T^j \end{aligned}$$

I.e., we never see $T^i = T^j$ among the chosen set of inputs, thus we have a distinguishing attack of complexity $q \approx 2^{n/2}$.

When the tweak is mixed with the key, but not good enough

Let us now try to mix the tweak with the key in a linear way and create TLR3 subkeys as $(k_0||k_1) \oplus (t||t) \oplus i$, where i serves as a constant in order to force the three subkeys to be distinct. Then we have:

$$\begin{aligned} X &= E((k_0||k_1) \oplus \text{`0'} \oplus (R||R), L) \\ T &= E((k_0||k_1) \oplus \text{`1'} \oplus (X||X), R) \\ S &= E((k_0||k_1) \oplus \text{`2'} \oplus (T||T), X) \end{aligned}$$

The three secret permutations will be distinct due to distinct subkeys, and the tweak plays the role of selecting the plurality of permutations, along with the secret key.

As a related key attack (RKA) example in a combination with a distinguishing attack, let us have $2^{n/2}$ related pairs $((k_0||k_1)^i, R^i)$ such as

$$\begin{aligned} (k_0||k_1)^i &= (k_0 \oplus i||k_1 \oplus i)^0 \oplus (i||i) \\ R^i &= R^0 \oplus i \end{aligned}$$

for some random initial $(k_0||k_1)^0$ and R^0. Then, the subkey for the first E will be

$$\texttt{subkey1}^i = (k_0||k_1)^i \oplus (R^i||R^i) = (k_0||k_1)^0 \oplus (i||i) \oplus (R^0 \oplus i||R^0 \oplus i) = (k_0||k_1)^0 \oplus (R^0||R^0) = c$$

i.e., some constant value c for all $2^{n/2}$ related pairs $((k_0||k_1)^i, R^i)$. Then we make $q \approx 2^{n/2}$ queries of the form $(L, R)^i = (0, R^i)$, and for the ith query we apply TLR3 with the key $(k_0||k_1)^i$. Since now $L^i = L^j = 0$ for all queries and the first subkey also coincides, then we get that for all $2^{n/2}$ queries the first E outputs the same value, i.e. $X^i = c'$.

The attacker observes $2^{n/2}$ output pairs (S^i, T^i). Among the output pairs, one can find such pair (i, j) where $T^i \oplus i = T^j \oplus j$. At least one such pair should exist among $2^{n/2}$ queries due to the birthday paradox. Let $W = T^i \oplus i$ for simplicity. For such a "specially obtained" pair we derive:

$$\begin{aligned} \texttt{subkey3}^i &= (k_0, k_1)^i \oplus \text{`2'} \oplus (T^i||T^i) = (k_0, k_1)^0 \oplus (i||i) \oplus \text{`2'} \oplus (W||W) \oplus (i||i) \\ &= (k_0, k_1)^0 \oplus \text{`2'} \oplus (W||W) \\ \texttt{subkey3}^j &= (k_0, k_1)^j \oplus \text{`2'} \oplus (T_j||T_j) = (k_0, k_1)^0 \oplus (j||j) \oplus \text{`2'} \oplus (W||W) \oplus (j||j) \\ &= (k_0, k_1)^0 \oplus \text{`2'} \oplus (W||W) \end{aligned}$$

i.e., we get a matching $X^i = X^j$ and also matching 3rd subkeys. Thus, the resulting S^i and S^j must coincide with probability 1, while in a pure random case it should happen with probability 2^{-n}. This example demonstrates how having $2^{n/2}$ related keys we can have a distinguishig attack of complexity $q \approx 2^{n/2}$.

Yet another linear mixing

One more attempt was made to construct subkeys as $(k_0||k_1) \oplus (t||0) \oplus i$, by adding the tweak to only one half of the $2n$-bit key. Then we have:

$$\begin{aligned} X &= E((k_0||k_1) \oplus \text{`0'} \oplus (R||0); \quad L) \\ T &= E((k_0||k_1) \oplus \text{`1'} \oplus (X||0); \quad R) \\ S &= E((k_0||k_1) \oplus \text{`2'} \oplus (T||0); \quad X) \end{aligned}$$

For an attack, let us have $2^{n/2}$ related pairs $((k_0||k_1)^i, R^i)$ such as

$$\begin{aligned} (k_0||k_1)^i &= (k_0||k_1)^0 \oplus (i||0) \\ R^i &= R^0 \oplus i \end{aligned}$$

for some random initial $(k_0||k_1)^0$ and R^0. Then the subkey for the first E is

$$\texttt{subkey1}^i = (k_0||k_1)^i \oplus (R^i||0) = (k_0||k_1)^0 \oplus (i||0) \oplus (R^0 \oplus (i||0)) = (k_0||k_1)^0 \oplus (R_0||0) = c$$

i.e., some constant value c for all $2^{n/2}$ related pairs $((k_0||k_1)^i, R^i)$. We make $q \approx 2^{n/2}$ queries of the form $(L, R)^i = (0, R^i)$, and for the ith query we apply TLR3 with the key $(k_0||k_1)^i$. Since now $L^i = L^j = 0$ for all queries and the first subkey also coincides, all outputs from the first E is a constant $X^i = c'$ for all $2^{n/2}$ queries.

The attacker can observe $2^{n/2}$ output pairs (S^i, T^i). Among the output pairs, one can find such pair (i, j) where $T^i \oplus i = T^j \oplus j$. At least one such pair should exist among $2^{n/2}$ queries due to the birthday paradox. Let $W = T^i \oplus i$, and for such a "special" pair we get:

$$\begin{aligned}
\texttt{subkey3}^i &= (k_0||k_1)^i \oplus \text{`2'} \oplus (T^i||0) = (k_0||k_1)^0 \oplus (i||0) \oplus \text{`2'} \oplus (W||0) \oplus (i||0) \\
&= (k_0||k_1)^0 \oplus \text{`2'} \oplus (W||0) \\
\texttt{subkey3}_j &= (k_0||k_1)^j \oplus \text{`2'} \oplus (T_j||0) = (k_0||k_1)^0 \oplus (j||0) \oplus \text{`2'} \oplus (W||0) \oplus (j||0) \\
&= (k_0||k_1)^0 \oplus \text{`2'} \oplus (W||0)
\end{aligned}$$

i.e., we get matching $X^i = X^j$ and also matching 3rd subkeys. Thus, the resulting S^i and S^j must coincide with probability 1, while in a pure random case it should be 2^{-n}. The attack is similar to the previous case and could be generalised to when the tweak and the key are mixed in a linear way.

Nonlinear mixing of the tweak and the key

A preferable case for security is when there exists a security proof for beyond birthday bound (BBB) with the number of queries closer to 2^n. An example of such a constructs, thought with only n-bit keys, is Mennink's $\tilde{F}[2](k, t, m) = E(k \oplus t, m \oplus z) \oplus z$ with $z = E(2k, t)$ [Men15], where the nonlinear mixing of the tweak and the key happens in z. In [WGZ+16] it was shown that the original design of $\tilde{F}[2]$ where $z = E(k, t)$ is without multiplication, has security of only $2^{n/2}$ queries when the padding and encryption uses the same key of the tweakable block cipher and $t = 0$. The patched design with "$2k$" seems now more secure but that multiplication happens in $GF(2^n)$ and there are still certain weak combinations of keys and tweaks when, for example, $k \oplus t \equiv 2k$ in $GF(2^n)$.

On the contrary, the authors of [WGZ+16] analysed 32 TBC constructs $\tilde{E}1..\tilde{E}32$, such as $\tilde{E}12(k, t, m) = E(k \oplus t, m \oplus y) \oplus y$, where $y = E(0, k)$, with the claimed full security 2^n. Here we would favour $\tilde{E}12$ as an interesting and practical case since y can be computed only once, as it has also been noted in [Men20].

However, the above schemes are only secure for n-bit keys and when a $2n$-bit security is needed (i.e., when the attacker has an extra 2^n of time) for TLR3 instantiation these TBCs seem not secure enough. Just consider the attacker can simply guess n bits of the key while still having 2^n of time to perform the remaining part of, for example, a complete key recovery. Moreover, there is no gain in $\tilde{E}12$ for the case of TLR3 when there $(2n, n)$ block ciphers are already available – as mentioned earlier an alternative would then be simply $E(k||t, m)$.

3 Conclusions

Likewise with AES-128 and AES-256, we would like to have a secure tweakable block cipher that can accept a secret key up to $2n$ bits, while the block width and the tweak itself can be n bits. Some practical tweakable ciphers with full security that we considered support only n-bit keys and thus the attacker only has 2^n of time. A block cipher with $(3n, n)$ parameters would be a perfect solution to a tweakable block cipher $(2n, n, n)$, which

may in turn be used as a building block for other constructs, whether it is a block width or key length extension, or other applications. Standardising a $(2n, 2n)$ block cipher such as a 256-bit width Rijndael-256 would be a perfect building block for relatively simple external extension schemes (also, for constructs such as tweakable block ciphers) that would in turn easily reach security up to 2^n queries, and, perhaps, an even higher security on the cost of more complex constructs.

Any design of a tweakable block cipher should clearly state it's security model (e.g., whether it is secure in KPA, CPA, CCA, RKA, indistinguishability or indifferentiability models, etc), and the number of queries as well as the attacker's time are also crucial parameters to take into consideration, while many security proofs omit the latter and simply assume ideal building blocks. It might also be noted that a secret permutation selected by both the key and the tweak should be done in a pseudo-random fashion, and there should not be any obvious relation between the key, the tweak, and the message.

Finally, larger constructs using tweakable block ciphers as a building block should also specify how distinct subkeys to be derived from a master key, and whether they should be of size n or $2n$ bits. For example, TLR3 assumes three *independent* TBCs, which implies an additional complexity for obtaining these subkeys.

References

[BGIM19] Zhenzhen Bao, Jian Guo, Tetsu Iwata, and Kazuhiko Minematsu. ZOCB and ZOTR: Tweakable blockcipher modes for authenticated encryption with full absorption. *IACR Trans. Symm. Cryptol.*, 2019(2):1–54, 2019.

[BNR21] Ritam Bhaumik, Mridul Nandi, and Anik Raychaudhuri. Improved indifferentiability security proof for 3-round tweakable luby-rackoff. 89(10):2255–2281, 2021.

[CDMS10] Jean-Sébastien Coron, Yevgeniy Dodis, Avradip Mandal, and Yannick Seurin. A domain extender for the ideal cipher. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 273–289. Springer, Heidelberg, February 2010.

[GM16] Shay Gueron and Nicky Mouha. Simpira v2: A family of efficient permutations using the AES round function. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 95–125. Springer, Heidelberg, December 2016.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer, Heidelberg, August 2002.

[Men15] Bart Mennink. Optimally secure tweakable blockciphers. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 428–448. Springer, Heidelberg, March 2015.

[Men20] Bart Mennink. Beyond birthday bound secure fresh rekeying: Application to authenticated encryption. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part I*, volume 12491 of *LNCS*, pages 630–661. Springer, Heidelberg, December 2020.

[WGZ+16] Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, and Dawu Gu. How to build fully secure tweakable blockciphers from classical blockciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 455–483. Springer, Heidelberg, December 2016.