Acceleration of Lightweight Block Ciphers on Microprocessors

Etienne Tehrani, Tarik Graba, Jean-Luc Danger

June 2019

Cryptography is a key element to the development of secure communication in embedded environment such as within or between connected cars. In such constrained devices standard cryptographic algorithms have been considered too costly which lead to the emergence of specific Lightweight Block Ciphers (LBC). The lack of standards alongside industry's desire to use uniquely tweaked LBC calls for a generic and efficient implementation of those algorithms. Microprocessors are a part of most of these embedded systems which allows them to implement any of these algorithms but not efficiently way as it lacks specific instructions. For instance, the RiscV is an open source ISA which can be used in these microprocessors and is currently being enhanced by research through extensions. In this work we propose the study of this ISA and the development of an extension for efficient implementation of LBC.

From the state of the art [5], [7] we have selected some LBC based on the following criteria: at least a 128-bit key for security and a 64-bit block size to limit the necessary resources. In order to identify useful extensions, we first identified which parts of LBC are slow when implemented in pure software, and how common they are in state of the art LBC. We only studied the datapath of the cipher as we considered the key scheduling to be part of preprocessing. We used a software implementation of each of the studied algorithms to isolate the costly parts of the ciphers. The computation time was evaluated in number of RiscV assembly language instructions.

Studied LBC algorithms exhibit 3 main computation steps:

- The key addition which is a simple XOR and doesn't require additional instructions
- 8 or 16 4x4 Sbox (common for LBC) which can be implemented as LUT and can be accelerated thanks to the addition of a specific (SIMD) LUT instruction
- The diffusion is generally not trivial to implement in pure software and as it can be quite different from one algorithm to the other it is not obvious to provide a unique extension to implement it.

To propose an efficient extension for the diffusion, we propose a classification of the studied LBC. Each category can be implemented with a dedicated instruction which could compute the entire diffusion in a single processor cycle. 4 categories are proposed:

- bit level diffusion: where a generic hardware implementation has a huge cost, but specific instruction, dedicated to a couple of LBC can be added. For instance, PRESENT [4] which is a ISO standard and its evolution, GIFT [2], have similar, but still different permutations. While a specific instruction must be added for each of them, the hardware cost of such an instruction is close to none as it is the reorganisation of wires.
- bit level rotation: plenty of algorithms, especially Feistel-type (such as GOST [8], Simeck [11], Rectangle [12]) use a simple rotation of the state for the diffusion. While common in other processor families, RiscV does not provide a rotation instruction and therefore adding this rotation instruction will reduce from 3 to 1 instruction each round for those ciphers.
- nibble level diffusion: such a diffusion affects only the state at a nibble (4-bit) level, meaning that bits within a nibble cannot be swapped. This category will be implemented as a matrix multiplication between a 16x16 cipher-specific matrix and a 16 nibble state. Such a matrix requires 256 bits of information in order to be as generic as possible. Indeed, reducing the amount of parameters also reduces the amount of potential algorithms which can be considered for this generic implementation. Such an instruction would account for both a shift row type and a mix column type of diffusion. LBC of this category are Twine [10], Midori [1] and Skinny [3], the latter which is part of the NIST Lightweight Cryptography Standardisation candidates.
- nibble level diffusion, with Galois Field: Some algorithms use a matrix multiplication based on Galois Field and therefore require their specific instruction to account for the fact that the matrix is not composed of only 0s and 1s but with also 2s and 3s. Algorithms such as LED [6] or Piccolo [9] belong to this category.

References

[1] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 411–436. Springer, 2014.

[2] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. Gift: a small present. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 321–345. Springer, 2017.

[3] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The skinny family of block ciphers and its low-latency variant mantis. In *Annual Cryptology Conference*, pages 123–153. Springer, 2016.

[4] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 450–466. Springer, 2007.

[5] CryptoLux. Lightweight block ciphers. `https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers`. 2019-04-03.

[6] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'11, pages 326–341, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] NIST. Lightweight cryptography standardization: Nist announces round 1 candidates. `https://csrc.nist.gov/projects/lightweight-cryptography/round-1-candidates`. 2019-04-18.

[8] Axel Poschmann, San Ling, and Huaxiong Wang. 256 bit standardized crypto for 650 ge–gost revisited. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 219–233. Springer, 2010.

[9] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: an ultra-lightweight blockcipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 342–357. Springer, 2011.

[10] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. Twine: A lightweight block cipher for multiple platforms. In *Selected Areas in Cryptography*, page 339. Springer.

[11] Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D Aagaard, and Guang Gong. The simeck family of lightweight block ciphers. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 307–329. Springer, 2015.

[12] Wentao Zhang, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, 2015.