How to Build Fully Secure Tweakable Blockciphers from Classical Blockciphers

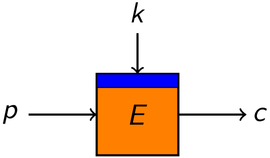
Lei Wang, Jian Guo, Guoyan Zhang, Jingyuan Zhao, Dawu Gu

ASIACRYPT 2016 - Hanoi, Vietnam

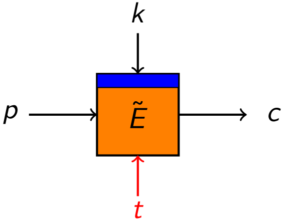
Outline

Tweakable Blockcipher (TBC)

- additional parameter: public tweak t
- more natural primitive for modes of operation
  - ◇ disk encryption, authenticated encryption, etc
- all wires have a size of n bits



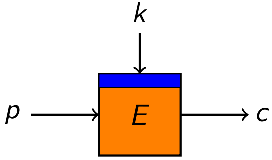
classical blockcipher



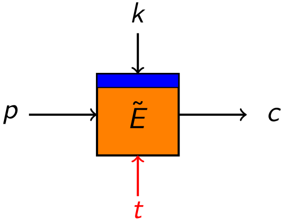
tweakable blockcipher [LRW02]

Tweakable Blockcipher (TBC)

- additional parameter: public tweak t
- more natural primitive for modes of operation
  - ◇ disk encryption, authenticated encryption, etc
- all wires have a size of n bits



classical blockcipher



tweakable blockcipher [LRW02]

Goal of this work

Find TBCs that can achieve full 2^n provable security

Three Approaches to Build TBCs

from the scratch

- Hasty pudding cipher [Sch98], Mercy [Cro00], Threefish [FLS+08]
- a drawback: no security proof

Three Approaches to Build TBCs

from the scratch

- Hasty pudding cipher [Sch98], Mercy [Cro00], Threefish [FLS+08]
- a drawback: no security proof

from blockcipher constructions

- tweak luby-rackoff [GHL+07], generalized feistel [MI08], key-alternating [JNP14,CLS15], etc
- provable security bound: (at most) $2^{2n/3}$ [CLS15]
- still far from full 2^n provable security

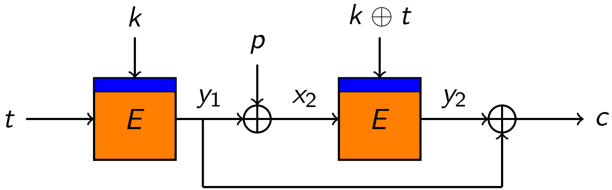
Three Approaches to Build TBCs

from blockcipher as a black-box

- tweak-dependent key (tdk): changing tweak values leads to rekeying blockciphers
- without using tdk
  - ◇ LRW1/2 [LRW02], XEX [Rog04], CLRW2 [LST12], etc
  - ◇ *asymptotically* approach full security [LS13]: $2^{sn/(s+2)}$ security with s blockcipher calls (low efficiency)
  - ◇ in the standard model: blockcipher as PRP
- with using tdk
  - ◇ Minematsu's design [Min09], Mennink's design [Men15]
  - ◇ full 2^n provable security [Men15]:
    the only TBC claiming full 2^n provable security
  - ◇ in the ideal blockcipher model [Men15]

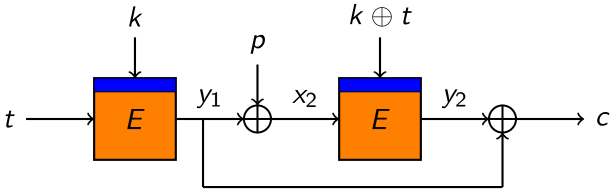
Mennink's Design [Men15]

- tweak-dependent key
- two blockcipher calls
- full 2^n provable security claimed



Mennink's Design [Men15]

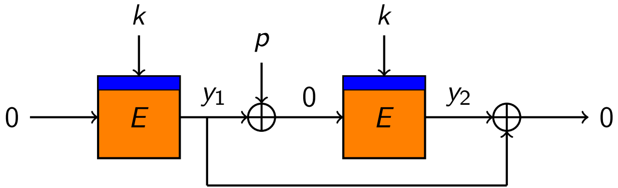
- tweak-dependent key
- two blockcipher calls
- full 2^n provable security claimed



A key-recovery attack can be launched with a birthday-bound complexity

an observation

When $(t, c) = (0, 0)$, it has $y_1 = y_2$, and in turn $x_2 = 0$. Hence, by querying $(t = 0, c = 0)$ to decryption $\widetilde{F2}^{-1}$, the received $p = y_1 = E_k(0)$.



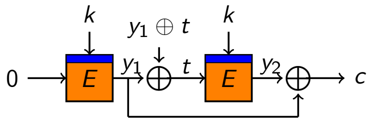
Key-recovery Attack on Mennink's Design $\widetilde{F2}$

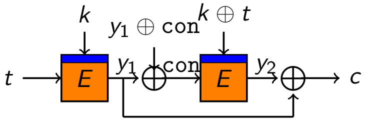
an observation

When $(t, c) = (0, 0)$, it has $y_1 = y_2$, and in turn $x_2 = 0$. Hence, by querying $(t = 0, c = 0)$ to decryption $\widetilde{F2}^{-1}$, the received $p = y_1 = E_k(0)$.

recover $E(k \oplus t, \mathtt{const})$ for any t

1. query $(0, E(k, 0) \oplus t)$ to $\widetilde{F2}$, get c, and compute $E(k, t) = c \oplus E(k, 0)$;
2. query $(t, E(k, t) \oplus \mathtt{const})$ to $\widetilde{F2}$, get c and compute $E(k \oplus t, \mathtt{const}) = c \oplus E(k, t)$.





Key-recovery Attack on Mennink's Design $\widetilde{F2}$

an observation

When $(t, c) = (0, 0)$, it has $y_1 = y_2$, and in turn $x_2 = 0$. Hence, by querying $(t = 0, c = 0)$ to decryption $\widetilde{F2}^{-1}$, the received $p = y_1 = E_k(0)$.

recover $E(k \oplus t, \texttt{const})$ for any t

1. query $(0, E(k, 0) \oplus t)$ to $\widetilde{F2}$, get c, and compute $E(k, t) = c \oplus E(k, 0)$;
2. query $(t, E(k, t) \oplus \texttt{const})$ to $\widetilde{F2}$, get c and compute $E(k \oplus t, \texttt{const}) = c \oplus E(k, t)$.

recover the key by a meet-in-the-middle procedure

Online. recover $E(k \oplus t, \texttt{const})$ for $2^{n/2}$ tweaks t;

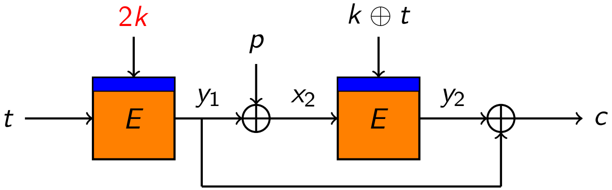
Offline. compute $E(\ell, \texttt{const})$ for $2^{n/2}$ values ℓ;

MitM. recover $k = \ell \oplus t$ from $E(k \oplus t, \texttt{const}) = E(\ell, \texttt{const})$.

Remark on Flaw and Patch of $\widetilde{F2}$

a small flaw in the original proof

In the proof, under the condition that the attacker cannot guess the key correctly (that is, (12a) defined in [M15] is not set), it claimed that the distribution of y_1 is independent from y_2. However, when the tweak $t = 0$, both the two blockcipher calls share the same key, and therefore the distribution of their outputs are highly related.

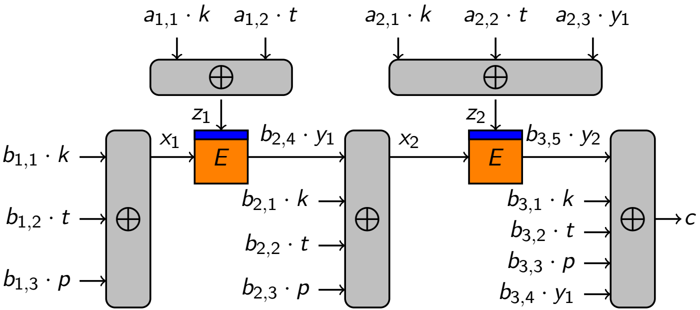


patched $\widetilde{F2}$ by the designer: full 2^n provable security

Outline

The Target Construction

- $a_{i,j},\ b_{i,j} \in \{0, 1\}$
- simple XORs as linear mixing
- this talk focuses on the case of two blockcipher calls
  - ◇ one blockcipher call with linear mixing can reach at most birthday-bound security [Men15]



Invertibility of Target Construction

Constraint 1

plaintext p must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.

Invertibility of Target Construction

Constraint 1

plaintext p must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.

Constraint 2

if y_1 is computed depending on plaintext p, it must not be used to compute z_2. Thus, if $b_{1,3} = 1$, $a_{2,3}$ must be 0.

Invertibility of Target Construction

Constraint 1

plaintext p must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.

Constraint 2

if y_1 is computed depending on plaintext p, it must not be used to compute z_2. Thus, if $b_{1,3} = 1$, $a_{2,3}$ must be 0.

Constraint 3

if both y_1 and y_2 are computed depending on plaintext p, they must not be used both as inputs to the final linear mixing. Thus, if $b_{1,3}$ and $b_{2,4}$ are 1, $b_{3,4}$ must be 0.

Invertibility of Target Construction

Constraint 1

plaintext p must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.

Constraint 2

if y_1 is computed depending on plaintext p, it must not be used to compute z_2. Thus, if $b_{1,3} = 1$, $a_{2,3}$ must be 0.

Constraint 3

if both y_1 and y_2 are computed depending on plaintext p, they must not be used both as inputs to the final linear mixing. Thus, if $b_{1,3}$ and $b_{2,4}$ are 1, $b_{3,4}$ must be 0.

Others

we always assume both blockciphers are indeed involved in the encrytion/decryption process.

Design Goal

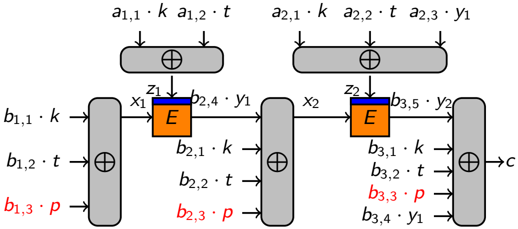
- first and top-priority goal: full 2^n provable security
- second goal: the minimum number of blockcipher calls
- third goal: (comparably) high efficiency of changing a tweak
  - ◇ start with (at most) one tweak-dependent key

Outline

Three Types of Instances

According to the position of plaintext p (Constraint 1)

- Type I: $b_{1,3} = 1,\ b_{2,3} = 0,\ b_{3,3} = 0$
- Type II: $b_{1,3} = 0,\ b_{2,3} = 1,\ b_{3,3} = 0$
- Type III: $b_{1,3} = 0,\ b_{2,3} = 0,\ b_{3,3} = 1$



Constraint 1

plaintext p must be used in exactly one linear mixing. Thus, one of $\{b_{3,1}, b_{3,2}, b_{3,3}\}$ is 1, and the other two are 0.

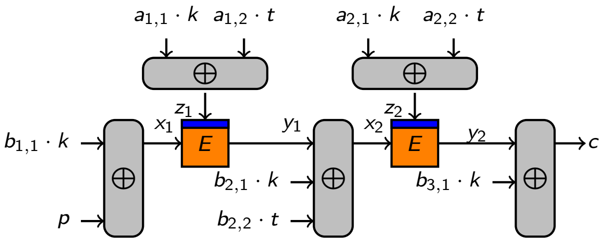
Type I

divided into two cases

Case (1). z_1 is a tweak-dependent key

Case (2). z_2 is a tweak-dependent key

$\star$ each case is divided into 4 subcases depending on $(a_{1,1}, b_{1,1})$.



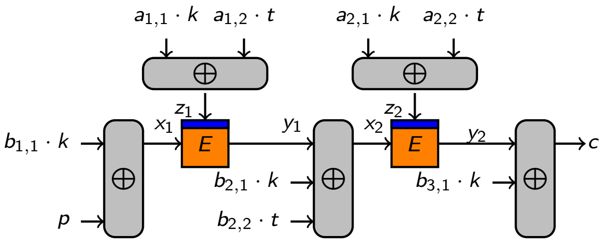
Type I

divided into two cases

Case (1). z_1 is a tweak-dependent key

Case (2). z_2 is a tweak-dependent key

$\star$ each case is divided into 4 subcases depending on $(a_{1,1}, b_{1,1})$.

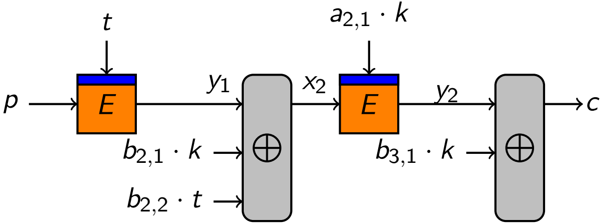


search result

Type I instances with one tweak-dependent key have at most birthday-bound security.

Subcase (1.1) as an example

- $(a_{1,1}, b_{1,1}) = (0, 0)$;
- the first blockcipher call is independent from k;
- y_1 can be obtained by querying $E(\cdot, \cdot)$, and hence essentially one blockcipher call in attackers' view;
- at most birthday-bound security [M15]

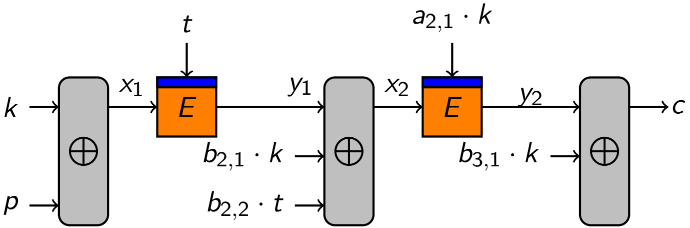


Subcase (1.2) as an example

- $(a_{1,1}, b_{1,1}) = (0, 1)$

an observation

for any pair (t, p, c) and (t', p', c'), it has that $c = c'$ implies $y_1 \oplus y_1' = b_{2,2} \cdot (t \oplus t')$.



Subcase (1.2) as an example

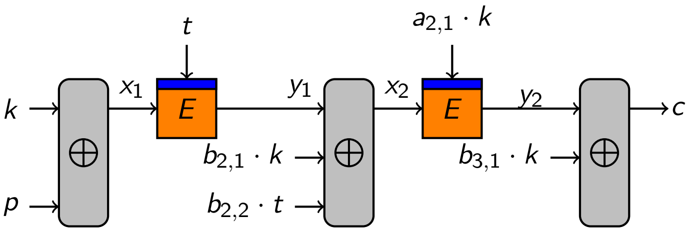
recover k by a meet-in-the-middle procedure

fix two distinct tweaks t and t';

Online. collect $p \oplus k \oplus E_{t'}^{-1}(E_t(p \oplus k) \oplus b_{2,2} \cdot (t \oplus t'))$ for $2^{n/2}$ distinct paintexts p;

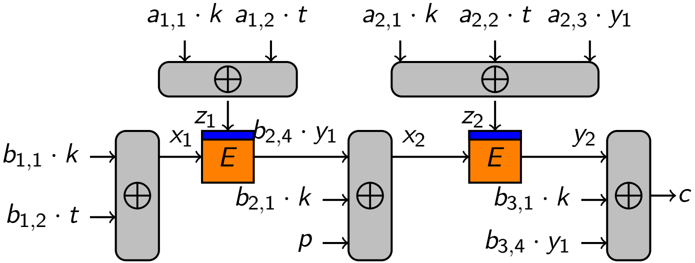
Offline. collect $\ell \oplus E_{t'}^{-1}(E_t(\ell) \oplus b_{2,2} \cdot (t \oplus t'))$ for $2^{n/2}$ distinct ℓ;

MitM. compute $k = p \oplus \ell$ from an online/offline collision

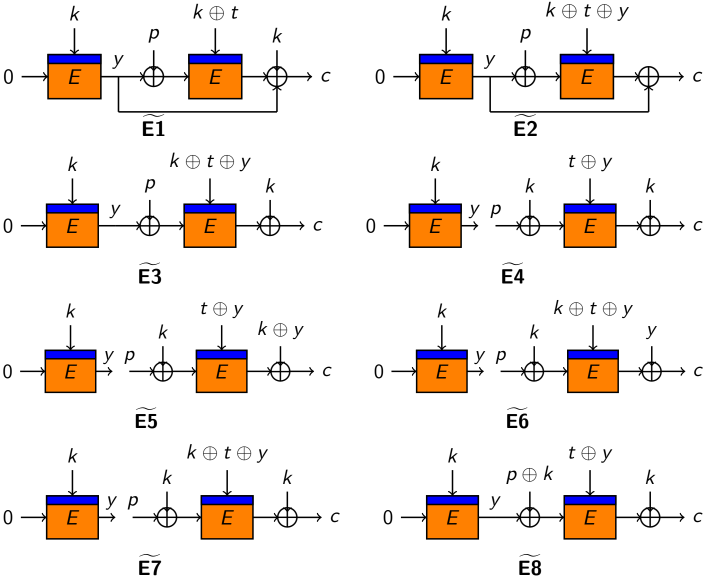


Type II

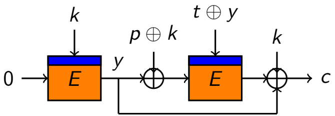
- two cases depending on z_1 or z_2 as a tweak-dependent key;
- each case is further divided into several subcases;
- 32 instances that no attack can be found



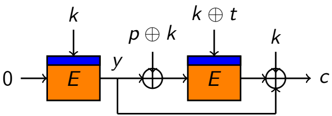
32 Plausible TBCs



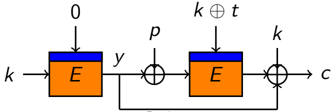
32 Plausible TBCs



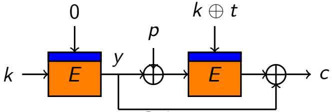
$\widetilde{\mathbf{E9}}$



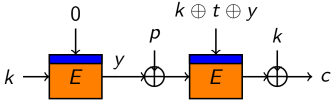
$\widetilde{\mathbf{E10}}$



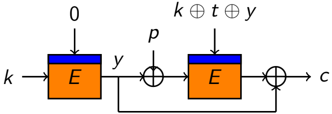
$\widetilde{\mathbf{E11}}$



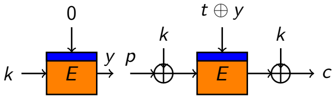
$\widetilde{\mathbf{E12}}$



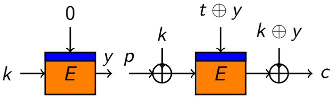
$\widetilde{\mathbf{E13}}$



$\widetilde{\mathbf{E14}}$

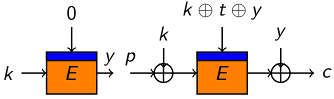


$\widetilde{\mathbf{E15}}$

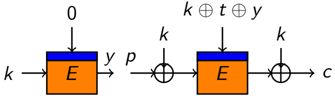


$\widetilde{\mathbf{E16}}$

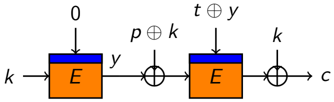
32 Plausible TBCs



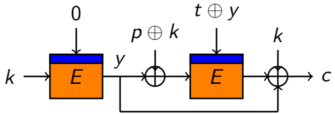
$\widetilde{\mathbf{E17}}$



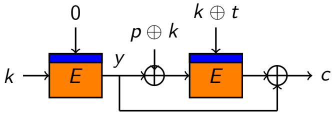
$\widetilde{\mathbf{E18}}$



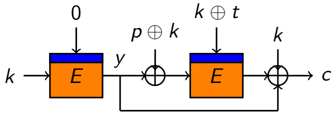
$\widetilde{\mathbf{E19}}$



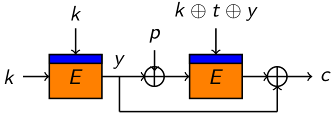
$\widetilde{\mathbf{E20}}$



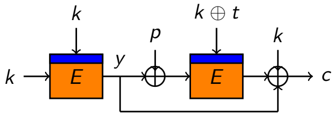
$\widetilde{\mathbf{E21}}$



$\widetilde{\mathbf{E22}}$

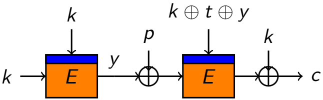


$\widetilde{\mathbf{E23}}$

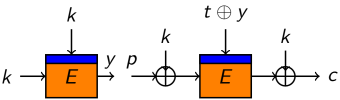


$\widetilde{\mathbf{E24}}$

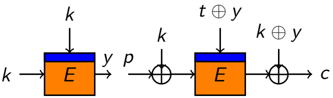
32 Plausible TBCs



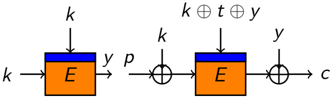
$\widetilde{\textbf{E25}}$



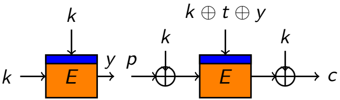
$\widetilde{\textbf{E26}}$



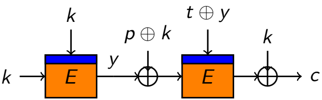
$\widetilde{\textbf{E27}}$



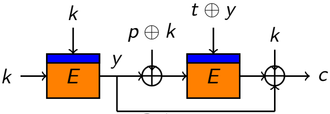
$\widetilde{\textbf{E28}}$



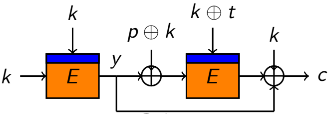
$\widetilde{\textbf{E29}}$



$\widetilde{\textbf{E30}}$



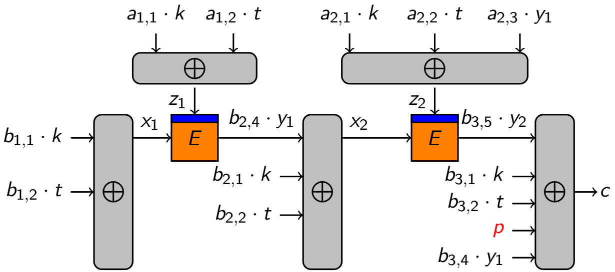
$\widetilde{\textbf{E31}}$



$\widetilde{\textbf{E32}}$

Type III

- plaintext p and ciphertext c are *linearly* related. Hence Type III instances are not secure.



Outline

Provable Security

Theorem

Let $\widetilde{E}$ be any tweakable blockcipher construction from the set of $\widetilde{E1}, \ldots, \widetilde{E32}$. Let q be an integer such that $q < 2^{n-1}$. Then the following bound holds.

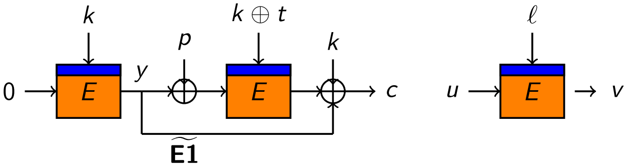
$$\mathbf{Adv}_{\widetilde{E}}^{\widetilde{\mathrm{sprp}}}(q) \le \frac{10q}{2^n}.$$

Proof Sketch for $\widetilde{E1}$

- the h-coefficient technique [P08, CS14]
- release k and $y = E(k, 0)$ to the distinguisher after the interaction and before the final decision
- distinguisher gets all the input-output tuples of E, divided into
  - ◇ $\mathcal{T}^1 = \{(0, k, y) : y = E(k, 0)\}$;
  - ◇ $\mathcal{T}^2 = \{(z, x, y) : E(z, x) = y\}$ from queries to $\widetilde{E1}$ (the 2nd E);
  - ◇ $\mathcal{T}^3 = \{(\ell, u, v) : E(\ell, u) = v\}$ from (offline) queries to E;

Good View

$\mathcal{T}^1 \cap \mathcal{T}^2 = \mathcal{T}^1 \cap \mathcal{T}^3 = \mathcal{T}^2 \cap \mathcal{T}^3 = \emptyset \implies$ the distinguisher fails.



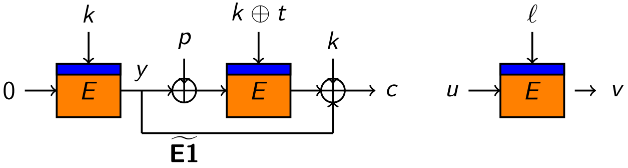
Proof Sketch for $\widetilde{E1}$

- $\Pr\left[\mathcal{T}^1 \cap \mathcal{T}^3 \neq \emptyset\right] \leq \frac{q}{2^n-q-1}$;
- $\Pr\left[\mathcal{T}^1 \cap \mathcal{T}^2 \neq \emptyset\right] \leq \frac{2q}{2^n-q-1}$;
- $\Pr\left[\mathcal{T}^2 \cap \mathcal{T}^3 \neq \emptyset\right] \leq \frac{2q^2}{(2^n-q-1)^2}$;

upper bound of probability of bad events

Supposing $q < 2^{n-1}$, we have that

$$\frac{q}{2^n-q-1} + \frac{2q}{2^n-q-1} + \frac{2q^2}{(2^n-q-1)^2} \leq \frac{10q}{2^n}$$



Outline

Conclusion

We find 32 TBCs with full 2^n provable security

- each TBC uses two blockcipher calls
- save one blockcipher call by precomputing and storing the subkey
- in the ideal blockcipher model

tweakable blockciphers	key size	security ($\log_2$)	cost E	cost $\otimes/h$	tdk	reference
LRW1	n	$n/2$	1	0	N	[LRW02]
LRW2	$2n$	$n/2$	1	2	N	[LRW02]
XEX	n	$n/2$	1	0	N	[R04]
LRW2[2]	$4n$	$2n/3$	2	2	N	[LST12]
LRW2[s]	$2sn$	$sn/(s+2)$	s	s	N	[LS13]
Min	n	$\max\{n/2, n-\lvert t\rvert\}$	2	0	Y	[M09]
$\widetilde{F}[1]$	n	$2n/3$	1	1	Y	[M15]
$\widetilde{F}[2]$	n	$n/2$	2	0	Y	[M15]
patched $\widetilde{F}[2]$	n	n	2	0	Y	[M15]
$\widetilde{E1}, \ldots, \widetilde{E32}$	n	n	2 (1)	0	Y	Ours

$\otimes/h$ stands for multiplications or universal hashes;

tdk stands for the tweak-dependent key. 'N' refers to not using tdk, and 'Y' refers to using tdk;

$|t|$ stands for the bit length of the tweak;

Thank you

https://eprint.iacr.org/2016/876