

On the provable security of BEAR/LION schemes

Bunny 2011

Matteo Piva

University of Trento

12nd September, 2011

Acknowledgments

This is a joint work with:

- prof. Massimiliano Sala (my supervisor),
- Lara Maines,
- Anna Rimoldi.

Acknowledgments

For their comments and suggestions the authors would like to thank E. Bellini (Univ. of Trento), G. Morgari and M. Coppola (both with Telsy).

This work has been supported by TELS Y Elettronica e Telecomunicazioni, an Italian company working in Information and Communication Security.

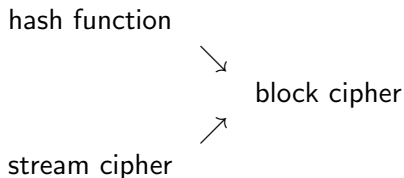
- 1 Preliminaries
- 2 Our results
- 3 Conclusions
- 4 Morin's attack

PRELIMINARIES

Two primitives, one primitive?

An interesting problem in cryptography is how to construct, given some cryptographic primitives, another primitive.

In particular, we are interested in **building** a block cipher starting **from** a (keyed) hash function and a stream cipher:



Often stream ciphers and hash functions are **already implemented** in hardware solutions, with good timing performance. And achieving a satisfactory implementation of a (traditional) block cipher is a challenge.

Luby-Rackoff construction

Luby and Rackoff in

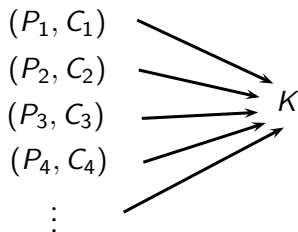


M. Luby and C. Rackoff, *How to construct pseudorandom permutations from pseudorandom functions*, SIAM J. Comput. **17** (1988), no. 2, 373–386.

propose a very general way to obtain **one** pseudo-random permutation **from two** pseudo-random functions, and this result might be used to design a block cipher.

Definition

Let $n \geq 1$. We say that \mathcal{A}_n is a **key-recovery oracle** for a given block cipher, if it is able to find **efficiently** the key, given **any** set of n plaintext-ciphertext pairs $\{ (P_i, C_i) \}_{1 \leq i \leq n}$.



The oracles

- If $n = 1$ then we call \mathcal{A}_1 a **single-pair** oracle.
- If $n \geq 2$ then we call \mathcal{A}_n a **multi-pair** oracle.

Note that \mathcal{A}_1 is **much more powerful** than \mathcal{A}_n , since it is able to recover the key using **just one** pair rather than \mathcal{A}_n , which needs more pairs.

Indeed, if $N \geq n$, then the existence of \mathcal{A}_n obviously implies the existence of \mathcal{A}_N (just dump $N - n$ pairs!).

Block cipher resistance to key-recovery attacks

We can classify informally block ciphers w.r.t. to their resistance to key-recovery attacks.

very strong \rightarrow there is not \mathcal{A}_n

strong \rightarrow there is not \mathcal{A}_1

weak \rightarrow there is \mathcal{A}_n

very weak \rightarrow there is \mathcal{A}_1

Caution!

In this presentation we are neglecting several aspects of the problem, that would otherwise lead us too far. In particular:

- We will not delve into complexity details and so you should use “common sense”; for example, you must consider also the **effort** by \mathcal{A}_n in collecting its input (and so n cannot be huge).
- although we present our oracle as a known-plaintext attack, our proofs can be modified (becoming more complex) to a chosen-plaintext attack or even a chosen-ciphertext attack; however, I **doubt** that our proofs can be modified easily to cover also an adaptive version.
- other kinds of attacks could be considered, such as global-reconstruction, partial-key recovery, etc. .


We make these assumptions:

- $\mathbb{F} = \mathbb{F}_2$,
- **small space** \mathbb{F}^l , **big space** \mathbb{F}^r ,
- the key space is $\mathcal{K} = \mathbb{F}^{2k}$ or $\mathcal{K} = \mathbb{F}^{3k}$,
- $K \in \mathbb{F}^k$, $L \in \mathbb{F}^l$, $R \in \mathbb{F}^r$, $r > l$, $k \geq l$,
- $S : \mathbb{F}^l \rightarrow \mathbb{F}^r$ is an injective function (stream cipher),
- $H : \mathbb{F}^r \rightarrow \mathbb{F}^l$ is a surjective function (hash function),
- \mathcal{H} is a set of surjective functions $H_K : \mathbb{F}^r \rightarrow \mathbb{F}^l$ (keyed hash function) satisfying:

for a random R 's the map $H^R : \mathbb{F}^k \mapsto \mathbb{F}^l$, $H^R(K) = H_K(R)$, is surjective.

The ciphers by Anderson and Biham

In

 R. Anderson and E. Biham, *Two practical and provably secure block ciphers: BEAR and LION*, Proc. of FSE 1996, LNCS, vol. 1039, Springer, 1996, pp. 113–120.

three block ciphers are presented:

- BEAR,
- LION,
- LIONESS.

They are built from hash functions and stream ciphers. In the same article several results on their provable security are shown (and a few are claimed without proofs).

Let $k > l$, $\mathcal{K} = \mathbb{F}^{2k}$.

ENCRYPTION	DECRYPTION
$\bar{L} = L + H_{K_1}(R)$	$\bar{L} = L' + H_{K_2}(R')$
$R' = R + S(\bar{L})$	$R = R' + S(\bar{L})$
$L' = \bar{L} + H_{K_2}(R')$	$L = \bar{L} + H_{K_1}(R)$

Definition (S property)

S is **one-way** if it is hard to find the seed X given any random Y such that $Y = S(X)$.

The following theorem is proved in the article of Anderson and Biham.

Theorem

If there exists \mathcal{A}_1 for BEAR, then S is not one-way.

The following corollary is obvious.

Corollary

If S is one-way then \mathcal{A}_1 does not exist for BEAR.

Definition (properties of H and \mathcal{H})

H is:

- **one-way** if it is hard to find the seed X given any random Y such that $Y = H(X)$;
- **collision-free** if it is hard to find unequal X and Y such that $H(Y) = H(X)$.

Similarly for the keyed function \mathcal{H} .

The following theorem is proved by Anderson and Biham.

Theorem

If there exists \mathcal{A}_1 for BEAR, then \mathcal{H} is neither one-way nor collision-free.

Corollary

If \mathcal{H} is one-way and collision-free, then \mathcal{A}_1 does not exist for BEAR.

Let $k = l$, $\mathcal{K} = \mathbb{F}^{2k}$.

ENCRYPTION	DECRYPTION
$\bar{R} = R + S(L + K_1)$	$\bar{R} = R' + S(L' + K_2)$
$L' = L + H(\bar{R})$	$L = L' + H(\bar{R})$
$R' = \bar{R} + S(L' + K_2)$	$R = \bar{R} + S(L + K_1)$

Anderson and Biham claim (without proofs) also similar results for LION.

Theorem

If there exists \mathcal{A}_1 for LION, then S is not one-way.

Corollary

If S is one-way then \mathcal{A}_1 does not exist for LION.

Theorem

If there exists \mathcal{A}_1 for LION, then H is neither one-way nor collision-free.

Corollary

If H is one-way and collision-free then \mathcal{A}_1 does not exist for LION.

ENCRYPTION	DECRYPTION
$\bar{R} = R + S(L + K_1)$	$\bar{L} = L' + H_{K_4}(R')$
$\bar{L} = L + H_{K_2}(\bar{R})$	$\bar{R} = R' + S(\bar{L} + K_3)$
$R' = R + S(\bar{L} + K_3)$	$L = \bar{L} + H_{K_2}(\bar{R})$
$L' = \bar{L} + H_{K_4}(R')$	$R = \bar{R} + S(L + K_1)$

LIONESS

The following results are claimed by Anderson and Biham.

Theorem

If there exists \mathcal{A}_1 for LIONESS, then

- S is not one-way
- \mathcal{H} is neither one-way nor collision-free.

Corollary

In LIONESS if

*H is collision-free **or** \mathcal{H} is one-way **or** S is one-way*

*then \mathcal{A}_1 does **not** exist.*

OUR RESULTS

Our results in one sentence

Our contribution consists in showing that **no** \mathcal{A}_n exists for BEAR, LION and LIONESS.

We are able to get this improvement by using slightly different hypotheses on the primitives.

Definition

Given a keyed hash function $\mathcal{H} = \{ H_K \}_{K \in \mathbb{F}^k}$, we say that

\mathcal{H} is **key-resistant**

if, given a pair (Z, R) such that $Z = H_K(R)$ for a random K and a random R , then it is **hard** to find K .

In other words, the equation

$$Z = H_K(R)$$

is **hard** to solve in K .

S key-resistance

Definition

Given a stream cipher S , we say that

S is **key-resistant**

if, given a pair (Z, L) such that $Z = S(L + K_1)$ for random $K_1, L \in \mathbb{F}'$, then it is hard to find K_1 .

Remark

*We could have a different action induced by the keys, say, $S(\tau_K(L))$ instead of $S(L + K)$. All subsequent results will still hold, provided the action is **regular** (i.e., when $\{\tau_K\}_{K \in \mathcal{K}}$ is a regular subgroup of $\text{Sym}(\mathbb{F}')$).*

Our improvements for BEAR

We can obtain the following result for BEAR if \mathcal{H} is key-resistant.

Theorem (\mathcal{H} key-resistant $\implies \nexists \mathcal{A}_n$)

*If there exists \mathcal{A}_n for BEAR, then \mathcal{H} is **not** key-resistant.*

As usual, the corollary is obvious.

Corollary

*If the (keyed) hash function is key-resistant, then **no multi-pair oracle** exists for BEAR.*

Proof.

We must solve (in K) the equation: $Z = H_K(R)$.

Let us choose a set $\{L_i\}_{1 \leq i \leq n} \subset \mathbb{F}^l$ and consider the set of plaintexts $\{(L_i, R)\}_{1 \leq i \leq n}$.

It is possible to generate a set of ciphertexts $\{(L'_i, R'_i)\}_{1 \leq i \leq n}$ by choosing $K_1 = K$ and any $K_2 \in \mathbb{F}^k$. Indeed, we can compute:

$$\bar{L}_i = L_i + Z$$

$$R'_i = R + S(L_i + Z)$$

$$L'_i = L_i + Z + H_{K_2}(R'_i).$$

With $\{((L_i, R), (L'_i, R'_i))\}_{1 \leq i \leq n}$ as input, \mathcal{A}_n outputs both K_2 , which was already known, and K_1 , which was unknown. □

We have not been able to obtain a similar result for BEAR with similar hypotheses on the **stream cipher** S .

We can consider a variation of BEAR's scheme. We call this scheme **BEAR2**.

ENCRYPTION	DECRYPTION
$\bar{L} = L + H_{K_1}(R)$	$\bar{L} = L' + H_{K_3}(R')$
$R' = R + S(\bar{L} + K_2)$	$R = R' + S(\bar{L} + K_2)$
$L' = \bar{L} + H_{K_3}(R')$	$L = \bar{L} + H_{K_1}(R)$

Theorem (\mathcal{H} key-resistant $\implies \nexists \mathcal{A}_n$)

If there exists \mathcal{A}_n for BEAR2, then \mathcal{H} is not key-resistant.

Proof.

Obvious adaption of the proof of Th. 16.

This time we choose any K_2 and K_3 , obtaining K_1 again. □

Our results on BEAR2

Theorem (S key-resistant $\implies \nexists \mathcal{A}_n$)

If there exists \mathcal{A}_n for BEAR2, then S is not key-resistant.

Corollary

If the stream cipher is key-resistant, no multi-pair key-recovery oracle exists for BEAR2.

Proof.

We must solve (in K) the equation: $Z = S(X + K)$.

Let us choose a set $\{R_i\}_{1 \leq i \leq n} \subset \mathbb{F}^r$ and any $K_1, K_3 \in \mathbb{F}^k$. It is possible to generate plaintext/ciphertext pairs by choosing $K_2 = K$ and computing $L_i = X + H_{K_1}(R_i)$, so that we can encrypt:

$$\bar{L}_i = L_i + H_{K_1}(R_i) = X$$

$$R'_i = R_i + Z$$

$$L'_i = X + H_{K_3}(R'_i).$$

We give in input to \mathcal{A}_n the set $\{(L_i, R_i), (L'_i, R'_i)\}_{1 \leq i \leq n}$. \mathcal{A}_n returns K_1 , K_3 which were already known, and K_2 , which was unknown. \square

We can summarize our findings on BEAR2 in the following corollary.

Corollary

No multi-pair key-recovery oracle exists for BEAR2 if the hash function is key-resistant or the stream cipher is key-resistant.

Note that for BEAR the non-existence of \mathcal{A}_n does not follow from properties of S but only from those of \mathcal{H} .

Our improvements for LION

We can obtain the following result for LION if S is key-resistant.

Theorem (S key-resistant $\implies \nexists \mathcal{A}_n$)

For LION, if there exists \mathcal{A}_n then S is not key-resistant.

Corollary

If the stream cipher is key-resistant, no (efficient) multi-pair oracle exists for LION.

Proof.

Let us choose a set $\{R_i\}_{1 \leq i \leq n} \subset \mathbb{F}^r$ and consider the set of plaintexts $\{(L, R_i)\}_{1 \leq i \leq n}$. It is possible to generate a set of ciphertexts $\{(L'_i, R'_i)\}_{1 \leq i \leq n}$ by choosing any sub-key K_2 and computing:

$$\bar{R}_i = R_i + S(L + K_1) = R_i + Z$$

$$L'_i = L_i + H(R_i + Z)$$

$$R'_i = R_i + Z + S(L'_i + K_2).$$

Using \mathcal{A}_n we can find K_2 , which was already known, and K_1 , which was unknown. □

Note that we are not able to obtain a similar result from properties of H .

We can consider a variation of LION's scheme in order to obtain a result similar to the previous theorem, when \mathcal{H} is key-resistant.

We call this scheme LION2.

ENCRYPTION	DECRYPTION
$\bar{R} = R + S(L + K_1)$	$\bar{R} = R' + S(L' + K_3)$
$L' = L + H_{K_2}(\bar{R})$	$L = L' + H_{K_2}(\bar{R})$
$R' = \bar{R} + S(L' + K_3)$	$R = \bar{R} + S(L + K_1)$

Theorem (S key-resistant $\implies \nexists \mathcal{A}_n$)

If there exists \mathcal{A}_n for LION2 then S is not key-resistant.

Proof.

Obvious adaption of the proof of Th. 22. □

Theorem (\mathcal{H} key-resistant $\implies \nexists \mathcal{A}_n$)

If there exists \mathcal{A}_n for LION2 then \mathcal{H} is not key-resistant.

Corollary

If the hash function is key-resistant, no (efficient) multi-pair oracle exists for LION2.

Proof.

Let us choose a set $\{L_i\}_{1 \leq i \leq n} \subset \mathbb{F}^l$ and any sub-keys $K_1, K_3 \in \mathbb{F}^l$. It is possible to generate plaintext/ciphertext pairs by choosing

$R_i = X + S(L_i + K_1)$ and computing:

$$\bar{R}_i = R_i + S(L_i + K_1) = X + S(L_i + K_1) + S(L_i + K_1) = X$$

$$L'_i = L_i + Z$$

$$R'_i = X + S(L'_i + K_3).$$

We give in input to \mathcal{A}_n the set $\{(L_i, R_i), (L'_i, R'_i)\}$. \mathcal{A}_n returns K_1, K_3 , which were already known, and K_2 , which was unknown. □

We can summarize our findings on LION2 in the following corollary.

Corollary

*No efficient multi-pair key-recovery oracle exists for LION2 **if** the hash function is key-resistant **or** the stream cipher is key-resistant.*

Recall that LION's resistance to key-recovery attacks is guaranteed only by the key-resistance of S .

Our improvements for LIONESS

Theorem (S key-resistant $\implies \nexists \mathcal{A}_n$)

For LIONESS, if there exists \mathcal{A}_n then S is not key-resistant.

Proof.

Let us choose a set $\{R_i\}_{1 \leq i \leq n} \subset \mathbb{F}^r$ and consider the set of plaintexts $\{(L, R_i)\}_{1 \leq i \leq n}$. It is possible to generate a set of ciphertexts $\{(L', R'_i)\}_{1 \leq i \leq n}$ by choosing any sub-keys K_2, K_3, K_4 and computing:

$$\bar{R}_i = R_i + Z$$

$$\bar{L}_i = L + H_{K_2}(\bar{R}_i)$$

$$R'_i = R_i + S(\bar{L}_i + K_3)$$

$$L' = \bar{L}_i + H_{K_4}(R'_i).$$

Using \mathcal{A}_n we can find K_2, K_3, K_4 , which were already known, and K_1 , which was unknown.

Our improvements for LIONESS

Theorem (\mathcal{H} key-resistant $\implies \nexists \mathcal{A}_n$)

For LIONESS, if there exists \mathcal{A}_n then \mathcal{H} is not key-resistant.

Proof.

Let us choose a set $\{L_i\}_{1 \leq i \leq n} \subset \mathbb{F}^l$ and consider the set of ciphertexts $\{(L'_i, R'_i)\}_{1 \leq i \leq n}$. It is possible to generate a set of plaintexts $\{(L_i, R_i)\}_{1 \leq i \leq n}$ by choosing any sub-keys K_1, K_2, K_3 and decrypting:

$$\overline{L}_i = L'_i + Z$$

$$\overline{R}_i = R'_i + S(\overline{L}_i + K_3)$$

$$L_i = \overline{L}_i + H_{K_2}(\overline{R}_i)$$

$$R_i = \overline{R}_i + S(L_i + K_1).$$

Using \mathcal{A}_n we can find K_1, K_2, K_3 , which were already known, and K_4 , which was unknown.

CONCLUSIONS

Conclusions on BEAR

Let us consider a keyed hash function with a very weak requirement, i.e., that it is (more or less) surjective both fixing the key and with respect to the keys. Anderson and Biham prove that **no single-pair** oracle exists for BEAR, under the assumption that **the stream seed is difficult to recover OR the hash function is collision resistant OR the hash preimage is hard to recover.**

Conclusions on BEAR

We prove that no **multi-pair** oracle exists for BEAR under the assumption that **the hash function is key-resistant**.

We also suggest a slight modification of BEAR, BEAR2, where we can prove that **no multi-pair** oracle exists under the assumption that **the hash function is key-resistant OR the stream cipher is key-resistant**.

Conclusions on LION

Anderson and Biham claim that **no single-pair** oracle exists for LION, under the assumption that

the stream seed is difficult to recover OR the hash function is collision resistant OR the hash preimage is hard to recover.

Conclusions on LION

As in the case of BEAR, we prove that no multi-pair oracle exists for LION under the assumption that **the stream cipher is key-resistant**, which is equivalent to “the stream preimage is hard to recover” in many practical situations.

We also suggest a slight modification of LION, LION2, where we can prove that **no multi-pair** oracle exists under the assumption that

the hash function is key-resistant OR the stream cipher is key-resistant.

Conclusions on LIONESS

As regards key-recovery attacks, LIONESS's virtues are the sum of LION's and BEAR's virtues.

So it is possible to prove the non-existence of **one-pair** oracles using the **assumptions by Anderson and Biham**, but we can indeed prove the non-existence of **multi-pair** oracles under only the **key-resistance** assumption.

MORRIN'S ATTACK

Morrin's attack

In



P. Morin, *Provably secure and efficient block ciphers*, Proc. of SAC 1996, 1996, pp. 30–37.

P. Morin proposed an attack on BEAR which can be generalized to any Luby-Rackoff scheme. We provide a sketch of his attack.

BEAR

ENCRYPTION	DECRYPTION
$\bar{L} = L + H_{K_1}(R)$	$\bar{L} = L' + H_{K_2}(R')$
$R' = R + S(\bar{L})$	$R = R' + S(\bar{L})$
$L' = \bar{L} + H_{K_2}(R')$	$L = \bar{L} + H_{K_1}(R)$

The complexity of a brute force search on BEAR is $= 2^{2k}$

Given a plaintext/ciphertext pair, $P = (L, R)$, $C = (L', R')$ the attacker

- computes $L + H_{K_1}(R)$ for all 2^k possible values of K_1
- computes $L' + H_{K_2}(R')$ for all 2^k possible values of K_2
- compares these tables of values until he finds
 $L + H_{K_1}(R) = L' + H_{K_2}(R')$
- tests if (K_1, K_2) is the correct pairs by checking:
 $S(L + H_{K_1}(R)) = R + R'$

the complexity of Morrin's attack is 2^{k+1} (recall that $k > l$).

The attack by Morin has somehow diminished the confidence in the robustness of these schemes.

However, the attack succeeds only because its brute-force search on the round functions **contradicts the key-resistance** of the hash function and of the stream function.

So, whenever \mathcal{H} or S remain key-resistant, both LION and BEAR are immune to such attacks.

Thank you for your attention