Optimally Secure Tweakable Blockciphers

Bart Mennink

Dept. Electrical Engineering, ESAT/COSIC, KU Leuven, and iMinds, Belgium
bart.mennink@esat.kuleuven.be

Abstract. We consider the generic design of a tweakable blockcipher from one or more evaluations of a classical blockcipher, in such a way that all input and output wires are of size n bits. As a first contribution, we show that any tweakable blockcipher with one primitive call and arbitrary linear pre- and postprocessing functions can be distinguished from an ideal one with an attack complexity of about $2^{n/2}$. Next, we introduce the tweakable blockcipher $\widetilde{F}[1]$. It consists of one multiplication and one blockcipher call with tweak-dependent key, and achieves $2^{2n/3}$ security. Finally, we introduce $\widetilde{F}[2]$, which makes two blockcipher calls, one of which with tweak-dependent key, and achieves optimal 2^n security. Both schemes are more efficient than all existing beyond birthday bound tweakable blockciphers known to date, as long as one blockcipher key renewal is cheaper than one blockcipher evaluation plus one universal hash evaluation.

Keywords. Tweakable blockcipher, Liskov-Rivest-Wagner, optimal security, beyond birthday bound.

1 Introduction

A blockcipher is a family of permutations indexed via a secret key. Tweakable blockciphers generalize over classical blockciphers by introducing the *tweak* as an additional parameter. More formally, a tweakable blockcipher $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ is a family of permutations on $\mathcal{M}$ indexed by a key $k \in \mathcal{K}$ and tweak $t \in \mathcal{T}$. Here, the key input is a secret parameter to guarantee security, while the tweak value is a public parameter with the main purpose to bring flexibility to the cipher. Tweakable blockciphers were formalized by Liskov, Rivest, and Wagner [29] and find a wide spectrum of applications, such as tweakable enciphering schemes [8, 13, 20–22, 34, 45, 49], authenticated encryption schemes and message authentication codes [2, 27, 41, 42], and online ciphers [2, 44].

Example tweakable blockciphers that admit tweaks by design are Schroeppel's Hasty Pudding Cipher [46], Crowley's Mercy [10], and the Threefish cipher used in SHA-3 finalist Skein [14]. Furthermore, Goldenberg et al. [18] demonstrated how to transform a Feistel scheme into a tweakable Feistel scheme that achieves birthday bound security, and Mitsuda and Iwata [35] derived similar results for generalized Feistel schemes. Jean et al. [23] considered the problem of tweaking key alternating ciphers by presenting TWEAKEY, a construction that elegantly blends the tweak with the key in the key scheduling algorithm.

A more generic approach is to design a tweakable blockcipher from an ordinary blockcipher (and possibly other cryptographic primitives) in a black-box way. Two such constructions were introduced in Liskov et al.'s original paper. The first construction $\mathsf{LRW1}$ makes two evaluations of an underlying blockcipher E, while the other construction $\mathsf{LRW2}$ is based on a blockcipher E and a universal hash function family H:

$$\mathsf{LRW1}(k,t,m) = E(k, E(k,m) \oplus t)\,, \tag{1}$$

$$\mathsf{LRW2}([k,h],t,m) = E(k, m \oplus h(t)) \oplus h(t)\,, \tag{2}$$

where $h \in H$. These constructions achieve security up to the birthday bound. Related to $\mathsf{LRW2}$ is the XEX construction by Rogaway [41], and extensions of it by Chakraborty and Sarkar [7] and Minematsu [32], which effectively reduces the keyspace to n bits.

Landecker, Shrimpton, and Terashima [27] considered the cascade of two $\mathsf{LRW2}$'s:

$$\mathsf{LRW2}[2]([k_1,k_2,h_1,h_2],t,m) = \mathsf{LRW2}([k_2,h_2],t,\mathsf{LRW2}([k_1,h_1],t,m))\,, \tag{3}$$

and proved it secure up to about $2^{2n/3}$ queries.[1] Lampe and Seurin [26] generalized this approach and considered a cascade of $\rho \geq 1$ evaluations:

$$\mathsf{LRW2}[\rho]([\mathbf{k},\mathbf{h}],t,m) = \mathsf{LRW2}([k_\rho,h_\rho],t,\cdots\mathsf{LRW2}([k_1,h_1],t,m)\cdots)\,, \tag{4}$$

where $\mathbf{k} = (k_1,\dots,k_\rho)$ are blockcipher keys and $\mathbf{h} = (h_1,\dots,h_\rho)$ instantiations of H. Lampe and Seurin proved that for even ρ, this construction is secure up to approximately $2^{\rho n/(\rho+2)}$ queries. Note that this bound only improves over the one of Landecker et al. for $\rho \geq 4$. Lampe and Seurin conjectured that their bound could be improved to $2^{\rho n/(\rho+1)}$. This term approaches the optimal 2^n for increasing ρ, but also the number of primitive calls and the key size increases linearly in ρ.

Tweak-Dependent Keys

Liskov et al. [29] suggested that a change in the tweak should be cheaper than a change in the key. As pointed out by Jean et al. [23], this may seem somewhat counter-intuitive because the adversary has full control over the tweak while it has only limited to no control over the key. They suggest that, in practice, the two inputs should be treated comparably. Additionally, the theoretical quest to derive an (almost) optimally secure tweakable blockcipher complying with this condition lead to an unrestrained increase of primitive calls and of the number of keys.

For example, the tweakable blockcipher $\widetilde{E}(k,t,m) = E(k \oplus t, m)$ is secure up to about $2^{n/2}$ evaluations (in the single-key setting,[2] and if the underlying

[1] Procter [39] pointed out a flaw in the original proof and suggested a fix. See also the ePrint version of [27].

[2] In the related-key model we have $\widetilde{E}(k,t,m) = \widetilde{E}(k\oplus\delta, t\oplus\delta, m)$ for any (k,t,m) and any δ [23].

cipher is sufficiently secure), and thus achieves the same level of security as, for instance, $\mathsf{LRW1}$. If we assume that the underlying cipher E consists of a key scheduling part and a message encryption part (such separation is easily made for key alternating ciphers), each evaluation of $\widetilde{E}$ requires one key scheduling and one message encryption, while each evaluation of $\mathsf{LRW1}$ requires two message encryptions (the key scheduling can be pre-computed). This means that $\widetilde{E}$ is more efficient than $\mathsf{LRW1}$ if the key scheduling part of E is cheaper than its message encryption part.

Minematsu [33] presented a construction of a tweakable blockcipher with tweak-dependent key that achieves beyond birthday bound security. In more detail, he proved that

$$\mathsf{Min}(k, t, m) = E(E(k, t \| 0^{n-|t|}), m) \tag{5}$$

is secure up to $\max\{2^{n/2}, 2^{n-|t|}\}$ where $|t|$ denotes the fixed tweak length. Unfortunately, this construction only achieves beyond birthday bound security as long as the tweak is shorter than $n/2$ bits and it can impossibly achieve optimal 2^n security (unless $|t| = 0$). Beyond Minematsu's scheme, no other tweakable blockciphers in this direction are known.[3]

Our Contributions

We investigate the following elementary question. *Can we design an optimally secure tweakable blockcipher $\widetilde{E}$ with n-bit in- and outputs using only a blockcipher E with n-bit in- and outputs?*

We approach this question generically, focusing on the way $\widetilde{E}$ is designed from E, which means that the preprocessing functions that prepare the inputs to the underlying blockcipher may be technically any function as long as the tweakable blockcipher itself is invertible. This also means that the preprocessing functions may utilize another cryptographic primitive (for $\mathsf{LRW2}$ the tweak and message are preprocessed as $(t, m) \mapsto m \oplus h(t)$ for some universal hash function $h \in H$). We will not rely on the potential cryptographic strength of the preprocessing functions: we only make a security assumption on E and assume the mixing functions are efficiently computable.

Formally, security is defined as the information-theoretic indistinguishability of $(\widetilde{E}, E)$ from $(\widetilde{\pi}, E)$, with $\widetilde{\pi}$ an ideal tweakable cipher, E an ideal cipher, and where the distinguisher has forward and inverse query access to both of its oracles. We remark that the same security model is, for instance, oft-employed in the area of key-length extenders [1, 4, 12, 16, 17, 28].

Generic Design. We start with presenting a generic description of a tweakable blockcipher design $\widetilde{E}[\rho]$ for $\rho \geq 1$. It consists of ρ calls to a classical blockcipher E interlaced with arbitrary mixing functions to generate the inputs to primitive

[3] We exclude schemes that use a blockcipher E with a larger key space, such as the tweakable blockcipher $\widetilde{E}(k, t, m) = E(k \| t, m)$ for a blockcipher E with $2n$-bit key.

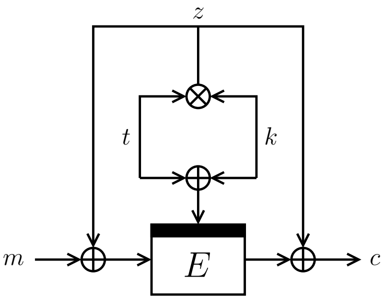

Fig. 1: Tweakable blockcipher $\widetilde{F}[1]$

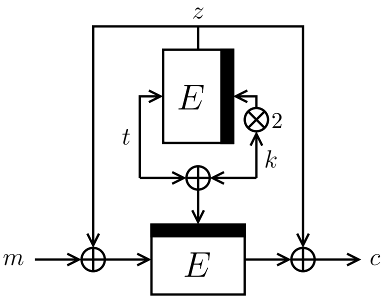

Fig. 2: Tweakable blockcipher $\widetilde{F}[2]$

calls and to generate the final output of the tweakable cipher. To assure invertibility of $\widetilde{E}[\rho]$, we pose a validity condition on the mixing functions, and only consider mixing functions that comply with this condition. Next, we consider various instances of $\widetilde{E}[\rho]$.

One Blockcipher Call with Linear Mixing. We first focus on the case $\rho = 1$, with the mixing functions being linear mappings over the finite field $\mathrm{GF}(2^n)$, and formally prove that any tweakable blockcipher of this form can be broken in a total complexity of about $2^{n/2}$. The attack covers for instance the tweakable cipher $E(k \oplus t, m)$ discussed before.

One Blockcipher Call with Polynomial Mixing. Next, we allow for mixing functions that involve multiplications, and introduce the tweakable blockcipher $\widetilde{F}[1] : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ (see also Fig. 1):

$$\widetilde{F}[1](k,t,m) = E(k \oplus t, m \oplus z) \oplus z, \text{ where } z = k \otimes t\,.$$

We prove that $\widetilde{F}[1]$ is indistinguishable from an ideal tweakable cipher as long as the distinguisher's complexity is at most $2^{2n/3}$. The proof is based on Patarin's H-coefficient technique [38] which has found recent adoption in, among others, generic blockcipher design [9, 12] and MAC security [36]. It additionally uses the finite field equivalent of Szemerédi-Trotter theorem [47], a result that was also used by Jetchev et al. [24] in the context of blockcipher based hashing. Informally, this theorem states that if L is a set of lines in a finite field and P a set of two-dimensional points, the number of point-line incidences is at most $\min\{|L|^{1/2}|P| + |L|, |L||P|^{1/2} + |P|\}$. This theorem is applied by viewing construction queries as lines and primitive queries as points.

Two Blockcipher Calls with Linear Mixing. Thirdly, we consider the case $\rho = 2$ and linear mixing functions, and introduce $\widetilde{F}[2] : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ (see also Fig. 2):

$$\widetilde{F}[2](k,t,m) = E(k \oplus t, m \oplus z) \oplus z, \text{ where } z = E(2k, t)\,.$$

It differs from $\widetilde{F}[1]$ in that the tweak processing $z = k \otimes t$ is replaced by $E(2k, t)$. We remark that PCLMULQDQ and AES are comparably efficient on the latest Intel Haswell processors [19]. Using slightly more involved techniques than for $\widetilde{F}[1]$, we prove that $\widetilde{F}[2]$ is an optimally secure tweakable cipher up to about 2^n queries.

Update After Observation by Guo et al. Guo et al. pointed out an oversight in the analysis of $\widetilde{F}[2]$. In detail, the original scheme performed the masking with $z = E(k, t)$, and the case where this subkey generation coincides with the encryption itself was simply not covered by bad transcripts, and could be exploited in inverse queries. Effectively, the original analysis of $\widetilde{F}[2]$ only covered the case $t \neq 0$. In this updated work, we have resolved this by a constant multiplication of the key k, and restored the security claim with an insignificant loss of $q/2^n$. The new $\widetilde{F}[2]$ offers optimal security for all tweaks $t \in \{0,1\}^n$.

Comparison

A comparison of $\widetilde{F}[1]$ and $\widetilde{F}[2]$ with the state of the art is given in Table 1. It shows that $\widetilde{F}[1]$ and $\widetilde{F}[2]$ compare favorably. For instance, both $\widetilde{F}[1]$ and $\mathsf{LRW2}[2]$ achieve $2n/3$-bit security, but the latter uses 2 blockcipher calls and 2 universal hash function calls. This means that $\widetilde{F}[1]$ is more efficient if one key renewal is cheaper than one blockcipher evaluation plus one universal hash evaluation. It additionally uses a key that is four times as small. Similarly, $\widetilde{F}[2]$ achieves optimal security using 2 cipher calls and 1 key renewal. The same bound is asymptotically achieved by $\mathsf{LRW2}[\rho]$ for $\rho \to \infty$, but this one requires ρ cipher calls and ρ universal hash calls, and has a key of size $2\rho n$.

On the other hand, $\widetilde{F}[1]$ and $\widetilde{F}[2]$ are proven in the information-theoretic model while the other schemes are analyzed in the complexity-theoretic model. Both schemes require a blockcipher that offers resistance against distinguishers that may freely choose the tweak that transforms the key input k under XOR. Fortunately, no related-key attacks of this form on the widely used blockciphers such as AES are known: Biryukov et al. [5, 6] derived a related-key attack on full AES-192 and AES-256, but using a more complicated and contrived key relation (see also Daemen and Rijmen [11]). We note that the proofs for $\widetilde{F}[1]$ and $\widetilde{F}[2]$ can straightforwardly be transformed to the complexity-theoretic model as long as the underlying blockcipher is related-key secure under XOR in the formalization of Bellare and Kohno [3]. This requires a hybrid proof, where the first step consists of replacing the underlying blockcipher E by an ideal primitive (at the cost of the related-key security of E). This step is, however, relatively loose, which can be seen from the fact that the ideal cipher achieves tight $2^{n/2}$ related-key security under XOR while it yields $2^{2n/3}$ and 2^n security for $\widetilde{F}[1]$ and $\widetilde{F}[2]$ in the information-theoretic model.

Table 1: Comparison of $\widetilde{F}[1]$ and $\widetilde{F}[2]$ with existing tweakable blockciphers. Universal hashes in $\mathsf{LRW2}[\rho]$ are instantiated as multiplications in the finite field of 2^n elements (see also Sect. 2). Cost is divided into plain E-calls, multiplications or universal hashes $\otimes/h$, and the number of E-calls with tweak-dependent key "tdk". For Min, $|t|$ denotes the fixed size of the tweak. The security bounds on $\widetilde{F}[1]$ and $\widetilde{F}[2]$ are derived in the information-theoretic model.

scheme	security	key	cost			reference
	($\log_2$)	length	E	$\otimes/h$	tdk	
LRW1	$n/2$	n	2	0	0	[29]
LRW2	$n/2$	$2n$	1	1	0	[29]
XEX	$n/2$	n	2	0	0	[41]
LRW2[2]	$2n/3$	$4n$	2	2	0	[27]
LRW2[ρ]	$\rho n/(\rho+2)$	$2\rho n$	ρ	ρ	0	[26]
Min	$\max\{n/2, n-\lvert t\rvert\}$	n	2	0	1	[33]
$\widetilde{F}[1]$	$2n/3$	n	1	1	1	Sect. 4.2
$\widetilde{F}[2]$	n	n	2	0	1	Sect. 5

Outline

We present the security model in Sect. 2. Our generic tweakable blockcipher design $\widetilde{E}[\rho]$ is given in Sect. 3. In Sect. 4, we consider $\rho = 1$: the impossibility result for linear mixing is given in Sect. 4.1 and our construction $\widetilde{F}[1]$ using polynomial mixing is introduced in Sect. 4.2. Then, in Sect. 5, we consider $\rho = 2$ and present $\widetilde{F}[2]$ based on linear mixing functions. The work is concluded in Sect. 6.

2 Model

By $\{0,1\}^n$ we denote the set of bit strings of length n. Let $\mathrm{GF}(2^n)$ be the field of order 2^n. We identify bit strings from $\{0,1\}^n$ and finite field elements in $\mathrm{GF}(2^n)$. This is done by representing a string $a = a_{n-1}a_{n-2}\cdots a_1a_0 \in \{0,1\}^n$ as polynomial $a(\mathtt{x}) = a_{n-1}\mathtt{x}^{n-1} + a_{n-2}\mathtt{x}^{n-2} + \cdots + a_1\mathtt{x} + a_0 \in \mathrm{GF}(2^n)$ and vice versa. There is additionally a one-to-one correspondence between $[0, 2^n - 1]$ and $\{0,1\}^n$, by considering $a(2) \in [0, 2^n - 1]$. For $a, b \in \{0,1\}^n$, we define addition $a \oplus b$ as addition of the polynomials $a(\mathtt{x}) + b(\mathtt{x}) \in \mathrm{GF}(2^n)$. Multiplication $a \otimes b$ is defined with respect to the irreducible polynomial $f(\mathtt{x})$ used to represent $\mathrm{GF}(2^n)$: $a(\mathtt{x}) \cdot b(\mathtt{x}) \bmod f(\mathtt{x})$.

If $\mathcal{A}$ is some set, $a \xleftarrow{\$} \mathcal{A}$ denotes the uniformly random drawing of a from $\mathcal{A}$. The size of $\mathcal{A}$ is denoted by $|\mathcal{A}|$.

Distinguishers and Advantages

Throughout this work, a distinguisher $\mathcal{D}$ is a computationally unbounded probabilistic algorithm. It is given query access to one or more oracles $\mathcal{O}$, which means that it can make a certain amount of queries to $\mathcal{O}$ adaptively. After this communication with $\mathcal{O}$, the distinguisher outputs a 0 or a 1. For two different oracles $\mathcal{O}$ and $\mathcal{P}$, we define the advantage of $\mathcal{D}$ in distinguishing both worlds by

$$\mathbf{Adv}(\mathcal{D}) = \left|\mathbf{Pr}\left[\mathcal{D}^{\mathcal{O}} = 1\right] - \mathbf{Pr}\left[\mathcal{D}^{\mathcal{P}} = 1\right]\right| . \tag{6}$$

We use the H-coefficient technique by Patarin [38] and Chen and Steinberger [9]. Consider a fixed deterministic distinguisher trying to distinguish two oracles $\mathcal{O}$ and $\mathcal{P}$, where its advantage function is denoted $\mathbf{Adv}(\mathcal{D})$ as in (6). Denote by X (resp. Y) the probability distribution of views when interacting with $\mathcal{O}$ (resp. $\mathcal{P}$). Let v be a view, i.e., a list of query-response tuples $\mathcal{D}$ may observe while interacting with $\mathcal{O}$ or $\mathcal{P}$. This view is called "attainable" if an interaction with $\mathcal{P}$ could render this view, or formally if $\mathbf{Pr}[Y = v] > 0$. We denote by $\mathcal{V}$ the set of attainable views.

Lemma 1 (Patarin's Technique). *Let $\mathcal{D}$ be a deterministic distinguisher. Consider a partition $\mathcal{V} = \mathcal{V}_{\text{good}} \cup \mathcal{V}_{\text{bad}}$ of the set of attainable views. Let $0 \leq \varepsilon \leq 1$ be such that for all $v \in \mathcal{V}_{\text{good}}$,*

$$\frac{\mathbf{Pr}[X = v]}{\mathbf{Pr}[Y = v]} \geq 1 - \varepsilon\,. \tag{7}$$

Then, the distinguishing advantage satisfies $\mathbf{Adv}(\mathcal{D}) \leq \varepsilon + \mathbf{Pr}[Y \in \mathcal{V}_{\text{bad}}]$.

A proof of this lemma is given in [9]. The idea of the technique is that only few views are significantly more likely to appear in $\mathcal{P}$ than in $\mathcal{O}$. In other words, the ratio (7) is close to 1 for all but a few views: the "bad" views. The definition of "bad" views is sometimes a delicate process, rendering a tradeoff between ε and $\mathbf{Pr}[Y \in \mathcal{V}_{\text{bad}}]$. Indeed, a too loose definition of bad views results in a larger second term, while a too tight one renders a larger ε.

Blockciphers and Tweakable Blockciphers

A blockcipher $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is a mapping such that for every key $k \in \mathcal{K}$, $E_k(\cdot) = E(k, \cdot)$ is a permutation on $\mathcal{M}$. We denote its inverse for fixed k by $E_k^{-1}(\cdot)$. We denote by $\mathsf{BC}(\mathcal{K}, \mathcal{M})$ the set of all such blockciphers.

A tweakable blockcipher $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ is a mapping such that for every $k \in \mathcal{K}$ and every tweak $t \in \mathcal{T}$, the function $\widetilde{E}_k(t, \cdot) = \widetilde{E}(k, t, \cdot)$ is a permutation on $\mathcal{M}$. Like before, its inverse is denoted by $\widetilde{E}_k^{-1}(\cdot, \cdot)$. Let $\widetilde{\mathsf{P}}(\mathcal{T}, \mathcal{M})$ be the set of all functions $\widetilde{\pi} : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that for all $t \in \mathcal{T}$, $\widetilde{\pi}(t, \cdot)$ is a permutation on $\mathcal{M}$.

Security of tweakable blockciphers considers a distinguisher $\mathcal{D}$ that has query access to a tweakable blockcipher $\widetilde{E}_k$ for $k \xleftarrow{\$} \mathcal{K}$ or an ideal tweakable permutation $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{P}}(\mathcal{T}, \mathcal{M})$, and tries to distinguish both worlds. It is typically bounded to have limited resources, such as q queries and τ time. In this work, we focus on modular designs for tweakable blockciphers, where $\widetilde{E}$ uses a blockcipher E as underlying primitive. If we denote by τ_E the time needed for one evaluation of E, the distinguisher can evaluate this underlying cipher at most $r := \tau/\tau_E$ times. We consider E to be perfectly secure and give $\mathcal{D}$ query access to E. More formally, we define the strong tweakable-PRP security of $\widetilde{E}$ based on E as

$$\mathbf{Adv}^{\widetilde{\text{sprp}}}_{\widetilde{E}}(\mathcal{D}) = \left|\mathbf{Pr}\left[\mathcal{D}^{\widetilde{E}_k^{\pm},E^{\pm}} = 1\right] - \mathbf{Pr}\left[\mathcal{D}^{\widetilde{\pi}^{\pm},E^{\pm}} = 1\right]\right| ,$$

where the probabilities are taken over the random choices of $k \xleftarrow{\$} \mathcal{K}$, $E \xleftarrow{\$} \mathsf{BC}(\mathcal{K}, \mathcal{M})$, and $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{P}}(\mathcal{T}, \mathcal{M})$, and the random coins of $\mathcal{D}$. Distinguisher $\mathcal{D}$ is bounded to make q queries to its first (construction) oracle and r queries to its second (primitive) oracle.

Universal Hash Functions

A hash function family $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is called *ε-almost 2-XOR-universal* if for all distinct $x, x' \in \mathcal{X}$ and $y \in \mathcal{Y}$, $\mathbf{Pr}\left[h \xleftarrow{\$} \mathcal{K} \ : \ H_h(x) \oplus H_h(x') = y\right] \leq \varepsilon$ [25,40]. A well-known universal hash function $H : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ with $\varepsilon = 2^{-n}$ is defined by multiplication in GF(2^n): $H_h(x) = h \otimes x$.

3 Generic Design

Here and throughout we consider $\mathcal{K} = \mathcal{T} = \mathcal{M} = \{0,1\}^n$ for some $n \geq 1$. Let $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. A generic tweakable blockcipher $\widetilde{E}[\rho] : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ based on $\rho \geq 1$ calls to E can be represented by mappings $A_i : \{0,1\}^{(i+2)n} \to \{0,1\}^n$ for $i = 1, \ldots, \rho+1$ and $B_i : \{0,1\}^{(i+1)n} \to \{0,1\}^n$ for $i = 1, \ldots, \rho$ as follows:

```
procedure $\widetilde{E}[\rho](k, t, m)$
    for $i = 1, \ldots, \rho$ do
        $x_i = A_i(k, t, y_1, \ldots, y_{i-1}, m)$
        $l_i = B_i(k, t, y_1, \ldots, y_{i-1})$
        $y_i = E(l_i, x_i)$
    return $c = A_{\rho+1}(k, t, y_1, \ldots, y_\rho, m)$
```

The tweakable blockcipher $\widetilde{E}[3]$ making $\rho = 3$ blockcipher calls is depicted in Fig. 3. The design resembles ideas of the permutation based hash function construction described by Rogaway and Steinberger [43] and the blockcipher based hash function construction described by Mennink [31]. However, $\widetilde{E}[\rho]$ is required to be invertible. In other words, on input of k, t, c, $\widetilde{E}[\rho]^{-1}(k, t, c) = m$ should be computable, and we will pose a validity condition on A_i, B_i to guarantee this.

Definition 1 (informal). *The mixing functions A_i for $i = 1, \ldots, \rho+1$ and B_i for $i = 1, \ldots, \rho$ are* valid *if there is exactly one function A_{i^*} that processes m, such that the first $i^* - 1$ rounds of $\widetilde{E}[\rho]$ can be computed in forward direction without knowledge of m, the last $\rho - (i^* - 1)$ rounds in inverse direction without knowledge of m, and A_{i^*} can be inverted to obtain m.*

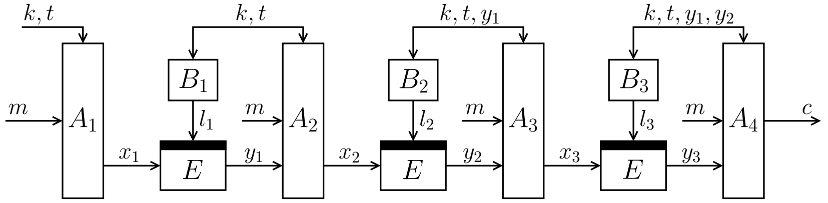

Fig. 3: Tweakable blockcipher $\widetilde{E}[3]$ making three blockcipher evaluations

Note that we already require that $B_1, \ldots, B_\rho$ do not get m as input. A formal definition of valid mixing functions is given in App. A; this definition is more technical and not strictly needed for a better understanding of the attacks and proofs in this work.

Apart from the validity condition, the mixing functions could be anything, and may technically even be of the form $A_1(k, t, m) = \mathsf{AES}(k \oplus t, m)$. However, it is reasonable to assume the mixing functions to be sufficiently efficient, and we focus on constructions with polynomial mixing functions.

4 One Blockcipher Call

In Sect. 4.1 we consider $\widetilde{E}[1]$ for any triplet of valid functions A_1, B_1, A_2 that are linear mappings over $\mathrm{GF}(2^n)$, hence only consist of addition and scalar multiplication. We show that any such tweakable cipher can be attacked by an information-theoretic distinguisher in at most $2^{n/2}$ queries, and thus that provable security beyond this bound cannot be achieved. In Sect. 4.2 we allow for mixing functions that consist of a finite field multiplication, and introduce $\widetilde{F}[1]$.

4.1 Linear Mixing

We present an attack on $\widetilde{E}[1]$ for any A_1, B_1, A_2 that comply with the invertibility condition and that are linear.

Proposition 1. *Let $n \geq 1$. Let $\widetilde{E}[1] : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable blockcipher based on valid linear A_1, B_1, A_2. Then, there is a distinguisher $\mathcal{D}$ making at most $2^{n/2+1}$ construction queries and $2^{n/2+1}$ primitive queries, such that*

$$\mathbf{Adv}^{\widetilde{\mathrm{sprp}}}_{\widetilde{E}[1]}(\mathcal{D}) \geq 1 - \frac{1}{2^n}\,.$$

Proof. The mixing functions are linear, and can be represented by matrices

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ b_{11} & b_{12} & 0 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} a_{21} & a_{22} & a_{23} & a_{24} \end{pmatrix},$$

where A_1, B_1 are evaluated on (k, t, m) and A_2 on (k, t, y_1, m). Additional conditions apply regarding the validity. Note that we have to distinguish two cases: $i^* = 1$ and $i^* = 2$, and we start with the latter.

Case $\boldsymbol{i^* = 2}$. Validity requires that A_1 is independent of m (hence $a_{13} = 0$) and A_2 is an invertible mapping $m \mapsto c$ for any k, t, y_1 (hence $a_{24} \neq 0$). Distinguisher $\mathcal{D}$ selects an arbitrary t and two arbitrary distinct m, m'. Then, it queries $c \leftarrow \mathcal{O}(t, m)$ and $c' \leftarrow \mathcal{O}(t, m')$, where $\mathcal{O}$ is either $\widetilde{E}[1]$ or $\widetilde{\pi}$. If $c \oplus c' = a_{24}(m \oplus m')$, the distinguisher outputs 1, otherwise it outputs 0. Note that the distinguisher always outputs 1 if it is in the real world: because $a_{13} = 0$, both queries have identical E-calls, and thus $y_1 = y_1'$. Therefore, $c \oplus c' = A_2\,(0, 0, 0, m \oplus m') = a_{24}(m \oplus m')$. On the other hand, this condition is set in the ideal world with probability $1/2^n$. This gives a distinguisher in 2 construction queries with a success probability of $1 - 1/2^n$.

Case $\boldsymbol{i^* = 1}$. This case is more technical. Validity requires that A_1 is an invertible mapping $m \mapsto x_1$ for any k, t (hence $a_{13} \neq 0$). A_2 is required to be independent of m (hence $a_{24} = 0$) and an invertible mapping $y_1 \mapsto c$ for any k, t (hence $a_{23} \neq 0$). At a high level, we consider a distinguisher $\mathcal{D}$ that queries its construction oracle $\mathcal{O}$ (either $\widetilde{E}[1]$ or $\widetilde{\pi}$) and/or its primitive oracle E, with the goal to find a colliding pair: a construction query (t_i, m_i, c_i) and a primitive evaluation (l_j, x_j, y_j) such that

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} \begin{pmatrix} k \\ t_i \\ m_i \end{pmatrix} = \begin{pmatrix} x_j \\ l_j \end{pmatrix}. \tag{8}$$

In this case, the attacker can verify if $A_2(k, t_i, y_j, m_i) \overset{?}{=} c_i$, and output 0/1 accordingly. Technicalities arise as the key k is unknown and it is not straightforward to find a pair of queries satisfying (8). Additionally, for some A_1, B_1 a different technique has to be employed. We make a further distinction among four cases. The case distinction is made based on the values b_{12}, b_{11}, and a_{11}.

Subcase $\boldsymbol{b_{12} = 0}$. In this case the key input to the blockcipher is $b_{11}k$. The distinguisher selects arbitrary distinct t, t' and an arbitrary m, and sets $m' = m \oplus a_{13}^{-1} a_{12}(t \oplus t')$. Then, it queries $c \leftarrow \mathcal{O}(t, m)$ and $c' \leftarrow \mathcal{O}(t', m')$. If $c \oplus c' = a_{22}(t \oplus t')$, the distinguisher outputs 1, otherwise it outputs 0. The remaining analysis is similar to previous case $i^* = 2$, using that $a_{24} = 0$ and $y_1 = y_1'$ in the real world. This gives a distinguisher in 2 construction queries with a success probability of $1 - 1/2^n$.

Subcase $\boldsymbol{b_{12} \neq 0}$, $\boldsymbol{b_{11} = a_{11} = 0}$. In this case k is not used as input to A_1 and B_1. The distinguisher selects an arbitrary t and arbitrary distinct m, m'. Then, it queries $c \leftarrow \mathcal{O}(t, m)$ and $c' \leftarrow \mathcal{O}(t, m')$. Additionally, it queries $y \leftarrow E(B_1(k, t, m), A_1(k, t, m))$ and $y' \leftarrow E(B_1(k, t, m'), A_1(k, t, m'))$ (which can be queried without knowledge of k as $a_{11} = b_{11} = 0$). If $c \oplus c' = a_{23}(y \oplus y')$, the distinguisher outputs 1, otherwise it outputs 0. The remaining analysis is similar to before. This gives a distinguisher in 2 construction queries and 2 primitive queries with a success probability of $1 - 1/2^n$.

Subcase $\boldsymbol{b_{12} \neq 0}$, $\boldsymbol{b_{11} \neq 0}$. This is the most general subcase. (8) is equivalent to finding a construction query (t_i, m_i, c_i) and a primitive evaluation (l_j, x_j, y_j) such that

$$\begin{pmatrix} 0 & a'_{12} & a_{13} \\ b_{11} & b_{12} & 0 \end{pmatrix} \begin{pmatrix} k \\ t_i \\ m_i \end{pmatrix} = \begin{pmatrix} x_j \oplus b_{11}^{-1} a_{11} l_j \\ l_j \end{pmatrix}, \tag{9}$$

where $a'_{12} = a_{12} \oplus b_{11}^{-1} a_{11} b_{12}$ and where $b_{11}, b_{12}, a_{13} \neq 0$. The distinguisher defines

$$\begin{aligned} &\text{for } i = 1, \dots, 2^{n/2}: \quad && t_i = b_{12}^{-1}\left(\langle i-1\rangle_{n/2} \| 0^{n/2}\right) \quad && \text{and} \quad m_i = a_{13}^{-1} a'_{12} t_i\,, \\ &\text{for } j = 1, \dots, 2^{n/2}: \quad && l_j = 0^{n/2} \| \langle j-1\rangle_{n/2} \quad && \text{and} \quad x_j = b_{11}^{-1} a_{11} l_j\,. \end{aligned}$$

Note that these values are selected such that the first equation of (9) holds for any (i, j): it reads $a'_{12} t_i \oplus a'_{12} t_i = 0$. Regarding the second equation, we have $b_{12}\{t_1, \dots, t_{2^{n/2}}\} \oplus \{l_1, \dots, l_{2^{n/2}}\} = \{0,1\}^n$, hence this equation will hold for exactly one $(i^\star, j^\star)$.

For $i = 1, \dots, 2^{n/2}$, it queries $c_i \leftarrow \mathcal{O}(t_i, m_i)$. For $j = 1, \dots, 2^{n/2}$, it queries $y_j \leftarrow E(l_j, x_j)$. For every i, j, the distinguisher writes $k_{ij} = b_{11}^{-1}(l_j \oplus b_{12} t_i)$ and verifies if $A_2(k_{ij}, t_i, y_j, m_i) \overset{?}{=} c_i$. For any i, j such that this equation holds, the adversary chooses an arbitrary *new* tweak t'_i and arbitrary message m'_i, sets $x'_j = A_1(k_{ij}, t'_i, m'_i)$ and $l'_j = B_1(k_{ij}, t'_i, m'_i)$, makes construction query $c'_i \leftarrow \mathcal{O}(t'_i, m'_i)$ and primitive query $y'_j \leftarrow E(l'_j, x'_j)$, and verifies if $A_2(k_{ij}, t'_i, y'_j, m'_i) \overset{?}{=} c'_i$.

If there is an i, j such that both verifications succeed, the distinguisher outputs 1, otherwise it outputs 0. Recall that in the real world there is exactly one solution $k = k_{i^\star j^\star}$ and both verifications succeed for this key. In the ideal world, the distinguisher outputs 1 if there is a combination of i, j such that both verifications succeed. This happens with probability at most $2^{n/2} \cdot 2^{n/2} \cdot (1/2^n)^2 = 1/2^n$. This gives a distinguisher that makes at most $2^{n/2+1}$ construction queries and $2^{n/2+1}$ primitive queries and succeeds with probability $1 - 1/2^n$.

Subcase $\boldsymbol{b_{12} \neq 0}$, $\boldsymbol{b_{11} = 0}$, $\boldsymbol{a_{11} \neq 0}$. This case is in fact the orthogonal of the previous one. Now, (8) translates to finding a construction query (t_i, m_i, c_i) and a primitive evaluation (l_j, x_j, y_j) such that

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & b_{12} & 0 \end{pmatrix} \begin{pmatrix} k \\ t_i \\ m_i \end{pmatrix} = \begin{pmatrix} x_j \\ l_j \end{pmatrix}, \tag{10}$$

where $a_{11}, b_{12}, a_{13} \neq 0$. The distinguisher defines

$$\begin{aligned} &\text{for } i = 1, \dots, 2^{n/2}: \quad && t_i = 0^n \quad && \text{and} \quad m_i = a_{13}^{-1}\left(\langle i-1\rangle_{n/2} \| 0^{n/2}\right), \\ &\text{for } j = 1, \dots, 2^{n/2}: \quad && l_j = 0^n \quad && \text{and} \quad x_j = 0^{n/2} \| \langle j-1\rangle_{n/2}\,. \end{aligned}$$

Note that the second equation of (10) holds for any (i, j), but there is exactly one combination for which the first equation holds. The remainder of the attack literally follows previous case. □

The authenticated encryption scheme $\mathsf{McOE\text{-}X}$ by Fleischmann et al. [15] uses the tweakable blockcipher $\widetilde{E}_{\mathsf{McOE\text{-}X}}(k,t,m) = E(k \oplus t, m)$, and Prop. 1 gives a distinguishing attack in about $2^{n/2}$ queries. In fact, the attack of Mendel et al. [30] on $\mathsf{McOE\text{-}X}$ uses a generalization of the attack of Prop. 1.

4.2 Polynomial Mixing

We consider the design of a tweakable blockcipher based on one blockcipher call where the mixing functions may consist of a finite field multiplication. Recall the $\mathsf{LRW2}$ tweakable blockcipher of (2) that is based on a 2-XOR-universal hash function h. We make two simplifications: firstly, we instantiate it with the optimally secure 2-XOR-universal hash function $h(x) = h \otimes x$ (see Sect. 2), and secondly, we put $h = k$. This results in the following function $\mathsf{LRW2}'$: $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$:

$$\mathsf{LRW2}'(k,t,m) = E(k, m \oplus z) \oplus z, \text{ where } z = k \otimes t\,.$$

This function achieves security up to at most $2^{n/2}$ queries [29]. However, it turns out that a significant security gain can be made by making the key input tweak-dependent.

In more detail, we propose the following tweakable cipher $\widetilde{F}[1]$: $\{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$:

$$\widetilde{F}[1](k,t,m) = E(k \oplus t, m \oplus z) \oplus z, \text{ where } z = k \otimes t\,.$$

The function is depicted in Fig. 1. In the following theorem, we prove that it achieves $2n/3$-bit security.

Theorem 1. *Let $n \geq 1$. Let $\mathcal{D}$ be a distinguisher making at most q construction queries and r primitive queries. Then,*

$$\mathbf{Adv}^{\widetilde{\text{sprp}}}_{\widetilde{F}[1]}(\mathcal{D}) \leq \frac{2\min\{q^{1/2}r + q, qr^{1/2} + r\}}{2^n}\,.$$

Equilibrium is achieved for $q = r$, for which $\widetilde{F}[1]$ achieves approximately $2^{2n/3}$ security. Note that the result implies something even stronger: if the online complexity q is at most $2^{n/2}$, the offline complexity r can be up to almost 2^{n-3}. The proof relies on the finite field equivalent of Szemerédi-Trotter theorem [47], which – to our knowledge – was first introduced to cryptography by Jetchev et al. [24].

Lemma 2 (Szemerédi-Trotter Theorem Over Finite Fields). *Let $\mathbb{F}$ be a finite field. Let P (resp. L) be a set of points (resp. lines) in $\mathbb{F}^2$. Define $I(P,L) = \{(p,\ell) \in P \times L \mid p \in \ell\}$. Then,*

$$|I(P,L)| \leq \min\{|L|^{1/2}|P| + |L|, |L||P|^{1/2} + |P|\}\,.$$

A proof of this lemma can be found in Tao [48] and Özen [37, Thm. 5.1.5]. (Tao [48] shows that the bound is more or less sharp: put P the set of all points in $\mathbb{F}^2$ and L the set of all lines in $\mathbb{F}^2$. Then, both $|P|$ and $|L|$ are approximately $|\mathbb{F}|^2$ and the number of point-line incidences $I(P, L)$ is about $|\mathbb{F}|^3$.) Using Lem. 2, we are ready to prove Thm. 1.

Proof (Proof of Theorem 1). Let $k \xleftarrow{\$} \{0,1\}^n$, $E \xleftarrow{\$} \mathsf{BC}(\{0,1\}^n, \{0,1\}^n)$, and $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{P}}(\{0,1\}^n, \{0,1\}^n)$. We consider a computationally unbounded distinguisher $\mathcal{D}$ that has bidirectional access to two oracles: $(\widetilde{F}[1]_k, E)$ in the real world and $(\widetilde{\pi}, E)$ in the ideal world. As $\mathcal{D}$ is computationally unbounded, we can without loss of generality assume that it is deterministic and we apply Lem. 1. The distinguisher makes q queries to $\mathcal{O}_1 \in \{\widetilde{F}[1]_k, \widetilde{\pi}\}$, and these are summarized in a view $v_1 = \{(t_1, m_1, c_1), \ldots, (t_q, m_q, c_q)\}$. Similarly, it makes r queries to $\mathcal{O}_2 = E$, which are summarized in a view $v_2 = \{(l_1, x_1, y_1), \ldots, (l_r, x_r, y_r)\}$. Without loss of generality, we assume that both v_1 and v_2 do not contain duplicate elements. Additionally, we assume that both views are attainable. For v_1, this is the case if and only if for any distinct i, i' such that $t_i = t_{i'}$, we have $m_i \neq m_{i'}$ and $c_i \neq c_{i'}$. The case of v_2 is equivalent.

After $\mathcal{D}$'s interaction with $(\mathcal{O}_1, \mathcal{O}_2)$, but *before* it outputs its decision 0/1, we disclose the key k to the distinguisher. In real world, this is the key used for the game, in the ideal world k will be a fake and freshly drawn key. This is truly without loss of generality, as it only leads to an increase in the distinguishing advantage (the distinguisher can ignore this information, if it wants). The complete view is denoted $v = (v_1, v_2, k)$.

Bad Views. We next present our definition of bad views, followed by an informal explanation. We define by $\mathcal{V}_{\text{bad}}$ the set of all views v such that at least one of the following two conditions holds:

$$\exists\, (t, m, c) \in v_1, (l, x, y) \in v_2 : \ (k \oplus t, m \oplus k \otimes t) = (l, x)\,, \tag{11a}$$

$$\exists\, (t, m, c) \in v_1, (l, x, y) \in v_2 : \ (k \oplus t, c \oplus k \otimes t) = (l, y)\,. \tag{11b}$$

Recall the partition $\mathcal{V} = \mathcal{V}_{\text{good}} \cup \mathcal{V}_{\text{bad}}$, implying that any attainable view such that (11) does not hold, is good.

We give a high-level explanation of the definition of bad views. Note that we can implicitly "map" all tuples in v_1 to their corresponding E-evaluation: a tuple $(t, m, c) \in v_1$ corresponds to E-evaluation $(k \oplus t, m \oplus k \otimes t, c \oplus k \otimes t)$, where k is given in v. Intuitively, we want that there are no two tuples in $v_1 \cup v_2$ whose E-evaluations "collide", in the sense that they render the same input to or output of E. Two different tuples from v_2 never collide, by attainability of v. Two different tuples from v_1 also never collide. Indeed, let $(t, m, c), (t', m', c') \in v_1$ be two different tuples. These collide if

$$(k \oplus t, m \oplus k \otimes t) = (k \oplus t', m' \oplus k \otimes t') \text{ or}$$
$$(k \oplus t, c \oplus k \otimes t) = (k \oplus t', c' \oplus k \otimes t')\,,$$

which is the case if and only if $(t,m)=(t',m')$ or $(t,c)=(t',c')$, impossible due to attainability of v. Finally, collisions between v_1 and v_2 imply (11).

$\mathbf{Pr}\left[\boldsymbol{Y}\in\boldsymbol{\mathcal{V}}_{\mathbf{bad}}\right]$. Consider the ideal world $(\widetilde{\pi},E)$. The key $k\xleftarrow{\$}\{0,1\}^n$ is a dummy key drawn independently of v_1,v_2. Starting with the first bad condition (11a), it is equivalent to

$$\exists\,(t,m,c)\in v_1,(l,x,y)\in v_2:\;(k\oplus t,m\oplus(l\oplus t)\otimes t)=(l,x)\,.$$

Note that the second equation is independent of k, it solely depends on the tuples $(t,m,c)\in v_1$ and $(l,x,y)\in v_2$, and we apply Lem. 2. For every $(t,m,c)\in v_1$ we ignore c and represent (t,m) as a line $\ell:\mathtt{y}=t\otimes\mathtt{x}\oplus(m\oplus t\otimes t)$ in $\mathrm{GF}(2^n)^2$. For every $(l,x,y)\in v_2$, we ignore y and consider (l,x) as a point $(\mathtt{x},\mathtt{y})$ in $\mathrm{GF}(2^n)^2$. The number of combinations $(t,m,c)\in v_1$ and $(l,x,y)\in v_2$ such that $m\oplus(l\oplus t)\otimes t=x$ is in fact the number of point-line incidences $I(v_2,v_1)$, which by Lem. 2 is at most $\min\{q^{1/2}r+q,qr^{1/2}+r\}=:f(q,r)$. Any of these tuples fixes one possible value $l\oplus t$. Therefore, there are at most $f(q,r)$ possible keys that could set (11a). A symmetric reasoning applies to (11b). As $k\xleftarrow{\$}\{0,1\}^n$, we find,

$$\mathbf{Pr}\left[Y\in\mathcal{V}_{\mathrm{bad}}\right]\leq\frac{2\min\{q^{1/2}r+q,qr^{1/2}+r\}}{2^n}\,.$$

$\mathbf{Pr}\left[\boldsymbol{X}=\boldsymbol{v}\right]/\mathbf{Pr}\left[\boldsymbol{Y}=\boldsymbol{v}\right]$. Let $v\in\mathcal{V}_{\mathrm{good}}$. For the computation of $\mathbf{Pr}\left[X=v\right]$ and $\mathbf{Pr}\left[Y=v\right]$, it suffices to compute the *fraction of oracles* that could result in view v, for both the real and ideal world. Formally, if we denote by all_X the set of all oracles in the real world, and by $\mathrm{comp}_X(v)$ the fraction of them compatible with v, we find $\mathbf{Pr}\left[X=v\right]=|\mathrm{comp}_X(v)|/|\mathrm{all}_X|$. Similarly for the ideal world.

Note that $|\mathrm{all}_X|=2^n\cdot(2^n!)^{2^n}$, the number of possible keys k times the number of possible ciphers E. Similarly, $|\mathrm{all}_Y|=2^n\cdot(2^n!)^{2^n}\cdot(2^n!)^{2^n}$, where the first term now corresponds to the disclosed dummy key. The computation of the number of oracles compatible with v is slightly more involved. We group the tuples in v_1 according to the tweak value and the tuples in v_2 according to the key value. More formally, for $t\in[0,2^n-1]$ define $\alpha_t=|\{(t',m',c')\in v_1\mid t'=t\}|$, and for $l\in[0,2^n-1]$ define $\beta_l=|\{(l',x',y')\in v_2\mid l'=l\}|$. Additionally, denote for $l\in[0,2^n-1]$:

$$\gamma_l=\alpha_{k\oplus l}+\beta_l\,.$$

This definition of γ_l is inspired by the fact that a tuple $(t,m,c)\in v_1$ corresponds to an E-evaluation with key input $l=k\oplus t$.

Using these definitions, we are ready to compute the number of compatible oracles. First consider $\mathrm{comp}_X(v)$. As v is a good view and does not satisfy (11), every query tuple in $v_1\cup v_2$ defines a unique E-evaluation. This leaves $\prod_{l=0}^{2^n-1}(2^n-\gamma_l)!$ blockciphers $E\in\mathsf{BC}(\{0,1\}^n,\{0,1\}^n)$ compliant with (v_1,v_2). Additionally, the key k is uniquely fixed as it is included in v. We find:

$$|\mathrm{comp}_X(v)|=\prod_{l=0}^{2^n-1}(2^n-\gamma_l)!\,.$$

Next, for the ideal world, a similar reasoning shows that there are $\prod_{t=0}^{2^n-1}(2^n-\alpha_t)!$ tweakable ciphers $\widetilde{\pi} \in \widetilde{\mathsf{P}}(\{0,1\}^n, \{0,1\}^n)$ compliant with v_1 and $\prod_{l=0}^{2^n-1}(2^n-\beta_l)!$ blockciphers $E \in \mathsf{BC}(\{0,1\}^n, \{0,1\}^n)$ compliant with v_2. We find:

$$\begin{aligned}|\mathrm{comp}_Y(v)| &= \prod_{t=0}^{2^n-1}(2^n-\alpha_t)! \cdot \prod_{l=0}^{2^n-1}(2^n-\beta_l)! \\ &= \prod_{l=0}^{2^n-1}(2^n-\alpha_{k\oplus l})! \cdot (2^n-\beta_l)! \leq (2^n)!^{2^n} \cdot \prod_{l=0}^{2^n-1}(2^n-\gamma_l)!\,,\end{aligned}$$

using that $(2^n-\alpha)! \cdot (2^n-\beta)! \leq (2^n-\alpha-\beta)! \cdot 2^n!$ for any $0 \leq \alpha, \beta \leq 2^n$. Assembling all bounds yields

$$\frac{\mathbf{Pr}\left[X=v\right]}{\mathbf{Pr}\left[Y=v\right]} = \frac{|\mathrm{all}_Y| \cdot |\mathrm{comp}_X(v)|}{|\mathrm{all}_X| \cdot |\mathrm{comp}_Y(v)|} \geq \frac{2^n \cdot (2^n!)^{2^n} \cdot (2^n!)^{2^n} \cdot \prod_{l=0}^{2^n-1}(2^n-\gamma_l)!}{2^n \cdot (2^n!)^{2^n} \cdot (2^n)!^{2^n} \cdot \prod_{l=0}^{2^n-1}(2^n-\gamma_l)!} = 1.$$

Lemma 1 thus carries over for $\varepsilon = 0$. □

5 Two Blockcipher Calls

We suggest an alternative to $\widetilde{F}[1]$ based on two blockcipher calls and linear mixing functions A_1, B_1, A_2, B_2, A_3. In more detail, we propose the following tweakable cipher $\widetilde{F}[2] : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$:

$$\widetilde{F}[2](k,t,m) = E(k \oplus t, m \oplus z) \oplus z, \text{ where } z = E(2k,t)\,.$$

The function is depicted in Fig. 2. $\widetilde{F}[2]$ differs from $\widetilde{F}[1]$ in that the tweak processing $z = k \otimes t$ is replaced by $E(2k,t)$. We remark that it is fair to make such transition, as multiplication and AES are comparably expensive on the latest Intel processors. In the following theorem, we prove that $\widetilde{F}[2]$ achieves optimal security.

Theorem 2. *Let $n \geq 1$. Let $\mathcal{D}$ be a distinguisher making at most q construction queries and r primitive queries. Then,*

$$\mathbf{Adv}_{\widetilde{F}[2]}^{\widetilde{\mathrm{sprp}}}(\mathcal{D}) \leq \frac{q+r}{2^n} + \frac{2qr}{(2^n-q)(2^n-q-r)}\,.$$

The bound guarantees security of $\widetilde{F}[2]$ up to almost 2^n queries to both the construction and the primitive. In more detail, the bound is at most 1/2 as long as $q, r \leq 2^{n-3}$.

We remark that the doubling of the key for the subkey generation, $z = E(2k,t)$, is new in the updated version of this work. If we restrict our scheme to $t \neq 0$, this doubling is not needed, and the resulting security bound is $\frac{r}{2^n} + \frac{2qr}{(2^n-q)(2^n-r)}$.

Proof. The proof is in the lines of the one of Thm. 1, but differences arise due to the evaluations of E involved in the transformation of $z = E(2k, t)$.

Let $k \xleftarrow{\$} \{0,1\}^n$, $E \xleftarrow{\$} \mathsf{BC}(\{0,1\}^n, \{0,1\}^n)$, and $\widetilde{\pi} \xleftarrow{\$} \widetilde{\mathsf{P}}(\{0,1\}^n, \{0,1\}^n)$. As before, we consider a computationally unbounded, deterministic, distinguisher $\mathcal{D}$ that has bidirectional access to $(\widetilde{F}[2]_k, E)$ in the real world and $(\widetilde{\pi}, E)$ in the ideal world. The distinguisher makes q queries to $\mathcal{O}_1 \in \{\widetilde{F}[2]_k, \widetilde{\pi}\}$, and these are summarized in a view $v_1 = \{(t_1, m_1, c_1), \ldots, (t_q, m_q, c_q)\}$. Similarly, it makes r queries to $\mathcal{O}_2 = E$, which are summarized in $v_2 = \{(l_1, x_1, y_1), \ldots, (l_r, x_r, y_r)\}$. Again, we assume that both v_1 and v_2 are attainable when interacting with the ideal world and do not contain duplicate elements.

After the $\mathcal{D}$'s interaction with $(\mathcal{O}_1, \mathcal{O}_2)$, but before it outputs its decision 0/1, we will again disclose the key k (fake k in the ideal world). We *additionally* disclose to the distinguisher all values $z_i = E(2k, t_i)$ for $i = 1, \ldots, q$. These will be disclosed in the form of a view $v_z = \{(2k, t_1, z_1), \ldots, (2k, t_{q'}, z_{q'})\}$, where q' denotes the number of *distinct* tweak values in v_1 (note that, indeed, the same tweak may appear in different tuples of v_1). Again, these disclosures are without loss of generality, as they only lead to an increase in the distinguishing advantage. The complete view is now denoted $v = (v_1, v_2, v_z, k)$.

Bad Views. We define by $\mathcal{V}_{\text{bad}}$ the set of all views v such that at least one of the following three conditions holds:

$$\exists\,(t, m, c) \in v_1 :\ 2k = k \oplus t\,, \tag{12a}$$

$$\exists\,(l, x, y) \in v_2 :\ 2k = l\,, \tag{12b}$$

$$\exists\,(t, m, c) \in v_1, (l, x, y) \in v_2, (2k, t, z) \in v_z :\ (k \oplus t, m \oplus z) = (l, x)\,, \tag{12c}$$

$$\exists\,(t, m, c) \in v_1, (l, x, y) \in v_2, (2k, t, z) \in v_z :\ (k \oplus t, c \oplus z) = (l, y)\,. \tag{12d}$$

Recall the partition $\mathcal{V} = \mathcal{V}_{\text{good}} \cup \mathcal{V}_{\text{bad}}$, implying that any attainable view such that (12) does not hold, is good. The bad conditions (12c-12d) match (11a-11b), with the difference that $z = E(2k, t)$ is involved. The bad conditions (12a-12b) are new and are used to rule out the event that any of the evaluations in v_z already "appears" in v_1 or v_2 (the condition is slightly stronger, assuring that v_1 and v_2 do not contain any query for key $2k$).[4]

$\mathbf{Pr}\,[\boldsymbol{Y} \in \boldsymbol{\mathcal{V}}_{\text{bad}}]$. Consider the ideal world $(\widetilde{\pi}, E)$. The key $k \xleftarrow{\$} \{0,1\}^n$ is a dummy key drawn independently of v_1, v_2. Basic probability theory:

$$\mathbf{Pr}\,[(12)] \leq \mathbf{Pr}\,[(12\text{a})] + \mathbf{Pr}\,[(12\text{b})] + \mathbf{Pr}\,[(12\text{c}) \vee (12\text{d}) \mid \neg(12\text{a}) \wedge \neg(12\text{b})]\ .$$

Condition (12a) holds with probability at most $q/2^n$, as there are at most q possible values t, and the key is randomly drawn from $\{0,1\}^n$. Similarly, condition (12b) holds with probability at most $r/2^n$, as there are at most r possible values

[4] Condition (12a) is new in the updated version of this paper. If we would leave out the doubling of the subkey in $\widetilde{F}[2]$, then the condition would read $k = k \oplus t$, which is satisfied with probability 0 if we impose $t \neq 0$.

l. Assume (12a) and (12b) are not set, hence both v_1 and v_2 do not contain any tuple $(2k, \cdot, \cdot)$. This particularly means that all values $z_1, \ldots, z_{q'}$ are drawn independently of v_1, v_2. Regarding condition (12c), we have q tuples in v_1 and r tuples v_2. Any combination fixes one possible $(l \oplus t, x \oplus m)$ and also fixes exactly one tuple in v_z. Therefore, there are at most qr possible drawings of (k, z) that could set (12c). A symmetric reasoning applies to (12d). As k is uniformly drawn from a set of size at least $2^n - q - r$ (condition ¬(12b)∧¬(12a) rules out at most $q + r$ values), and the corresponding z is drawn from a set of size at least $2^n - q$ (there are at most q values z, all different as E is a blockcipher), we find

$$\mathbf{Pr}\left[(12c) \vee (12d) \mid \neg(12b)\right] \leq \frac{2qr}{(2^n - q)(2^n - q - r)}\,.$$

Combining the bounds results in $\mathbf{Pr}\left[Y \in \mathcal{V}_{\text{bad}}\right] \leq \dfrac{q + r}{2^n} + \dfrac{2qr}{(2^n - q)(2^n - q - r)}$.

$\mathbf{Pr}\left[\boldsymbol{X} = \boldsymbol{v}\right] / \mathbf{Pr}\left[\boldsymbol{Y} = \boldsymbol{v}\right]$**.** The analysis of Thm. 1 carries over verbatim with the difference that we merge $v_2 \cup v_z$. Note that, by our definition of good views, these two sets do not overlap or conflict. □

The scheme $\widetilde{F}[2]$ is equally expensive as the tweakable blockcipher by Minematsu [33], which also makes two blockcipher calls, one with a tweak-dependent key. On the other hand, it achieves a significantly higher level of security: 2^n versus $2^{\max\{n/2, n-|t|\}}$, where $|t|$ denotes the size of the tweak.

6 Conclusions

We considered the generic design of n-bit tweakable blockciphers *only* based on calls to a classical blockcipher. $\widetilde{F}[1]$ and $\widetilde{F}[2]$ show that good beyond birthday bound security can be achieved quite elegantly. More detailed, the latter construction makes only two blockcipher calls and achieves optimal security.

As suggested in the original formalization of tweakable blockciphers by Liskov et al. [29], tweak renewal should be cheaper than key renewal. To a certain degree, this is a reasonable condition, but once generic constructions such as $\mathsf{LRW2}[\rho]$ require more and more primitive calls, it is of theoretical and practical interest to search for alternatives that release this side condition (see also Jean et al. [23]). In fact, $\widetilde{F}[1]$ and $\widetilde{F}[2]$ improve over the state of the art beyond birthday bound solutions, in the key size *and* in the efficiency as long as key renewal is reasonably cheap.

A direction for future research would be to investigate if improved bounds can be derived for $\widetilde{F}[1]$ or any other one-call scheme. Additionally, we note that our schemes are analyzed in the single-key model, and it may be of interest to investigate them under the related-key model where the adversary may influence the key input to the tweakable blockcipher. Finally, it is of interest to derive two-call schemes where the tweak transforms the key input to the underlying blockcipher in a more randomized way (in a similar fashion as Min of (5)).

Acknowledgments. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007). Bart Mennink is a Postdoctoral Fellow of the Research Foundation – Flanders (FWO). The author would like to thank Atul Luykx and the anonymous reviewers of FSE 2015 for their comments and suggestions. We would also like to thank Guo et al. for pointing out the missing term in the security analysis.

References

1. Aiello, W., Bellare, M., Di Crescenzo, G., Venkatesan, R.: Security amplification by composition: The case of doubly-iterated, ideal ciphers. In: CRYPTO '98. LNCS, vol. 1462, pp. 390–407. Springer (1998)
2. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Tischhauser, E., Yasuda, K.: Parallelizable and authenticated online ciphers. In: ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 424–443. Springer (2013)
3. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer (2003)
4. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer (2006)
5. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer (2009)
6. Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and related-key attack on the full AES-256. In: CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer (2009)
7. Chakraborty, D., Sarkar, P.: A general construction of tweakable block ciphers and different modes of operations. In: Inscrypt 2006. LNCS, vol. 4318, pp. 88–102. Springer (2006)
8. Chakraborty, D., Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-counter-hash approach. IEEE Transactions on Information Theory 54(4), 1683–1699 (2008)
9. Chen, S., Steinberger, J.: Tight security bounds for key-alternating ciphers. In: EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer (2014)
10. Crowley, P.: Mercy: A fast large block cipher for disk sector encryption. In: FSE 2000. LNCS, vol. 1978, pp. 49–63. Springer (2001)
11. Daemen, J., Rijmen, V.: On the related-key attacks against AES. Proceedings of the Romanian Academy, Series A 13(4), 395–400 (2012)
12. Dai, Y., Lee, J., Mennink, B., Steinberger, J.P.: Tight security bounds for multiple encryption. In: CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 20–38. Springer (2014)
13. Dworkin, M.: NIST SP 800-38E: Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices (2010)
14. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein Hash Function Family (2010), submission to NIST's SHA-3 competition
15. Fleischmann, E., Forler, C., Lucks, S.: McOE: A family of almost foolproof on-line authenticated encryption schemes. In: FSE 2012. LNCS, vol. 7549, pp. 196–215. Springer (2012)

16. Gaži, P.: Plain versus randomized cascading-based key-length extension for block ciphers. In: CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 551–570. Springer (2013)
17. Gaži, P., Maurer, U.: Cascade encryption revisited. In: ASIACRYPT 2009. LNCS, vol. 5912, pp. 37–51. Springer (2009)
18. Goldenberg, D., Hohenberger, S., Liskov, M., Crump Schwartz, E., Seyalioglu, H.: On tweaking Luby-Rackoff blockciphers. In: ASIACRYPT 2007. LNCS, vol. 4833, pp. 342–356. Springer (2007)
19. Gueron, S.: AES-GCM software performance on the current high end CPUs as a performance baseline for CAESAR competition. DIAC 2013 (2013)
20. Halevi, S.: EME*: Extending EME to handle arbitrary-length messages with associated data. In: INDOCRYPT 2004. LNCS, vol. 3348, pp. 315–327. Springer (2004)
21. Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer (2003)
22. Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer (2004)
23. Jean, J., Nikolić, I., Peyrin, T.: Tweaks and keys for block ciphers: the TWEAKEY framework. In: ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 274–288. Springer (2014)
24. Jetchev, D., Özen, O., Stam, M.: Collisions are not incidental: A compression function exploiting discrete geometry. In: TCC 2012. LNCS, vol. 7194, pp. 303–320. Springer (2012)
25. Krawczyk, H.: LFSR-based hashing and authentication. In: CRYPTO '94. LNCS, vol. 839, pp. 129–139. Springer (1994)
26. Lampe, R., Seurin, Y.: Tweakable blockciphers with asymptotically optimal security. In: FSE 2013. LNCS, vol. 8424, pp. 133–151. Springer (2013)
27. Landecker, W., Shrimpton, T., Terashima, R.S.: Tweakable blockciphers with beyond birthday-bound security. In: CRYPTO 2012. LNCS, vol. 7417, pp. 14–30. Springer (2012)
28. Lee, J.: Towards key-length extension with optimal security: Cascade encryption and xor-cascade encryption. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 405–425. Springer (2013)
29. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer (2002)
30. Mendel, F., Mennink, B., Rijmen, V., Tischhauser, E.: A simple key-recovery attack on McOE-X. In: CANS 2012. LNCS, vol. 7712, pp. 23–31. Springer (2012)
31. Mennink, B.: Optimal collision security in double block length hashing with single length key. In: ASIACRYPT 2012. LNCS, vol. 7658, pp. 526–543. Springer (2012)
32. Minematsu, K.: Improved security analysis of XEX and LRW modes. In: SAC 2006. LNCS, vol. 4356, pp. 96–113. Springer (2007)
33. Minematsu, K.: Beyond-birthday-bound security based on tweakable block cipher. In: FSE 2009. LNCS, vol. 5665, pp. 308–326. Springer (2009)
34. Minematsu, K., Matsushima, T.: Tweakable enciphering schemes from hash-sum-expansion. In: INDOCRYPT 2007. LNCS, vol. 4859, pp. 252–267. Springer (2007)
35. Mitsuda, A., Iwata, T.: Tweakable pseudorandom permutation from generalized Feistel structure. In: Provable Security 2008. LNCS, vol. 5324, pp. 22–37. Springer (2008)
36. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In: SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer (2014)

37. Özen, O.: Design and Analysis of Multi-Block-Length Hash Functions. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne (2012)
38. Patarin, J.: A proof of security in $O(2^n)$ for the Xor of two random permutations. In: ICITS 2008. LNCS, vol. 5155, pp. 232–248. Springer (2008)
39. Procter, G.: A note on the CLRW2 tweakable block cipher construction. Cryptology ePrint Archive, Report 2014/111 (2014)
40. Rogaway, P.: Bucket hashing and its application to fast message authentication. In: CRYPTO '95. LNCS, vol. 963, pp. 29–42. Springer (1995)
41. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer (2004)
42. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: a block-cipher mode of operation for efficient authenticated encryption. In: ACM Conference on Computer and Communications Security. pp. 196–205. ACM, New York (2001)
43. Rogaway, P., Steinberger, J.P.: Security/efficiency tradeoffs for permutation-based hashing. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 220–236. Springer (2008)
44. Rogaway, P., Zhang, H.: Online ciphers from tweakable blockciphers. In: CT-RSA 2011. LNCS, vol. 6558, pp. 237–249. Springer (2011)
45. Sarkar, P.: Efficient tweakable enciphering schemes from (block-wise) universal hash functions. IEEE Transactions on Information Theory 55(10), 4749–4760 (2009)
46. Schroeppel, R.: The Hasty Pudding Cipher (1998), submission to NIST's AES competition
47. Szemerédi, E., Trotter Jr., W.T.: Extremal problems in discrete geometry. Combinatorica 3(3-4), 381–392 (1983)
48. Tao, T.: The Szemerédi-Trotter theorem and the cell decomposition (2009), `http://terrytao.wordpress.com/2009/06/12/the-szemeredi-trotter-theorem-and-the-cell-decomposition`
49. Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: CISC 2005. LNCS, vol. 3822, pp. 175–188. Springer (2005)

A Valid Mixing Functions

We propose a formal definition of valid mixing functions, following upon Def. 1.

Definition 2. *Write $x_{\rho+1} := c$. The mixing functions A_i for $i = 1, \dots, \rho+1$ and B_i for $i = 1, \dots, \rho$ are* valid *if there exists an index $i^* \in \{1, \dots, \rho+1\}$ such that*

(a) $\forall_{i=1,\dots,i^*-1}$ there exists a function $\widehat{A}_i$ such that for all $k, t, y_1, \dots, y_{i-1}, m$:

$$A_i(k, t, y_1, \dots, y_{i-1}, m) = \widehat{A}_i(k, t, y_1, \dots, y_{i-1}) \, ;$$

(b) A_{i^*} is invertible in $m \mapsto x_{i^*}$ for all $k, t, y_1, \dots, y_{i^*-1}$;

(c) $\forall_{i=i^*+1,\dots,\rho+1}$ there exists a function $\widehat{A}_i$ such that for all $k, t, y_1, \dots, y_{i-1}, m$:

$$A_i(k, t, y_1, \dots, y_{i-1}, m) = \widehat{A}_i(k, t, y_1, \dots, y_{i^*-1}, y_{i-1}) \, ,$$

where $\widehat{A}_i$ is furthermore invertible in $y_{i-1} \mapsto x_i$ for all $k, t, y_1, \dots, y_{i^*-1}$;

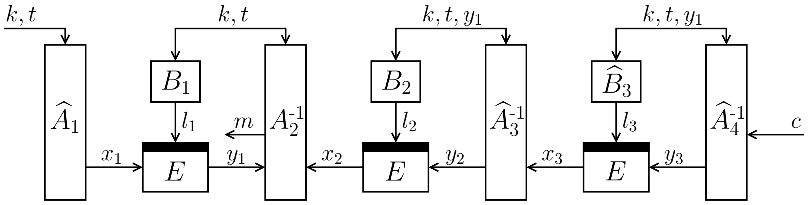

Fig. 4: Inverse of tweakable blockcipher $\widetilde{E}[3]$, where $i^* = 2$

(d) $\forall_{i=i^*+1,\dots,\rho}$ *there exists a function* $\widehat{B}_i$ *such that for all* $k, t, y_1, \dots, y_{i-1}$*:*

$$B_i(k, t, y_1, \dots, y_{i-1}) = \widehat{B}_i(k, t, y_1, \dots, y_{i^*-1})\,.$$

It is straightforward to verify that $\widetilde{E}[\rho]$ is invertible if A_i, B_i are valid mixing functions. Formally, the inverse $\widetilde{E}[\rho]^{-1}$ can be described as follows (for $\rho = 3$ and $i^* = 2$, the inverse $\widetilde{E}[3]^{-1}$ is depicted in Fig. 4):

```
**procedure** $\widetilde{E}[\rho]^{-1}(k, t, c)$
    **for** $i = 1, \dots, i^* - 1$ **do**
        $x_i = \widehat{A}_i(k, t, y_1, \dots, y_{i-1})$
        $l_i = B_i(k, t, y_1, \dots, y_{i-1})$
        $y_i = E(l_i, x_i)$
    **for** $i = \rho, \dots, i^*$ **do**
        $y_i = \widehat{A}_{i+1}^{-1}(k, t, y_1, \dots, y_{i^*-1}, x_{i+1})$
        $l_i = \widehat{B}_i(k, t, y_1, \dots, y_{i^*-1})$
        $x_i = E^{-1}(l_i, y_i)$
    **return** $m = A_{i^*}^{-1}(k, t, y_1, \dots, y_{i^*-1}, x_{i^*})$
```