# A Case for Energy-Aware Security Mechanisms

Xun Li          Frederic T. Chong
*Department of Computer Science*
*University of California, Santa Barbara*
*Goleta, CA 93117, USA*
{*xun, chong*}*@cs.ucsb.edu*

***Abstract*—The coming age of “big data” has been a recent focus of IT companies, governments, and even the World Economic Forum (WEF). The rapid, exponential growth of global data outpaces Moore’s law improvements in computing and storage efficiency, creating a growing “energy wall” in coping with global data. We observe in this paper that this problem is significantly exacerbated by (needed) security mechanisms. We propose for more systematically designed energy-aware security mechanisms.**

***Keywords*-security; energy efficiency**

## I. Introduction

Transistor scaling, as described by Moore’s Law, has continued exponentially for several decades. Dennard Scaling [9], which relies upon lowering operating voltage, has helped keep chips within practical power envelopes as transistor counts and operating frequencies have increased. Unfortunately, Dennard Scaling has ended and Moore’s Law is slowing down.

On the other hand we have also seen an exponential growth in data triggered by the increasing popularity of user-generated content and crowd-sourcing applications. This gigantic amount of data has started to go beyond our capability of storing and processing, leading to a vast inefficiency gap between silicon technology and the big data. Figure 1 shows the data warehouse growth rate compared to the silicon growth rate guided by Moores Law, estimated in 2003 and projected to today. The current worlds largest data warehouse is about 3 petabytes from IBM DB2. We can see that there has been a fast increasing gap between the two, leading to significant inefficiency and demand to energy. If, as the World Economic Forum proposes [1], leveraging this data is key to economic growth in this century, then we face some severe technical challenges.

The energy wall, however becomes even worse when security becomes another important design metric other than performance. Not only do we have all kinds of personal devices that all require various levels of security, but we also have large amounts of private data in the cloud which needs to be protected. Todays security approaches are mostly ad-hoc: each computing device is protected with some hardware security features as well as software anti-virus tools and firewall. Network traffic are secured by various security protocols and spam filtering techniques. As the global internet data volume keeps increasing as shown in Figure 1, the overhead in security will grow as well and soon becomes a nontrivial portion of the total energy budget. These security mechanisms cannot be designed independently in an ad-hoc manner any more and energy-aware systematic security will play an important role in system design.

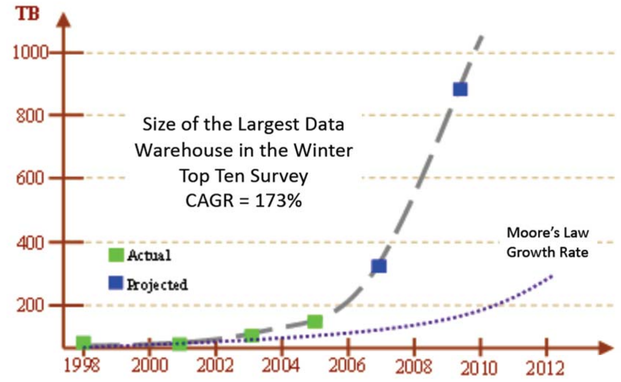


Figure 1. Size of largest data warehouse growing trend from Winter Top Ten Survey 2003 [20], compared to Moore’s Law. Such large and still increasing gap indicates a serious energy inefficiency problem in dealing with the large volume of data.

Our end goal of this work is to provide an understanding of how security mechanisms at different levels interact with energy from the perspective of internet data growth, so as to emphasize the importance of systematic energy-aware security. As a first step, in this short paper we survey the energy cost of several major security mechanisms applied in both individual devices and the cloud. We estimate their impact to the energy budget today and also predict the future trend. We hope this work serves as a complementary analysis to existing work towards green security [16], [5], [6], as well as a reference for future study on energy-aware security.

In the rest of the paper, we survey the following security mechanisms that contribute to the majority of energy overhead in security: anti-virus software (mostly PCs, more mobile devices in the future), spam filtering (on the cloud), hardware primitives (all kinds of devices), network security (mostly protocol overhead, spent on every bit of data). In order to compare them with computation energy, we try to normalize them into energy consumption per byte of data.

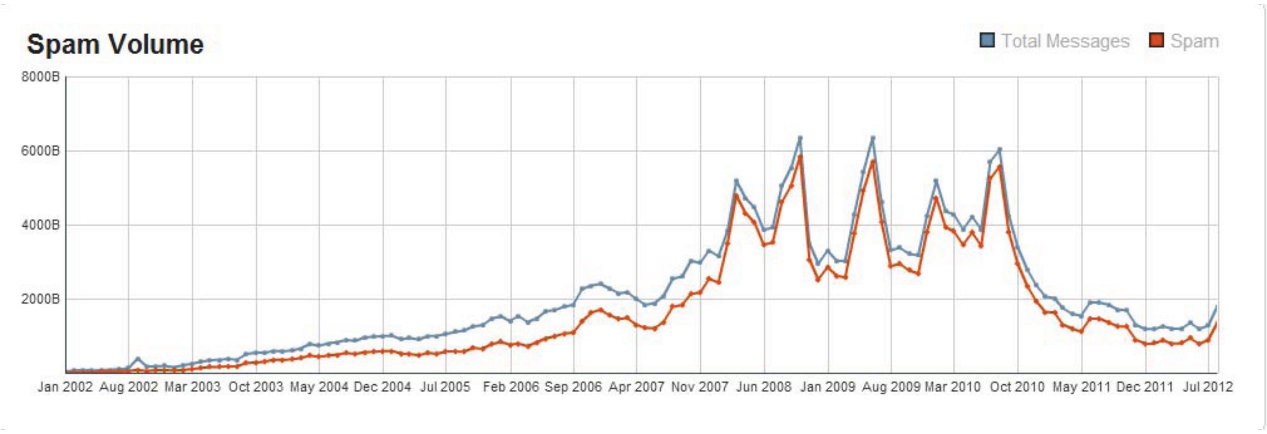


Figure 2. Global Spam Email Volume since 2002 [18]. It has gone through a steady increase since the beginning and a quick jump around 2007. We have also seen a big drop since 2010 after several major internet bots taken-down.

## II. Energy Cost of Security

### A. Spam Filtering

About 90-95% of all emails in the web are spam emails. Those spam emails can cause significant problem for both individual users and businesses if not filtered properly. Due to the large volume, spam filtering can consume a large amount of energy itself. Many different parts in the internet are involved in spam filtering collaboratively, from filtering spam emails at ISPs to more advanced learning schemes at mail servers such as Gmail.

Based on a report from McAfee [15], about 5,542 million kWh energy is spent annually (2008) on spam filtering. The amount of spam emails sent on 2008 is about 62 trillion. The energy consumption per spam email filtering can be estimated as the following:

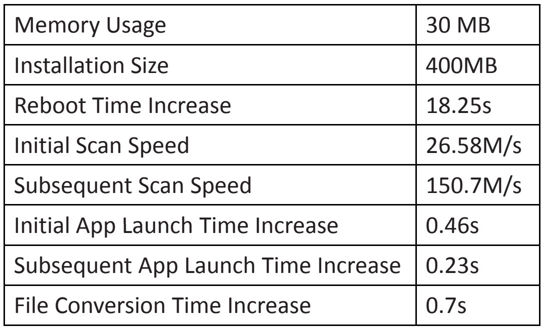
$$\begin{aligned} Energy\ Per\ Spam &= Total\ Energy \div Spam\ Count \\ &= 5542 \times 10^{6} kWh \div (62 \times 10^{12}) \\ &= 8.94 \times 10^{-5} kWh \\ &= 321.84\ Joule \end{aligned}$$

Roughly $321.84 Joule$ energy is required to filter a single spam email. Figure 2 shows the amount of global spam email volume in the past 8 years. Although the spam email growth speed seems to be much slower than the global internet data growth rate shown later in Figure 6, a large portion of the spams in fact have shifted into different forms. The amount of spam users and accounts now have shown up on various social network platforms such as Facebook and Twitter, which becomes a much more effective method compared to traditional email spam, and they are much harder to detect and filter as well [12], [11]. Hence the total amount of effort required for spam filtering will still grow proportionally to the global data volume in general. Given the amount of global data volume as 2.55 zettabytes in 2012, the energy per byte (EPB) for spam filtering can be estimated as:

$$\begin{aligned} EPB_{spam} &= Energy\ Per\ Spam \times Spam\ Count \\ &\div Data\ Volume \\ &= 321.84 \times (11 \times 10^{12}) \div (2.55 \times 10^{21}) \\ &= 1.39 \times 10^{-6} \end{aligned}$$

### B. Anti-virus

Anti-virus software has been heavily used in desktop/laptop PCs for many years. It has been well known that anti-virus software are using significant computing resource as well as consuming large amount of energy. Although it is difficult to precisely determine the carbon footprint impact from anti-virus software, it can still be estimated based on system performance impact. Anti-virus software has major impacts to system boot time, application launch time and file I/O operations. Many independent organizations are running performance tests on major commercial anti-virus software yearly. Table I shows the summarized results from AntiVirus Ware Report [3] on 15 anti-virus software. The system performance is affected in many different ways, all leading to extra energy consumption. The major overhead lies in I/O operations, that antivirus software is involved in all file accesses through scanning and signature checks. As a representative, the energy overhead can be estimated based on the ”File Conversion Time Increase” in Table I. We estimate the energy spent in the extra time during file conversion by estimating the unit energy for typical computation. The experiment in the paper was performed on typical Mp3 files with an average size of 4MB. Processor frequency is 2GHz Dual-Core (4GHz maximum computation capability). Assuming an average CPI of 1.0 and energy per instruction (EPI) of 11nJ [10], the energy overhead per byte can be



| | |
|---|---|
| Memory Usage | 30 MB |
| Installation Size | 400MB |
| Reboot Time Increase | 18.25s |
| Initial Scan Speed | 26.58M/s |
| Subsequent Scan Speed | 150.7M/s |
| Initial App Launch Time Increase | 0.46s |
| Subsequent App Launch Time Increase | 0.23s |
| File Conversion Time Increase | 0.7s |

Table I
AVERAGE PERFORMANCE IMPACT OF 15 ANTI-VIRUS SOFTWARE, SUMMARIZED FROM [3].

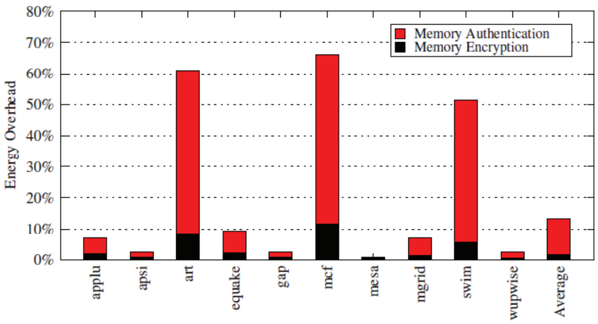


Figure 4. Energy overhead in adding extra security features to the memory [8]. Memory Authentication protects integrity, while Memory Encryption protects secrecy.

derived as:

$$
\begin{aligned}
EPB_{AV} &= Time \div Size \times Frequency \div CPI \times EPI \\
&= 0.7 \div (4 \times 10^6) \times (4 \times 10^9) \div 1 \times (11 \times 10^{-9}) \\
&= 7.7 \times 10^{-6}
\end{aligned}
$$

An important trend in personal computing is the pervasive availability of mobile devices. As the complexity of mobile operating system becomes more complex, anti-virus software as well as software firewall will start to become an increasing demand for mobile users. A recent study [4] on the energy consumption of mobile anti-virus software shows that they can contribute up to 40% of overall energy under ping floods (1KB payload, 100 times per second), shown in Figure 3. In the "Normal - UP" case when users are actively using the device with wifi on, the energy per byte consumed by mobile antivirus tools can be estimated as:

$$
\begin{aligned}
EPB_{MobileAV} &= Power \div (Payload \times Frequency) \\
&= 0.461 \div (1024 \times 100) \\
&= 4.5 \times 10^{-6}
\end{aligned}
$$

### C. Hardware Security

Many security mechanisms are implemented at the hardware level as primitives, in order to provide a much stronger guarantee in a more efficient way, such as Trusted Platform Modules that are available from many chip vendors. Furthermore, as the exponential increase in the availability of pins on chip, hardware resource has become rich enough for more aggressive security approaches implemented in the hardware. Most previous research that focuses on the energy breakdown of computing devices are only considering the major energy hungry components such as CPU, cache and memory. However the increasing complexity of hardware security primitives indicates that energy consumption of hardware security needs to be an important factor in hardware design.

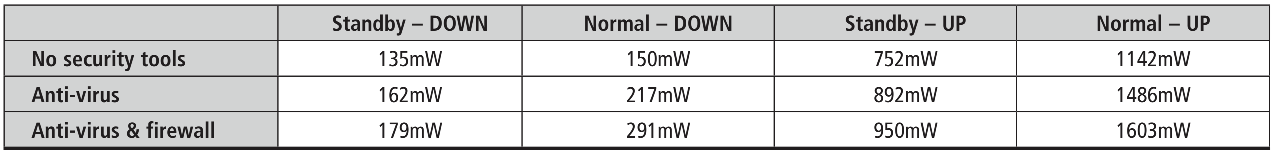
As an example of security mechanisms embedded in various hardware components, main memory has been one of the most widely targeted component for enhanced security policy. Research in [8] studies the energy overhead of two potential types of security strategies implemented in the memory: memory authentication that protects the integrity of code and data, and memory encryption that prevents private data leaking and ensures privacy. The results are shown in Figure 4. It can be seen from the figure that advanced security mechanisms embedded in the memory can impose an average energy overhead of 13.42%. The experiments were performed on SPEC 2000 benchmark suite, which has an average instruction count of 131 billion, CPI of 1.97 and data set size of 128MB [13]. Assuming a typical processor with 65W power consumption and 2.8GHz frequency from the paper, we can estimate the energy per byte consumed in hardware memory security as:

$$
\begin{aligned}
EPB_{HDW} &= Power\ Overhead \div Data\ Process\ Speed \\
&= 65 \times 13.42\% \div (128 \times 10^6 \div \\
&\quad (131 \times 10^9 \times 1.97 \div (2.8 \times 10^9))) \\
&= 6.28 \times 10^{-6}
\end{aligned}
$$

More aggressively, security can be enforced from ground up through every logic gate in the hardware, providing a strong guarantee to the system above. Recent approaches such as GLIFT [19] and Caisson [14] both tackle the problem through massive changes to the hardware design with dynamic or static information flow tracking techniques. The energy overhead of such systematic hardware security approaches can vary from 9% up to 180% for a simple processor design, which is expected to further increase when the complexity of hardware grows.

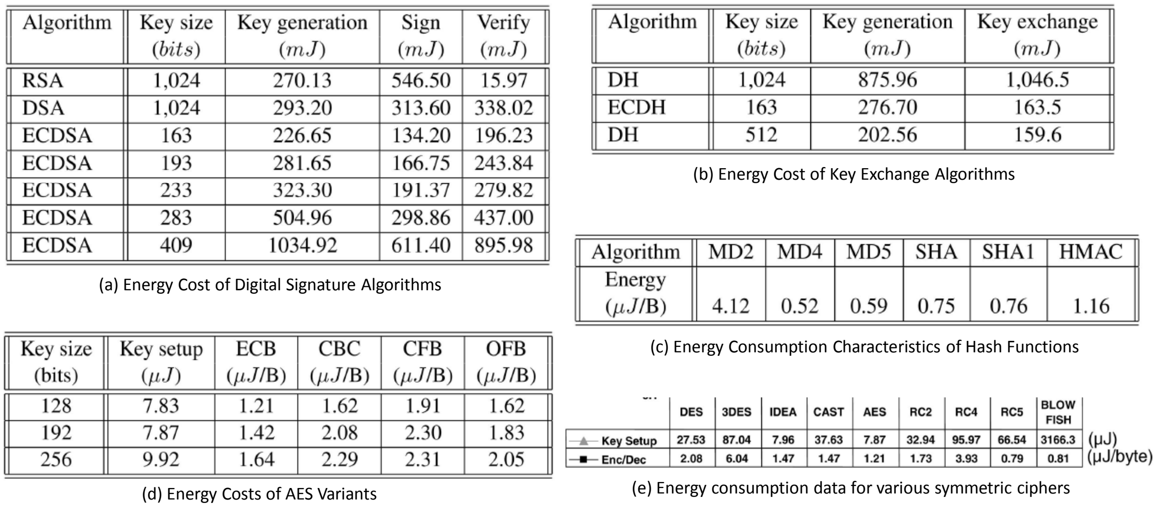
### D. Network Security

Other than providing security guarantees locally on devices and systems, another important but hidden energy sink is the cost of securing every piece of data while they are been transmitted in the network. The integrity and secrecy



| | Standby – DOWN | Normal – DOWN | Standby – UP | Normal – UP |
|---|---|---|---|---|
| No security tools | 135mW | 150mW | 752mW | 1142mW |
| Anti-virus | 162mW | 217mW | 892mW | 1486mW |
| Anti-virus & firewall | 179mW | 291mW | 950mW | 1603mW |

Figure 3. Anti-virus and firewall software energy consumption on mobile devices [4]. DOWN/UP indicates the state of the Wi-Fi. Typical smartphones should spend the majority of time in a standby mode with Wi-Fi up, which gives a 200mW overhead in running security software.



| Algorithm | Key size $(bits)$ | Key generation $(mJ)$ | Sign $(mJ)$ | Verify $(mJ)$ |
|---|---|---|---|---|
| RSA | 1,024 | 270.13 | 546.50 | 15.97 |
| DSA | 1,024 | 293.20 | 313.60 | 338.02 |
| ECDSA | 163 | 226.65 | 134.20 | 196.23 |
| ECDSA | 193 | 281.65 | 166.75 | 243.84 |
| ECDSA | 233 | 323.30 | 191.37 | 279.82 |
| ECDSA | 283 | 504.96 | 298.86 | 437.00 |
| ECDSA | 409 | 1034.92 | 611.40 | 895.98 |

(a) Energy Cost of Digital Signature Algorithms

| Algorithm | Key size $(bits)$ | Key generation $(mJ)$ | Key exchange $(mJ)$ |
|---|---|---|---|
| DH | 1,024 | 875.96 | 1,046.5 |
| ECDH | 163 | 276.70 | 163.5 |
| DH | 512 | 202.56 | 159.6 |

(b) Energy Cost of Key Exchange Algorithms

| Algorithm | MD2 | MD4 | MD5 | SHA | SHA1 | HMAC |
|---|---|---|---|---|---|---|
| Energy ($\mu J$/B) | 4.12 | 0.52 | 0.59 | 0.75 | 0.76 | 1.16 |

(c) Energy Consumption Characteristics of Hash Functions

| Key size (bits) | Key setup ($\mu J$) | ECB ($\mu J$/B) | CBC ($\mu J$/B) | CFB ($\mu J$/B) | OFB ($\mu J$/B) |
|---|---|---|---|---|---|
| 128 | 7.83 | 1.21 | 1.62 | 1.91 | 1.62 |
| 192 | 7.87 | 1.42 | 2.08 | 2.30 | 1.83 |
| 256 | 9.92 | 1.64 | 2.29 | 2.31 | 2.05 |

(d) Energy Costs of AES Variants

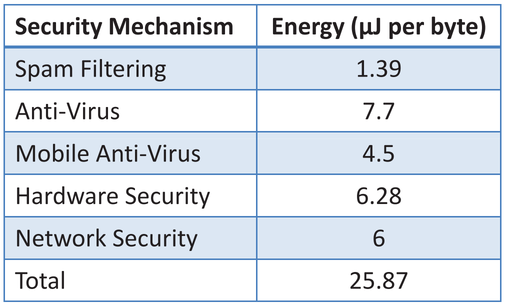
| | DES | 3DES | IDEA | CAST | AES | RC2 | RC4 | RC5 | BLOW FISH | |
|---|---|---|---|---|---|---|---|---|---|---|
| ▲ Key Setup | 27.53 | 87.04 | 7.96 | 37.63 | 7.87 | 32.94 | 95.97 | 66.54 | 3166.3 | (μJ) |
| ■ Enc/Dec | 2.08 | 6.04 | 1.47 | 1.47 | 1.21 | 1.73 | 3.93 | 0.79 | 0.81 | (μJ/byte) |

(e) Energy consumption data for various symmetric ciphers

Figure 5. Energy Overhead Analysis of Network Security Protocol [17].

of all the data flowing around the internet is enforced through a combination of many different security protocols, including various encryption/decryption mechanisms and signature systems. A recent study [17] analyzes the energy overhead of all major network security protocols. A summary of the results is shown in Figure 5. More frequently performed cryptographic and hash algorithms typically cost micro-joule per byte, while signature/key exchange algorithms cost milli-joule per byte. It is estimated that 50mJ energy overhead is required for a mobile client side to communicate 1 Megabytes over SSL. Typical mobile processors consume less than 1/120 (eg. 0.5W for ARM Cortex$^{TM}$-A9) power compared to desktop counterparts, hence the energy per byte consumed in network security for typical PCs can be estimated as:

$$\begin{aligned} EPB_{NET} &= 50 \times 10^{-3} \times 120 \div 10^{6} \\ &= 6 \times 10^{-6} \end{aligned}$$



| Security Mechanism | Energy (μJ per byte) |
|---|---|
| Spam Filtering | 1.39 |
| Anti-Virus | 7.7 |
| Mobile Anti-Virus | 4.5 |
| Hardware Security | 6.28 |
| Network Security | 6 |
| Total | 25.87 |

Table II

SUMMARY: ENERGY CONSUMPTION PER BYTE OF DIFFERENT SECURITY MECHANISMS.

### *E. Summary*

To put things together, Table II shows the energy consumption per byte for each security mechanism we have discussed. To have a sense of how much overhead security mechanisms contribute to compared with computation, we look at energy efficiency for typical data center computation such as Hadoop Sort. It is estimated that under default configurations Hadoop sort can process 100 records per Joule energy (each record contains 100 bytes value and 10 bytes key) [7]. we can estimate the energy per byte for computation as:

$$\begin{aligned} EPB_{COM} &= 1 \div (100 \times (100 + 10)) \\ &= 90.9 \times 10^{-6} \end{aligned}$$

Compared with the total energy consumption of $25.87\mu J/B$ for security, security mechanisms cost almost 30% of energy consumed in useful computation. Such significant overhead makes the energy wall we have been facing even worse. The

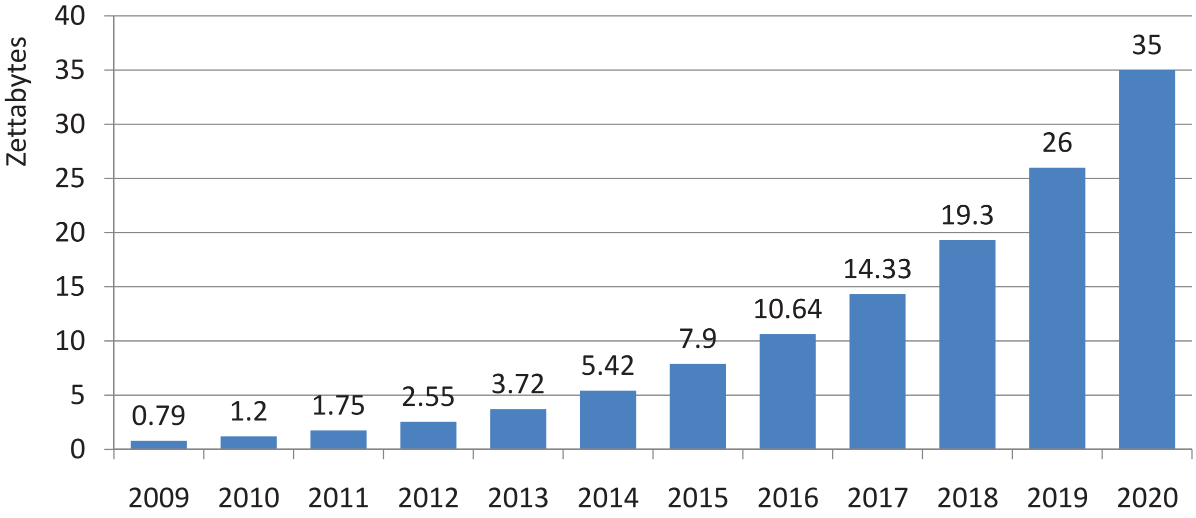


Figure 6. The Rapid Growth of Global Data [2].

overhead of security mechanisms is also expected to further increase as the data volume increases, as shown in Figure 6 which gives the global data growth up to 2020 [2]. Based on the data growth rate, we can estimate the amount of energy consumed by pure computation compared to the total energy with security overhead included. The results are shown in Figure 7. We also include the imaginary energy consumed by computation if data grows at the same rate as Moore's Law. We can see that there is already a big gap between Moore's Law and the reality, while the massively applied security mechanisms further increase such gap. We shall pay more attention to energy-aware security mechanisms with more systematic design methodologies.

## III. Conclusion

Global data has been growing much faster than Moores Law, creating an "energy wall" in designing systems to exploit that data. This problem is exacerbated by the increasing demand for security to ensure both secrecy and integrity of all the data in the cloud. Those security mechanisms can consume significant energy overhead if designed in an ad-hoc manner. In this work we estimate the energy consumption per byte for several major security mechanisms, and compare them with the energy consumed in useful computation. We estimate that security mechanisms can consume almost 30% of computation energy, becoming a giant target for energy efficiency and optimization. We propose that more systematic energy-aware security mechanism designs should merit more attention and will play an important role in green IT.

## References

[1] Big data, big impact: New possibilities for international development. Technical report, World Economic Forum, 2012.

[2] Big data infographic. Technical report, Computer Science Corporation, 2012. http://www.csc.com/insights/flxwd/78931-big_data_growth_just_beginning_to_explode.

[3] AntiVirus Ware. Antivirus performance tests. http://www.antivirusware.com/testing/performance/.

[4] L. Caviglione and A. Merlo. The energy impact of security mechanisms in modern mobile devices. *Network Security*, 11(2), 2012.

[5] L. Caviglione, A. Merlo, and M. Migliardi. What is green security? In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 366–371. IEEE, 2011.

[6] L. Caviglione, A. Merlo, and M. Migliardi. Green-aware security: Towards a new research field. *The International Journal of Information Assurance and Security (JIAS)*, 7:338–346, 2012.

[7] Y. Chen, L. Keys, and R. H. Katz. Towards energy efficient mapreduce. Technical Report UCB/EECS-2009-109, EECS Department, University of California, Berkeley, Aug 2009.

[8] S. Chhabra and Y. Solihin. Transactions on computational science X. chapter Green secure processors: towards power-efficient secure processor design, pages 329–351. Springer-Verlag, Berlin, Heidelberg, 2010.

[9] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc. Design of ion-implanted mosfet's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 1974.

[10] M. A. Ed Grochowski. Energy per instruction trends in Intel®Microprocessors. *Technology@Intel Magazine*, 2006.

[11] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 35–47, New York, NY, USA, 2010. ACM.

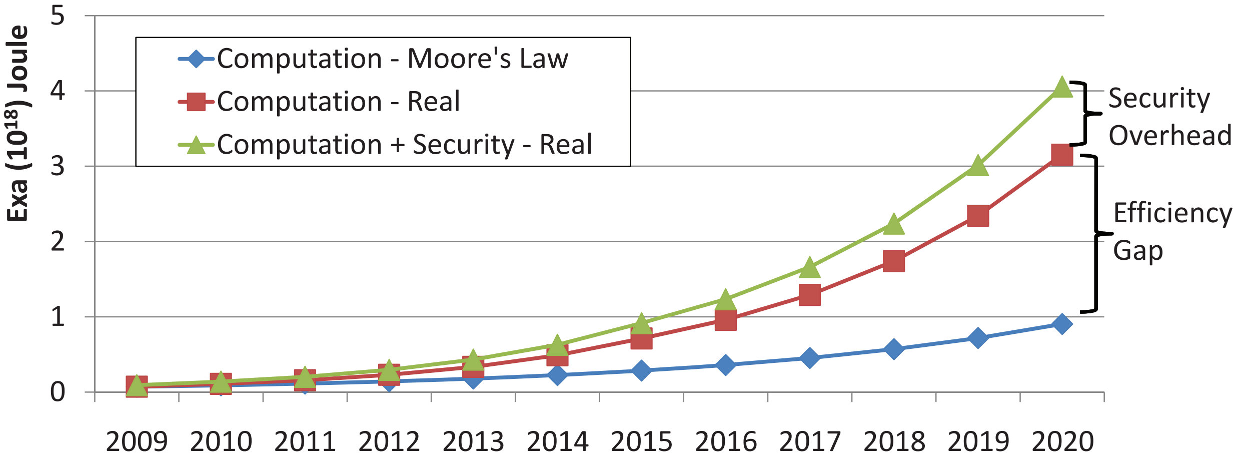


Figure 7. Energy consumed by computation only compared with energy consumed by adding security overhead. The trend is predicted up to 2020 based on data growth rate.

[12] C. Grier, K. Thomas, V. Paxson, and M. Zhang. @spam: the underground on 140 characters or less. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

[13] A. Jaleel. Memory characterization of workloads using instrumentation-driven simulation – a Pin-based memory characterization of the SPEC CPU2000 and SPEC CPU2006 benchmark suites, 2010.

[14] X. Li, M. Tiwari, J. K. Oberg, V. Kashyap, F. T. Chong, T. Sherwood, and B. Hardekopf. Caisson: a hardware description language for secure information flow. In *Proceedings of the 32nd ACM SIGPLAN conference on Programming language design and implementation*, PLDI '11, pages 109–120, New York, NY, USA, 2011. ACM.

[15] McAfee, ICF. The carbon footprint of email spam report. 2009. http://resources.mcafee.com/content/NACarbonFootprintSpam.

[16] M. Migliardi and A. Merlo. Modeling the energy consumption of distributed ids: A step towards green security. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 1452–1457. IEEE, 2011.

[17] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on Mobile Computing*, 5(2):128–143, Feb. 2006.

[18] Symantec. Spam Volume, Brightmail IQ Services.

[19] M. Tiwari, H. M. Wassel, B. Mazloom, S. Mysore, F. T. Chong, and T. Sherwood. Complete information flow tracking from the gates up. In *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems*, ASPLOS '09, pages 109–120, New York, NY, USA, 2009. ACM.

[20] Winter Corporation. Winter TopTen Program, 2003.