SAC Project

Bogdan Ursu, Adina Nedelcu

University Joseph Fourier Grenoble

bogdanbear@gmail,com, nedelcu.adina.ioana@gmail.com

November 16, 2014

Implementation Problems

- Memory Leakage
- Heap Corruption
- Bad choice of the block cipher mode
- Inefficient library functions

Solutions

Tweaks

- Data storage in binary format
- Minimize data copy using dynamic memory
- ECB as block cipher mode
- OOP approach (ex. Wrapper ...)
- Buffer Comparison

Table Creation

- Ensure uniqueness of SPs by using a binary search tree
- Sort table by endpoints before writing
- Pair pointers

Endpoint matching

- Binary Search + Exponential Search

Hellman TMTO

We have used fairly large tables, in spite of the fact that Hellman's initial recommendation was that the tables should satisfy $m * t^2 = N$

Chain overlapping

1. Hypothesis: Chain overlapping and looping are frequent
2. Testing chain convergence
3. Our success rate for random keys iterated with the step function for 512/1024/2048 times is much lower

Custom function

Our custom function extracts the 4 less significant bits of the fifth to last byte of ciphertext. Then, for each one of the last 4 bytes of ciphertext it extract the 7 most significant bits.

It is very efficient, since it requires just a buffer copy, taking the reminder modulo 32, then an assignment.

It behaves rather similar to the LSB function, statistics-wise.

Hellman: LSB versus custom function

Success rate LSB (nr. of hits per 1000 tests)

tm	**524288**	**1048576**	**2097152**
512	17	23	32
1024	13	20	30
2048	15	24	34

Success rate custom function (nr. of hits per 1000 tests)

tm	**524288**	**1048576**	**2097152**
512	18	24	36
1024	17	24	30
2048	19	21	-

Similar success rates for both functions.

Hellman: LSB versus custom

Alarm rate LSB

tm	**524288**	**1048576**	**2097152**
512	16.6	31.2	66.2
1024	62.9	124.1	254.8
2048	257.3	524.1	1032.4

Alarm rate custom

tm	**524288**	**1048576**	**2097152**
512	16.0	31.7	65.4
1024	62.6	126.8	260.8
2048	254.3	519.5	-

Similar alarm rates for both functions.

Using both functions

Stats	**1LSB**	**2CUST**	**both**	**same size LSB**
Success	30	24	54	47
Alarms	254	267	500	527
Nr. encr.	86870	95054	173307	185250

Better results using both tables, for the same memory usage.

Rivest TMTO

The overall success rate for our implementation of the Rivest algorithm is far lower than Hellman's. We have minimized table space by not storing the chain length.

We have also tested the strategy of not breaking the generation of intermediary keys when we reach an endpoint. Even if the alarm rate is higher, the success rate varies negligibly around the same values.

Results for Rivest - d = 7 d = 11

Success rate (nr. of hits per 1000 tests)

d=11

tm	**262144**	**524288**	**1048576**	**2097152**
512	5	10	11	12
1024	8	8	12	-
2048	9	8	-	-

d=7

tm	**262144**	**524288**	**1048576**	**2097152**
512	4	6	14	27
1024	1	10	12	27
2048	8	10	11	25

Similar results, except for t=512 and m=2097152.

Results for Rivest - d = 7 d = 11

Alarm rate

d=11

tm	**262144**	**524288**	**1048576**	**2097152**
512	4	9	22	43
1024	18	44	75	-
2048	71	162	-	-

d=7

tm	**262144**	**524288**	**1048576**	**2097152**
512	3	6	11	23
1024	7	13	28	54
2048	15	30	60	117

Noticeable better alarm rate for d=7.

Choice of d

Choosing d must take into account the value of t since 2^d is supposed to approximate the average chain length. We have chosen d=7 and d=11 because of the choices of t= $2^8, 2^9, 2^{10}$ The alarm rate is higher for d = 7, but the succes rate is better for the $512x2^{21}$ table.

Hellman versus Rivest

Success rate Hellman (nr. of hits per 1000 tests)

tm	**524288**	**1048576**	**2097152**
512	17	23	32
1024	13	20	30
2048	15	24	34

Success rate Rivest d = 7 (nr. of hits per 1000 tests)

tm	**524288**	**1048576**	**2097152**
512	6	14	27
1024	10	12	27
2048	10	11	25

Lower success rate registered for Rivest.

Hellman versus Rivest

Alarm rate Hellman

tm	**524288**	**1048576**	**2097152**
512	16	31	66
1024	62	124	254
2048	257	524	1032

Alarm rate Rivest d = 7

tm	**524288**	**1048576**	**2097152**
512	6	11	23
1024	13	28	54
2048	30	60	117

Significantly lower alarm rate registered for Rivest.

References

Ali Doganaksoy, Cagdas Calik, Fatih Sulak

Observations on Hellmans Cryptanalytic Time-Memory Trade-off

Jin Hong, Kyung Chul Jeong, Eun Young Kwon, In-Sok Lee, and Daegun Ma

Variants of the Distinguished Point Method for Cryptanalytic Time Memory Trade-offs

Questions