Towards Designing a Greener Advanced Encryption Standard (AES)

S. Raghu Talluri
School of Computing
University of North Florida
Jacksonville, Florida 32224
Email: n00926109@ospreys.unf.edu

Swapnoneel Roy
School of Computing
University of North Florida
Jacksonville, Florida 32224
Email: s.roy@unf.edu

***Abstract*—In this work we study the energy consumption by Advanced Encryption Standard (AES), a symmetric key encryption protocol from the *algorithmic* perspective. Our work is motivated by the frequent use of AES as a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.**

We use a generic energy complexity model designed by Roy et. al. to analyze the energy consumed by AES. We then show how to reduce the energy consumption by AES by performing the processing of the blocks of AES encryption (of size 16 bytes) *in parallel*.

I. Introduction

Motivation to consider energy efficiency in delivering information technology solutions comes from: 1. Data centers with strong focus on energy management for server class systems. 2. Personal computing devices such as smartphones, handhelds, and notebooks, which run on batteries and perform a significant amount of computation and data transfer. 3. Telecom providers expecting to invest in equipment that will form an integral part of the global network infrastructure. On the other hand, information security has become a natural component of all technology solutions. Security protocols consuming additional energy are often incorporated in these solutions. Thus, the impact of security protocols on energy consumption needs to be studied. Ongoing research in this context has been mainly focused on energy efficiency/consumption on specific hardware and/or different systems/platforms. Very little is known or has been explored regarding energy consumption or efficiency from an *applications* perspective, although apps for smartphones and handhelds abound.

Our Contributions: Our work makes two key contributions. Since energy or power has become a first class component in computing now a days, this could be very expensive especially for the battery driven devices like laptops and PDAs. As a conclusion to this observation, we found a lot of work done to reduce energy consumption in network communication and security protocols in the hardware, virtual machines, operating systems, and the system software levels [1], [2], [3], [4], [5]. But not much work has been done from the application or algorithmic perspective to minimize energy or power consumption in such protocols. In other words, the problem which we try to investigate is can we design energy aware security protocols? Or can we modify existing protocols to make them energy optimal, without compromising on the level of security they provide?

We next analyze the energy consumption by AES. Specifically we estimate the energy consumed by the AES algorithm using the energy model of [6]. Finally we modify the AES algorithm to lower the level of energy consumption pertaining to the energy model. Specifically we *parallelize* the input by accessing blocks of size 16 bytes in parallel for AES and observe it lowers the energy consumption of the protocol.

AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael which has a fixed block size of 128 bits (or 16 bytes), and a key size of 128, 192, or 256 bits. By contrast, the Rijndael specification per se is specified with block and key sizes that may be any multiple of 32 bits, both with a minimum of 128 and a maximum of 256 bits.

The rest of the paper is structured in the following manner. Section II describes the AES algorithm in detail. The energy complexity model we use, and the techniques we use to optimize energy consumption by AES is described in Section III. Section IV describes our experimental set and presents our key experimental results on energy consumption of AES. Finally, Section V summarizes the results and discusses future research directions.

II. Advanced Encryption Standard (AES)

The AES algorithm on each block of 16 bytes can be described as follows:

1) **KeyExpansion.** Round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
2) **InitialRound.**
   a) `AddRoundKey`: Each byte of the state is combined with a block of the round key using bitwise xor.
3) **Rounds**
   a) `SubBytes`: A non-linear substitution step where each byte is replaced with another according to a lookup table.
   b) `ShiftRows`: A transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

Data: A Plaintext block of 16 bytes `byte` $in[4 * Nb]$
Result: A Ciphertext block of 16 bytes `byte` $out[4 * Nb]$
`word` $w[Nb * (Nr + 1)]$;
$state = in$;
$AddRoundKey(state, w[0, Nb - 1])$;
for *round = 1 step 1 to Nr1* **do**
  $SubBytes(state)$;
  $ShiftRows(state)$;
  $MixColumns(state)$;
  $AddRoundKey(state, w[round * Nb, (round + 1) * Nb - 1])$;
end
$SubBytes(state)$;
$ShiftRows(state)$;
$AddRoundKey(state, w[Nr * Nb, (Nr + 1) * Nb - 1])$;
$out = state$;

Algorithm 1: The AES Algorithm

c) `MixColumns`: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
d) `AddRoundKey`

4) **Final Round (no MixColumns).**
a) `SubBytes`
b) `ShiftRows`
c) `AddRoundKey`

AES is a symmetric key encryption cipher. That is the same set of keys are used for both encryption and decryption. For a detailed description of each of the operations, please see [7], [8], [9], [10].

III. Engineering AES for Energy Efficiency

A. Energy Complexity Model

An asymptotic energy complexity model for algorithms was proposed in [6]. Inspired by the popular DDR3 architecture, the model assumes that the memory is divided into P *banks* each of which can store multiple blocks of size B. In particular, P blocks in P *different* memory banks can be accessed in parallel (Figure 1). The main contribution of the model in [6] was to highlight the effect of parallelizability of the memory accesses in energy consumption. In particular, the energy consumption of an algorithm was derived as the weighted sum $T + (PB) \cdot I$, where T is the total time taken and I is the number of *parallel* I/Os made by the algorithm.

B. P-way Parallelism for AES input

Energy optimal algorithms proposed in [6] require data to be laid out in memory with a controlled degree of parallelism. We first propose a way to ensure desired memory parallelism for a given input M to the AES algorithm. We ensure memory parallelism for the processing of 16-byte blocks by the AES algorithm.

We treat the AES algorithm as a black box. Given an input M, AES divides M into blocks of 16 bytes and processes each block for encryption to produce the ciphertext (Figure 2).

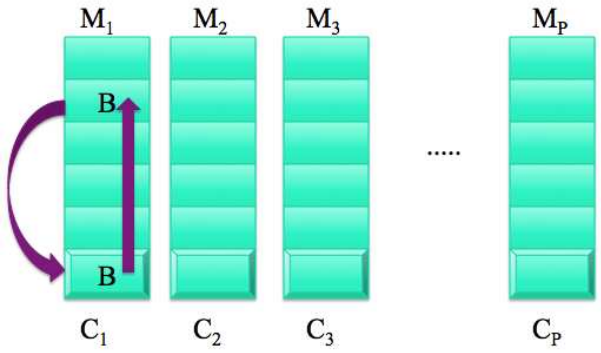

Figure 1. Memory divided in banks.

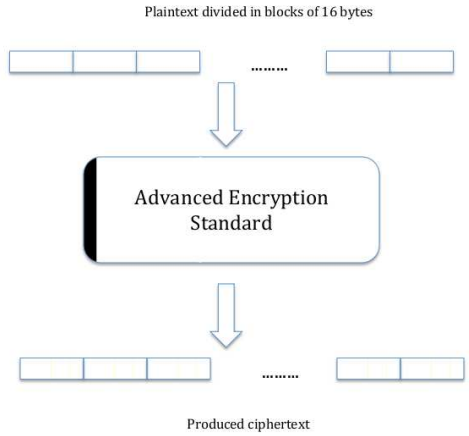

Figure 2. AES input in blocks of size 16 bytes.

For the message M which is a multiple of blocks of 16 bytes, we created a logical mapping which ensures access to the blocks in P-way parallel fashion, where P ranges from 1 to 8. More specifically, when $P = 1$, (almost) all the blocks of 16 bytes are clustered in a single bank. While for $P = 8$, the blocks are evenly spread across all 8 banks to ensure the maximum degree of parallelism of the access to the input (M) to AES. We also experiment for $P = 2$, and $P = 4$.

To achieve the above, we create a mapping function which maps the physical input M into the logical input which defines the degree of parallelism. In other words, we define an *ordering* among the blocks of 16 bytes which defines the logical input (and the degree of parallelism).

IV. Experimental Results

We next evaluate the energy consumed by AES algorithm for $\{1, 2, 4, \text{and } 8\}$-way parallel data. Again, for a recap, a k-way parallelism in the input data for AES suggests that the 16 bytes blocks in the input to AES algorithm are spread across k banks in the memory. So a 8-way parallelism in an 8 bank memory means full parallelism (optimal case), and a 1-way parallelism is the worst case. According to the energy complexity model of [6], an 8-way parallelism should account for lower energy consumption in AES.

The experiments were performed on a PC machine. The machine has an Intel i5-3427U processor with inbuilt graphics

with 4-GB ram running Windows 7 SP1. The machine was run on battery during the experiments.

We measured the power drawn by an application using the Joulemeter tool [11] developed by Microsoft Research. Joulemeter runs on Windows XP and Windows 7 currently. It is a software tool that estimates the power consumption of your computer. This software gives the user the ability to tag a CPU process and measure the power consumed by the application in real-time. The data over a period of time is logged and plotted to give us a visualization of the power and energy requirements of the software. It also tracks computer resources, such as CPU utilization and screen brightness, and estimates power usage.

We calculate energy by the product of the time to execute and the (average) power consumed during the time of execution. All the experiments were repeated a hundred times, and the mean value has been reported. The benchmark code was written in C and was compiled using `gcc`. We note that DDR3 has 8 banks.

The first results reported measures the energy consumed by AES with input sizes of 8MB, 16MB, 32MB, and 64MB. These numbers have been obtained for the best case (8-way parallelism). A key size of 128 bits was used for all the experiments.

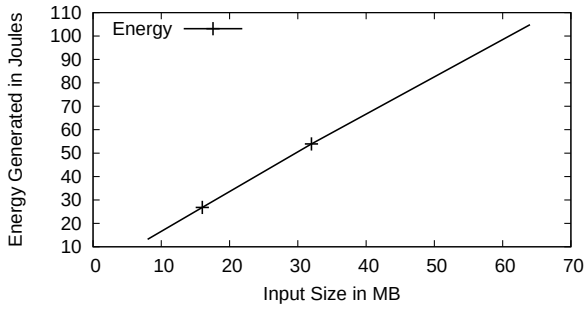

Figure 3. Energy consumption in joules by AES with various sizes of inputs.

We do not see any surprises in Figure 3. The energy consumption for AES varies linearly with respect to the input size.

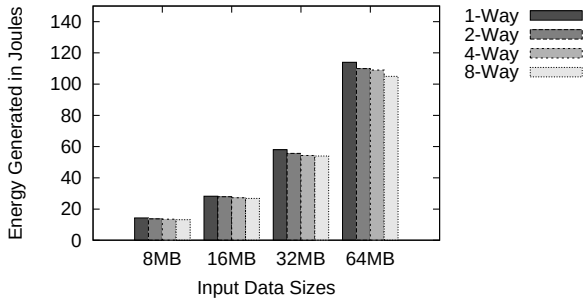

Figure 4. Energy consumption in joules by AES with various degrees of parallelism.

Figure 4 shows us interesting numbers. We measure the energy consumed by AES over fixed size input,varying the degree of parallelism of the input. We see change in energy consumption based in the degree of parallelism of the input. Higher degree of parallelism lowers the energy consumption. We note the difference is not very high between the best and worst cases, and that is partly due to other factors like the code overhead, noise due to other processes, etc. The numbers indicate the applicability of the energy model of [6] on the AES algorithm. It signifies that the energy consumption of AES can be lowered by parallelizing the input data.

V. Conclusion

In this paper, we have made two key contributions. We first call for designing algorithmic techniques to bring down the energy consumption of security protocols which build them. We next experiment on the applicability of the generic energy complexity model [6] on AES algorithm. We observe the model to be applicable to AES. Our numbers show a reduction in the energy consumption of AES by increasing the degree of parallelism in the input.

It would be interesting to compute the energy consumption for other security protocols like RSA (public key), or the advanced hash functions like MD4, and MD5. We conjecture the applicability of our techniques to lower the level of energy consumption in them.

References

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. IEEE, 2000, pp. 10–pp.

[2] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace conference proceedings, 2002. IEEE*, vol. 3. IEEE, 2002, pp. 3–1125.

[3] M. Handy, M. Haase, and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*. IEEE, 2002, pp. 368–372.

[4] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1567–1576.

[5] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 40–50, 2002.

[6] S. Roy, A. Rudra, and A. Verma, "An energy complexity model for algorithms," in *ITCS 2013*.

[7] "Announcing the advanced encryption standard (aes)," .

[8] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer, 2002.

[9] E. Conrad, "Advanced encryption standard," *White Paper*, 1997.

[10] D. Selent, "Advanced encryption standard," *Rivier Academic Journal*, vol. 6, no. 2, 2010.

[11] "Joulemeter: Computational energy measurement and optimization," http://research.microsoft.com/en-us/projects/joulemeter/.