Dobbertin Challenge 2022

Ruhr University Bochum, Workgroup for Symmetric Cryptography

February 11, 2022

1 A Tweakable AES Variant

A *tweakable block cipher* [3] with block size n, tweak size τ, and key size κ is defined as a family of permutations on $\mathbb{F}_2^n$ parameterized by $(t, k) \in \mathbb{F}_2^\tau \times \mathbb{F}_2^\kappa$. Compared to a usual block cipher, i.e., a family of permutation parameterized by the key k, a tweakable block cipher gets as an additional (public) input t, which is called the *tweak*. One particular use case for a tweakable block cipher is memory encryption [2]. The idea is that each block is encrypted under a fixed key k and a variable tweak $t^{(i)}$, where $t^{(i)}$ encodes the memory address.

In this challenge we propose a tweakable variant of the well-known Advanced Encryption Standard (AES). For the sake of completeness, we briefly recall the definition of the AES round function. For further details, we refer to the book by Daemen and Rijmen [1].

1.1 Description of the AES

The AES is a family of block ciphers with a block size of $n = 128$ bits, supporting three different key sizes of 128, 192, and 256 bits. In this challenge, we concentrate on the AES variant with a 128-bit key. For each fixed key, the AES operates as a permutation on $\mathbb{F}_2^{128}$. For a simpler description of the algorithm, we represent the AES as a family of permutations on $\mathbb{F}_{2^8}^{4\times 4}$. The internal state can then be described by a 4×4 array with elements in $\mathbb{F}_{2^8}$ (also called *cells*) as

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}.$$

The unkeyed round function $\mathsf{R}\colon (\mathbb{F}_{2^8})^{4\times 4} \to (\mathbb{F}_{2^8})^{4\times 4}$ of AES is defined as the composition of the operations SubBytes, ShiftRows and MixColumns such that

$$\mathsf{R} = \mathsf{MixColumns} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes}\,.$$

The functions on the right-hand side are defined as follows.

SubBytes is a parallel application of the 8-bit AES S-box $S\colon \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ to all 16 cells of the internal state. We refer to [1] for the definition of S.

ShiftRows cyclically rotates the i^{th} row of the state i positions to the left, for all $i \in \{0, 1, 2, 3\}$.

MixColumns multiplies the columns of the state with a matrix M. Again, we refer to [1] for the definition of M.

The unkeyed AES rounds are interleaved by the (XOR) addition of a round key. The latter operation will be denoted by $\mathsf{Add}_{k_i}\colon x \mapsto x \oplus k_i$. The i^{th} round function of the AES is then given by

$$\mathsf{R}_i = \mathsf{Add}_{k_i} \circ \mathsf{R}\,.$$

The round keys k_i are generated from the 128-bit master key k by the AES key schedule, i.e., we have $(k_0, k_1, \ldots, k_{10}) = \mathsf{KeySchedule}(k)$. We refer to [1] for the specification of the function $\mathsf{KeySchedule}$. With the above notation, the AES variant with a 128-bit key can then be described as

$$\mathsf{AES}_k = \mathsf{Add}_{k_{10}} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes} \circ \mathsf{R}_9 \circ \cdots \circ \mathsf{R}_2 \circ \mathsf{R}_1 \circ \mathsf{Add}_{k_0}\,,$$

where $(k_0, k_1, \ldots, k_{10}) = \mathsf{KeySchedule}(k)$.

1.2 Specification of TweakableAES

In this section, we define $\mathsf{TweakableAES}$, i.e., a tweakable variant of the AES. The block, tweak, and key size is $n = 128$, $\tau = 128$, and $\kappa = 80$ bits, respectively. The (unkeyed, untweaked) round function of $\mathsf{TweakableAES}$ is identical to that of the AES. We only define an alternative key schedule and introduce the (XOR) addition of *round tweaks*.

Key schedule. Let $k \in \mathbb{F}_2^{80}$ be the 80-bit master key. The 64-bit values $k'_0, \ldots, k'_{10} \in \mathbb{F}_2^{64}$ are derived from the master key k as follows: For $i \in \{0, \ldots, 10\}$, we have

$$k'_i = \mathsf{Truncate}(\mathsf{Update}^i(k)),$$

where the function Update first applies the permutation $\pi = (0, 4, 5, 8, 9, 1, 3, 2, 6, 7)$ to the 10 bytes of the input and then applies the AES S-box to the first (i.e., most significant) byte, and where the function $\mathsf{Truncate}$ takes the first 8 (most significant) bytes of the 10-byte value.

The actual round keys used in our design are then equal to $k_0, \ldots, k_{10}$, where the i^{th} round key k_i is defined by

$$k_i = \begin{bmatrix} k'_{i,0} \oplus c_{i,0} & k'_{i,4} \oplus c_{i,4} & k'_{i,0} \oplus c_{i,8} & k'_{i,4} \oplus c_{i,12} \\ k'_{i,1} \oplus c_{i,1} & k'_{i,5} \oplus c_{i,5} & k'_{i,1} \oplus c_{i,9} & k'_{i,5} \oplus c_{i,13} \\ k'_{i,2} \oplus c_{i,2} & k'_{i,6} \oplus c_{i,6} & k'_{i,2} \oplus c_{i,10} & k'_{i,6} \oplus c_{i,14} \\ k'_{i,3} \oplus c_{i,3} & k'_{i,7} \oplus c_{i,7} & k'_{i,3} \oplus c_{i,11} & k'_{i,7} \oplus c_{i,15} \end{bmatrix}, \text{ for } i = 0, \ldots, 9 \quad \text{and}$$

$$k_{10} = \begin{bmatrix} k'_{10,0} & k'_{10,4} & 0 & 0 \\ k'_{10,1} & k'_{10,5} & 0 & 0 \\ k'_{10,2} & k'_{10,6} & 0 & 0 \\ k'_{10,3} & k'_{10,7} & 0 & 0 \end{bmatrix},$$

with $k'_{i,0}, \ldots, k'_{i,7}$ being the bytes (from lsb to msb) of k'_i and $c_{i,0}, \ldots, c_{i,15}$ being the bytes (from lsb to msb) of the round constant c_i. The round constants $c_0, \ldots, c_9 \in \mathbb{F}_2^{128}$ are defined as follows:

$$\begin{aligned} c_0 &= \texttt{0x13dab5de9f61d0fe2bff63a0efaf0e90} \\ c_1 &= \texttt{0xdb60e2c38577700434ca45fd2440341c} \\ c_2 &= \texttt{0x943590b7043c164ac10c65bebe1cd6e8} \\ c_3 &= \texttt{0x40d75fc4ff599a468f7f1215f95c8b81} \\ c_4 &= \texttt{0x2f5a3cf103adb5c46d34b4c12b238ddb} \\ c_5 &= \texttt{0x9fbe7d0dc2b0bb20ebfffc49b370446b} \\ c_6 &= \texttt{0x94f17a38ee22e8db98fb59b6e9d8b7ef} \\ c_7 &= \texttt{0x78a5aa73fd62877629b9da4c2f9b2711} \\ c_8 &= \texttt{0x630a48dedddc7bf2808bb64910df4607} \\ c_9 &= \texttt{0x1c00bfbcefb8827013f401d4f65728d7}. \end{aligned}$$

Tweak schedule. Let $t \in \mathbb{F}_2^{128}$ be the 128-bit master tweak. The 64-bit values $t'_0, \ldots, t'_9 \in \mathbb{F}_2^{64}$ are derived from the master tweak t using a linear function L, i.e., $(t'_0, \ldots, t'_9) = \mathsf{L}(t)$. More precisely, we take t'_i as the 64 least significant bits of $t \lll (11 \cdot (i+1))$. The round tweaks are then equal to $t_0, \ldots, t_9$ where t_i is defined by

$$t_i = \begin{bmatrix} 0 & 0 & t'_{i,0} & t'_{i,4} \\ 0 & 0 & t'_{i,1} & t'_{i,5} \\ 0 & 0 & t'_{i,2} & t'_{i,6} \\ 0 & 0 & t'_{i,3} & t'_{i,7} \end{bmatrix},$$

with $t'_{i,0}, \ldots, t'_{i,7}$ being the bytes (from lsb to msb) of t'_i.

Overall structure. The i^{th} round function is defined by $\mathsf{R}'_i = \mathsf{Add}_{k_i \oplus t_i} \circ \mathsf{R}$ and the tweakable block cipher $\mathsf{TweakableAES}$ can then be described as

$$\mathsf{TweakableAES}_{k,t} = \mathsf{Add}_{k_{10}} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes} \circ \mathsf{R}'_9 \circ \cdots \circ \mathsf{R}'_1 \circ \mathsf{Add}_{k_0 \oplus t_0} .$$

2 The Challenge

A secret message block msg was encrypted with $\mathsf{TweakableAES}$ using a tweak t^* and the secret key k^*. You only observe the ciphertext $cph = \mathsf{TweakableAES}_{t^*,k^*}(msg)$ and the tweak that was used. Additionally, you can query the encryption of arbitrary plaintexts under arbitrary tweaks, i.e., you can choose $x \in \mathbb{F}_2^{128}$ and $t \in \mathbb{F}_2^{128}$ and query $\mathsf{TweakableAES}_{t,k^*}(x)$. Your challenge is to recover msg.

References

[1] Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

[2] Michael Henson and Stephen Taylor. Memory encryption: A survey of existing techniques. *ACM Comput. Surv.*, 46(4), mar 2014.

[3] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. *J. Cryptol.*, 24(3):588–613, 2011.