CRYPTANALYSIS of PRESENT via CUDA DEVICES

Cihangir Tezcan[1,2] **Alptekin Temizel**[3]

cihangir@metu.edu.tr atemizel@metu.edu.tr

[1]Institute of Applied Mathematics, Department of Cryptography, Middle East Technical University, Ankara, Turkey

[2]Department of Mathematics, Middle East Technical University, Ankara, Turkey

[3]Graduate School of Informatics, Middle East Technical University, Ankara, Turkey

Abstract

The security of block ciphers depends on the design and the length of the key that is used for encryption and decryption processes. For a k-bit key, exhaustively checking every possible key to find the correct one requires 2^k encryptions. Since $k \geq 80$ for modern ciphers, 2^k encryptions required for the exhaustive search is beyond our computational power. However, finding weaknesses in the cipher can provide a better attack with time complexity less than 2^k. Yet we need to experimentally verify this theoretically obtained statistical observations.

For the block cipher PRESENT, in this project we show that GPUs can provide more than 9.1x speed up for verifying theoretically obtained statistical weaknesses of ciphers. Moreover, we show that an agency having 2^{20} Tesla k20s can capture 80-bit keys that consist of 95 printable ASCII characters in a day on average and an agency with a budget of 250 billion dollars can capture any 80-bit PRESENT key in a month by performing exhaustive search using Geforce GPUs.

PROBLEMS

Problem 1: A differential path $\alpha \rightarrow \beta$ with probability p_1 is a statistical weakness saying that two plaintexts P_1 and P_2 with P_1 XOR $P_2 = \alpha$ provides the ciphertexts C_1 and C_2 with probability p_1 where C_1 XOR $C_2 = \beta$. To verify the correctness of such a theoretical observation, we need to perform experiments on many plaintexts, whose difference is α, by encrypting them and checking whether the corresponding ciphertext difference is β. In [1], Tezcan provided a 13-round attack on PRESENT that uses an improbable differential of probability $p_1 = 2^{-13.002}$ and in [2] Blondeau, due to other experiments on the cipher, claimed that this probability should be higher and thus the attack should not wok. In order to verify the correctness of the attack, we need to distinguish this differential from a random permutation which has probability $p = 2^{-13}$. Thus, we need to experiment using $2^{36.86}$ plaintext pairs for 100 different keys, which requires $2^{44.5}$ 8-round PRESENT encryptions and that would take days on a CPU.

Problem 2: Lightweight block cipher PRESENT is designed to be used with either 80-bit keys or 128-bit keys. Exhaustive search on 80-bit keyed PRESENT requires one plaintext P and corresponding ciphertext C. Hence the attacker encrypts P with every possible 80-bit key and stops when a key provides the ciphertext C.

Implementation Details

In this work, PRESENT [3] block cipher, which is an ISO/IEC standard for lightweight cryptography [4], is implemented to perform exhaustive key search and differential path verification.

First Approach

1) Key Schedule: Problem 1 consists of just 1 64-bit XOR operation as the round keys are fixed. Problem 2 requires computation of round keys at every round – calculated 8 shift, 6 XOR, 5 AND and 1 4-bit table look-up operations.
2) Substitution: Basic approach is to create an array of size 16 which maps the 4-bit input to a 4-bit output. However, since we are programming on the software level, the smallest variable we can create is of 8 bits, not 1. Thus, we combined every consecutive two S-boxes and created an array of size 256 which maps 8-bits to 8-bits. This change significantly improved the performance.
3) Permutation: Naive approach is to permute bits one by one. This would mean 64 shift, 64 AND and 64 XOR operations. However, we observed that performing one shift operation actually permutes 2 or 3 bits to corect places instead of one. This way, we observed that this step can be performed by performing 31 shift, 31 AND and 31 XOR operations, instead of 64.

Second Approach

1) This step is the same as the first approach.
2) In Step 3, we perform too many operations just to permute the bits. Since we know the permutation, we can apply it to the outputs of the S-box beforehand, which would provide 64-bit values. Thus, XORing the outputs of the S-boxes provides us the state of the output of the round function. This way we reduce the Steps 2 and 3 to 7 shifts, 8 ANDs, 7 XORs and 8 64-bit table look-ups. Note that for the table look-ups, now we use 256*64=16384 bits (16 KB) shared memory for these steps (And Step 1 requires 16*8=128 bits of shared memory).

Comparison

First approach uses a very small amount of shared memory which prevents us from occupancy problems, especially when using Fermi devices and the serialization is around 1.98%. The second approach uses a little more than 16 KB shared memory which can cause occupancy problems and the serialization is around 19.98%. However, number of operations are significantly reduced in the second approach and it performs a lot faster than the first approach in both GPU and CPU implementations. For this reason, we used the second approach in our final implementations.

Geforce GT 540M	Million encryptions per second
First Approach	15.33
Second Approach	27.92

The default shared memory bank size in Kepler devices are 32 bits but it can be configured to 64 bits.

Tesla k20 Shared memory bank size	Million encryptions per second
32 bits	279.50
64 bits	340.93

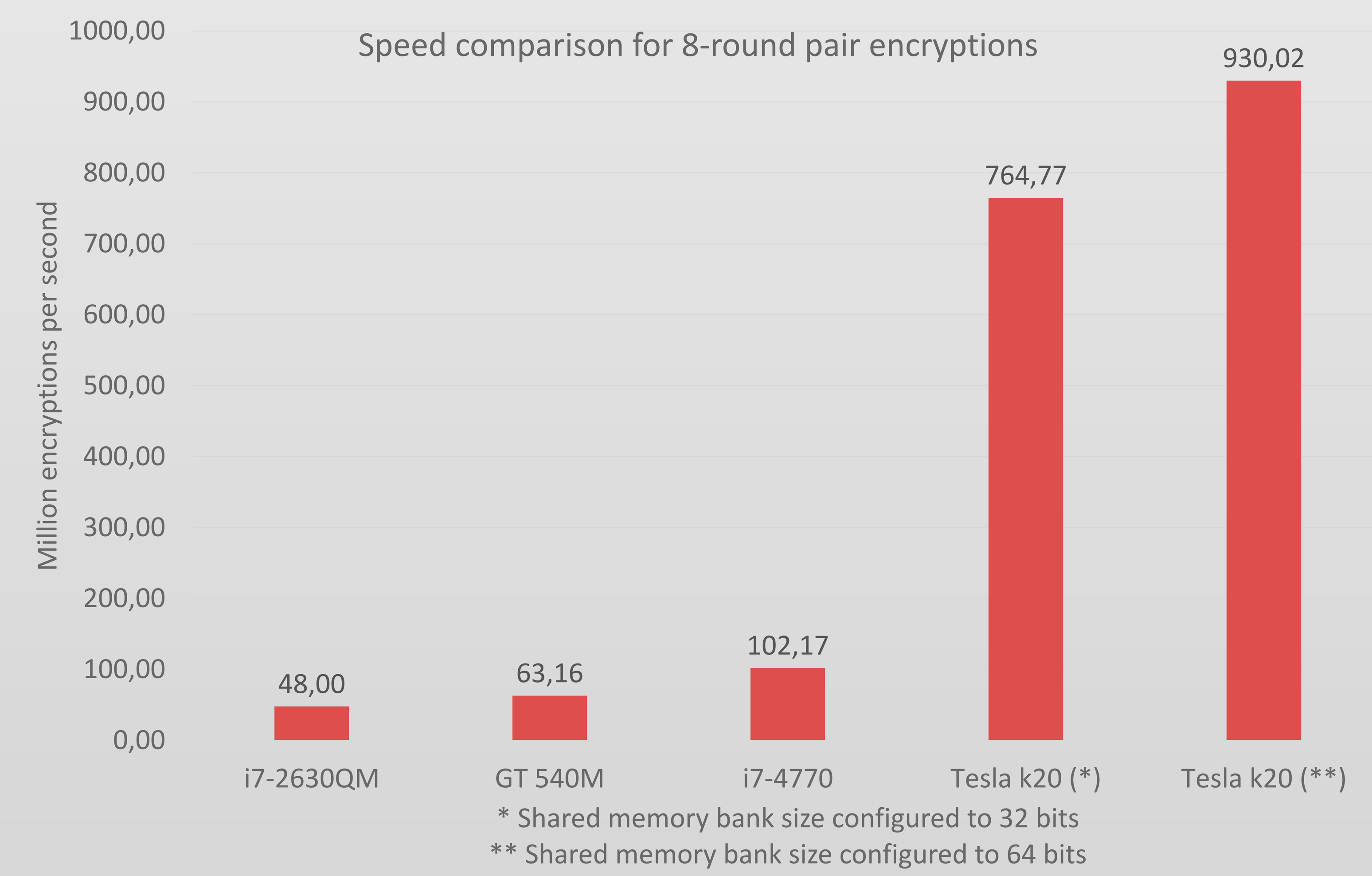
Conclusions

- GPUs can provide more than 9.1x speed up when verifying theoretically obtained statistical weaknesses of PRESENT and more than 10.9x speed up for exhaustive key search.
- For Problem 1, it took less than 7 and a half hours to perform $2^{44.5}$ reduced round encryptions on a single Tesla k20 and we confirm that the attacks of [1] are correct.
- If the 80-bit key of PRESENT is generated from the 95 printable characters of ASCII table, the key space reduces to $2^{65.7}$ from 2^{80}. Thus, an organization having 2^{20} Tesla k20s can capture such a key in a day on average.
- An agency with a budget of 250 billion dollars can capture any 80-bit PRESENT key in a month on average by performing exhaustive search using Geforce GPUs (e.g. GTX 780).

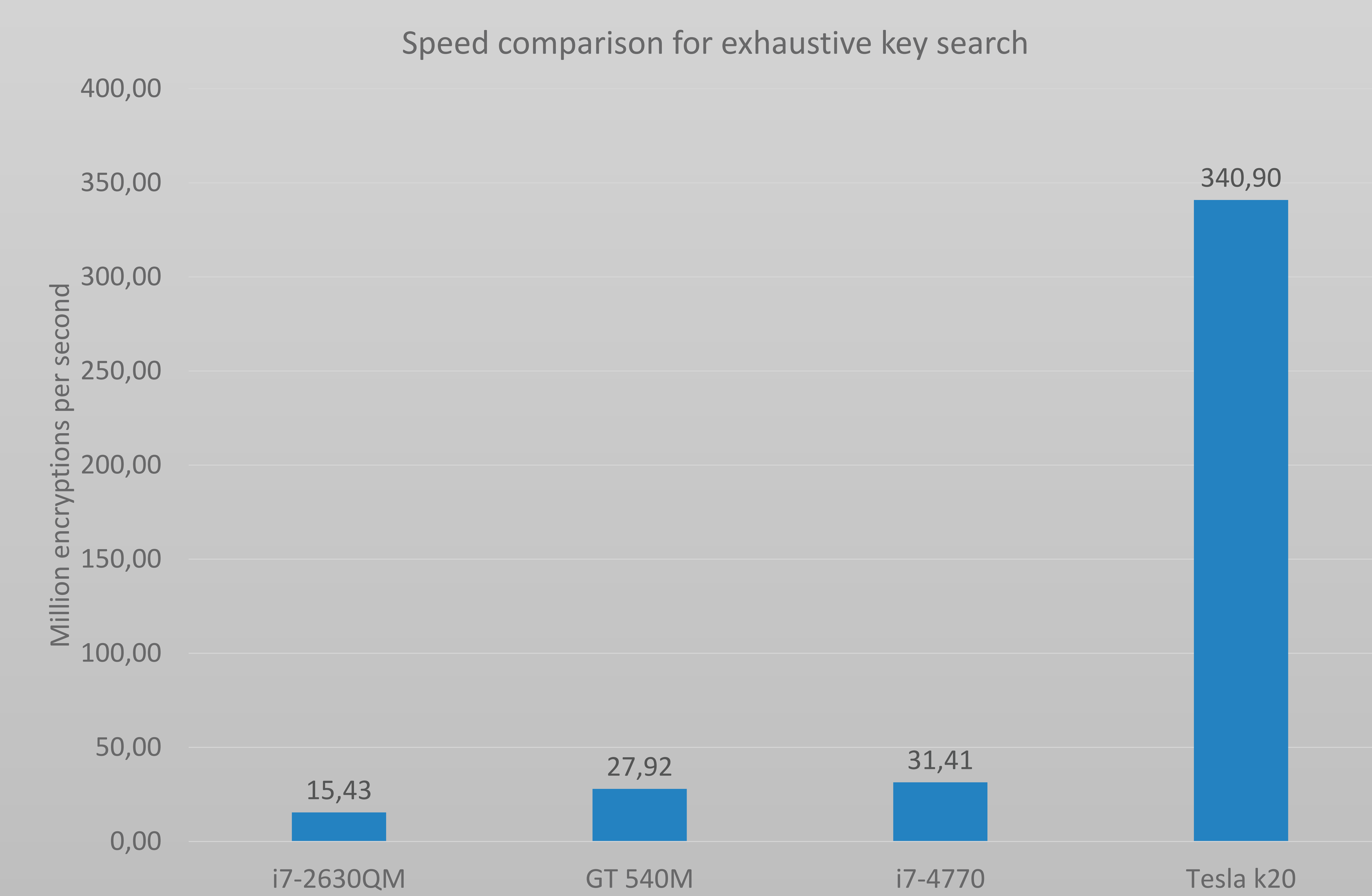
Results of Used Algorithms

Experiments are performed on a notebook having Intel Core i7 2630QM CPU, 8 GB usable RAM with Geforce GT 540M GPU and a PC having Intel Core i7 4770 CPU, 8 GB usable RAM with Tesla k20c GPU. We used all 4 cores of the CPUs in the experiments (actually we used all 8 threads of the CPUs).

CPU and GPU Performance Comparison for Problem 1



CPU and GPU Performance Comparison for Problem 2



PRESENT Cipher

Present is a 31-round Substitution Permutation Network type block cipher with block size of 64 bits that supports 80 and 128-bit secret key. The round function of PRESENT, which is depicted in Fig. 1, is the same for both versions of PRESENT and consists of 3 steps:

1. XORing the state with the 64-bit round key
2. dividing the state into 16 nibbles and perform four bit substitutions using the S-box
3. Permutation of 64 bits of the state

Key Schedule: 80-bit secret key is stored in a key register K and represented as $k_{79}k_{78} \ldots k_0$. The round keys K_i ($0 \leq i \leq 31$) consist of 64 leftmost bits of the actual content of register K. After round key K_i is extracted, the key register K is rotated by 61 bit positions to the left, then S-box is applied to the left-most four bits of the key register and finally the round counter value, which is constant, is XORed with bits $k_{19}k_{18}k_{17}k_{16}k_{15}$.

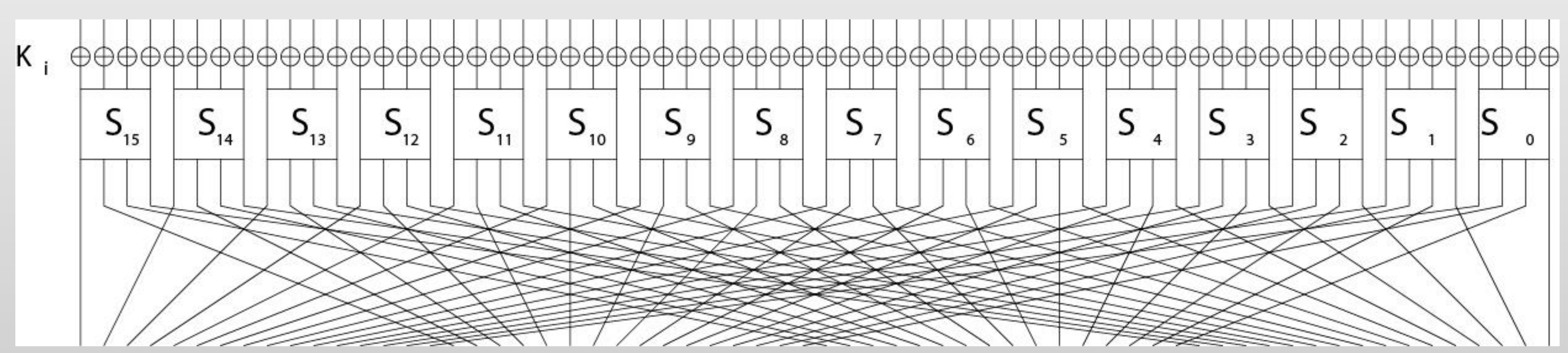


Fig. 1: i-th round of PRESENT

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Fig. 2: 4 x 4 S-box of PRESENT

References

[1] Tezcan, C.: Improbable differential attacks on PRESENT using undisturbed bits. Journal of Computational and Applied Mathematics 259, Part B(0) (2014) 503-511 Recent Advances in Applied and Computational Mathematics: ICACM-IAMMETU. On the occasion of 10th anniversary of the foundation of Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

[2] Blondeau, C.: Improbable differential from impossible differential: On the validity of the model. In Paul, G., Vaudenay, S., eds.: INDOCRYPT. Volume 8250 of Lecture Notes in Computer Science., Springer (2013) 149-160.

[3] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: An ultra-lightweight block cipher, in: P. Paillier, I. Verbauwhede (Eds.), CHES, in: Lecture Notes in Computer Science, vol. 4727, Springer, 2007, pp. 450–466.

[4] ISO/IEC 29192-2:2012, Information technology — security techniques — lightweight cryptography — part 2: block ciphers, 2011.