Accelerating Techniques for Rapid Mitigation of Phishing and Spam Emails

Pranil Gupta, Ajay Nagrale and Shambhu Upadhyaya
Computer Science and Engineering
University at Buffalo
Buffalo, NY 14260
{pagupta, anagrale, shambhu}@cse.buffalo.edu

Abstract - *Spam filters that are implemented using Naïve Bayesian learning techniques are widely deployed worldwide with email clients such as Outlook®. These filters that are deployed on end user's computers and typically used to filter out spam for individual users are effective when the spam load is around 400-500 spam emails per day per user. However, when the spam load increases, these solutions prove to be slow and hence insufficient for practical use. In this paper, we identify the computation intensive functions of such machine learning algorithms and solve the performance issues by implementing these functions on hardware. Earlier similar approaches made use of specialized hardware chips or co-processors to achieve such acceleration. These chips being dedicated hardware represent a cost and scalability limitation. Our approach makes use of a more generic Intel desktop processor, viz. Tolapai (Intel EP80579) that has several built-in cryptographic functionalities, viz. security accelerators for bulk encryption, authentication, hashing and public/private key generation. Experimental results show that significant acceleration can be achieved by migrating some of the functionalities to hardware in a transparent way.*

Keywords - Hardware acceleration, Hashing, Intel EP80579, Naïve Bayesian, Phishing attacks, Spam filters, Tolapai

I. INTRODUCTION

A common synonym for spam is unsolicited bulk email (UBE). Definition of spam usually includes the aspects that email is unsolicited and sent in bulk [8]. As of 2009, the estimated damage due to spam worldwide is $130 billion, of which $42 billion is in the U.S. alone [5]. According to Symantec report, in the year 2008, around 80% of all Internet traffic has been spam emails [10]. Spam is a medium for fraudsters to scam users to enter personal information on fake Web sites using email forged to look like it is from a bank or other organization such as PayPal®[14]. This is known as phishing [8]. According to recent Gartner report, in the year 2007, more than 25,000 unique phishing emails hijacking 150 different brands were sent out on a monthly basis resulting over $3 billion dollars in damage worldwide. Furthermore, the report also estimates that due to their lucrative nature phishing attacks are going to skyrocket through 2009 [9].

There are various techniques used to detect phishing and fight spam. Among them Bayesian filtering methods are most popular because of their simplicity and high filtering accuracy [1, 3, 11]. Machine learning algorithms such as Naïve Bayesian are computation-intensive algorithms and hence overload the processors. Email spam filters that make use of Naïve Bayesian and are implemented in software tend to become sluggish as spam traffic increases.

Our goal in this paper is to introduce a hardware accelerating SOC processor which will improve the performance of Naïve Bayesian spam filters and phishing attack detectors. We achieve this performance improvement by moving hashing functions used in Naïve Bayesian spam filters to hardware. We, however, are not trying to design a new algorithm for spam filtering. On the other hand, we are proposing acceleration technique for existing Naïve Bayesian Spam filters.

The organization of the paper is as follows. Section 2 describes the related work on spam filters and earlier approaches to move spam filter's functionalities to hardware. Section 3 details the Intel processor, the Naïve Bayesian approach and the acceleration techniques. The experimentation that were carried out to prove our performance goals are described in Section 4. The results, conclusions and future work are briefed out in sections 5 and 6 respectively.

II. RELATED WORK

Sahami et al. [2] have shown the use of Naïve Bayesian in classifying email as spam and legitimate (ham). They have tested their Naïve Bayesian spam filter on real usage scenario with an accuracy of 92%. They have shown that by considering domain-specific features of this problem in addition to the raw text of Email messages, much more accurate filters can be produced. Chandrasekaran et al. [13] have shown the ability to identify phishing based on structural properties of email using machine learning algorithms.

Graham [3] has used statistical filtering for spam detection using one corpus of spam and another one of non-

spam emails with each having about 4000 messages in it. The entire text, including headers and embedded html and JavaScript of each message in each corpus was scanned and tokenized. Two large hash tables, one for each corpus, mapping tokens to number of occurrences were created. Next a third hash table was created, this time mapping each token to the probability that an email containing it is a spam. As noted much of the intensive part of the algorithm is hashing.

Alkabani et al. [4] have tried an approach to move functionality of a Naïve Bayesian filter to hardware, namely hash table and tokenizing. A software Bayesian spam filter was implemented and run on the Microblaze processor soft-core on a Xilinx FPGA. This was used as a first step towards designing an efficient spam filtering platform based on the Microblaze processor. When the hash table was replaced by a content addressable memory, the overall performance achieved an average improvement of 10%. This technique however requires a special co-processor for spam filtering. Our approach is to use a multi-purpose cryptographic processor provided by Intel to achieve these results. Being a generic processor, it can run like a normal desktop processor and at the same time make use of its accelerating capabilities for security applications.

III. EXPERIMENTAL BASIS

We are using Intel EP80579 (Tolapai) processor to achieve acceleration of Naïve Bayesian Spam filtering process. The details are described next.

3.1 Intel EP80579 Processor

The Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology, Tolapai, is a complete System-on-a-Chip for Security, Communications, Storage and Embedded Designs. This SOC processor delivers a significant leap in architectural design, with an outstanding combination of performance, power efficiency, footprint savings and cost-effectiveness compared to discrete, multi-chip solutions. Using multi-chip solutions for different security applications pose scalability and cost issues. Tolapai aims to provide a single chip solution for security applications. The integrated accelerators in this SOC processor support Intel QuickAssist Technology through software packages provided by Intel. These software packages provide the library structures to integrate security and/or VoIP functionality into the application, completely adjunct to the Intel architecture complex, freeing up CPU cycles to support additional features and capabilities. This provides the efficiency of customized hardware with the flexibility to design diverse applications with one platform. The design also includes PCI Express, High Speed Serial1 (HSS) ports for TDM or analog voice connectivity, security accelerators for bulk encryption, hashing and public/private key generation [6].

3.2 Naïve Bayesian Spam Filtering Algorithm

Naïve Bayesian is a text classifier algorithm that analyzes textual features of an email to identify it as a ham or spam email based on probabilistic scoring of its textual attributes. The Naïve Bayesian approach consists of two phases – training phase and the learning phase.

3.2.1 Training Phase

The training phase scans an existing corpus of spam and ham emails. It consists of three main steps.

3.2.1.1 Parsing

An email is parsed to identify different sections such as headers, body, to, from, subject, etc. Based on different filters different parsing techniques are used.

3.2.1.2 Tokenization

Tokenization consists of creating tokens from different sections of email. These tokens will be later used to classify emails. Tokenization process is different for different spam filters. But it is one of the computation intensive functions of Naïve Bayesian spam filters.

3.2.1.3 Hash Maps

Hash tables are created for tokens. Normally two separate hash tables are created for spam and ham emails. Probabilistic scores are calculated for each token and a third hash table is created for mapping probabilities to tokens.

3.2.2 Classification Phase

In classification phase an incoming email is classified as a spam or ham. An incoming email is first tokenized to get individual tokens. The corresponding probabilities for each token are retrieved from the hash table by hash lookup. Finally, Naïve Bayesian formula is used to classify this email as ham or spam using these probabilities. The formula that the software uses to determine these probabilities is derived from Bayes' theorem. It is, in its most general form:

$$\Pr(S|W) = \frac{\Pr(W|S) \cdot \Pr(S)}{\Pr(W|S) \cdot \Pr(S) + \Pr(W|H) \cdot \Pr(H)}$$

Where:

- *Pr*(*S* | *W*) is the probability that a message is a spam, knowing that the word W is in it;
- *Pr*(*S*) is the overall probability that any given message is spam;
- *Pr*(*W* | *S*) is the probability that the word W appears in spam messages;
- *Pr*(*H*) is the overall probability than any given message is not spam (is "ham");
- *Pr*(*W* | *H*) is the probability that the word W appears in ham messages [12].

3.3 Achieving Acceleration

As may be noted from above section, much of the computation goes into tokenizing and calculating hashes.

Section 4 gives actual statistics about these functions. Both functions form integral part of the training and classification phase.

The Intel® EP80579 Integrated Processor is a System-On-a-Chip (SOC), integrating the Intel® Architecture core processor, the Integrated Memory Controller Hub (IMCH) and the Integrated I/O Controller Hub (IICH) all on the same die. In addition, it has integrated Intel® QuickAssist Technology, which provides acceleration of cryptographic and packet processing. Fig. 1 shows the architecture of Intel EP80579.

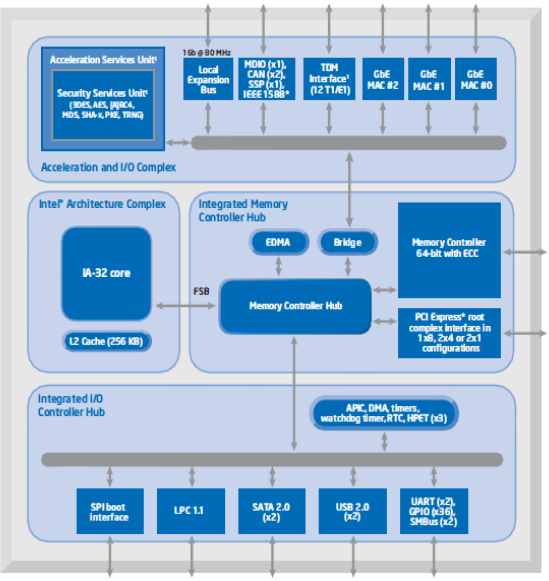

Fig. 1. Block Diagram for Intel EP80579 [6]

The Intel® QuickAssist Technology components, housed in the Acceleration and I/O Complex (AIOC), are as follows:

- The Security Services Unit (SSU) provides acceleration of cryptographic processing for most common symmetric cryptography (cipher algorithms such as AES, 3DES, DES, (A)RC4, and messages digest/hash functions such as MD5, SHA-1, SHA-2, HMAC, etc.); asymmetric cryptography (modular exponentiation to support public key encryption such as RSA, Diffie-Hellman, DSA); and true random number generation.
- The Acceleration Services Unit (ASU) includes packet processing acceleration engines.

We utilize this acceleration capability of Intel EP80579 to improve the performance of Naïve Bayesian spam filter. The hashing function, as identified in Section 3.2 is moved to hardware using the hashing APIs provided by Intel QuickAssist technology.

IV. EXPERIMENTAL SETUP

Intel EP80579 Development Board was assembled as per the instructions by Intel User guide for EP80579 [7]. Fig. 2 depicts a setup for experiments in our lab. RedHat Linux kernel was installed on this system along with software drivers and kernel modules for QuickAssist Technology provided by Intel. These modules consist of hardware APIs necessary to utilize the accelerating capabilities of Tolapai.

4.1 Naïve Bayesian Spam Filter Profiling

A C code for naïve Bayesian spam filter was implemented in software. The training phase was completed using 40 emails of spam and 40 emails of ham. Sample emails were classified using this training data. The various computation times were calculated by profiling the program. The timings are listed as below.

Table 1. Training Phase

Function	Time (S)	Uses Hashing
Parse	0.00344	No
Tokenize	0.03753	No
Add token	0.09573	Yes

Table 2. Classification Phase

Function	Time (S)	Uses Hashing
Tokenize	0.00034	No
Find token	0.00583	Yes
Classify	0.00421	No

It is evident from the above results that hashing and tokenization are the most computation intensive functions of the program.

4.2 Hashing acceleration

A program was written that makes use of hardware acceleration APIs of Tolapai to calculate SHA1 hash digest of input strings. Additionally, for the purpose of comparison, a software code was implemented that does the same SHA1 calculation and does not make use of these hardware APIs.

An input sample of 700000 string tokens was created. SHA1 hashing digests for these input tokens were calculated in a gradual manner in software. The same input sample was fed to the hardware program and SHA1 hashing digests were calculated. The number of operations and processing times were noted for the software and hardware experiments. The results are described in the following section.

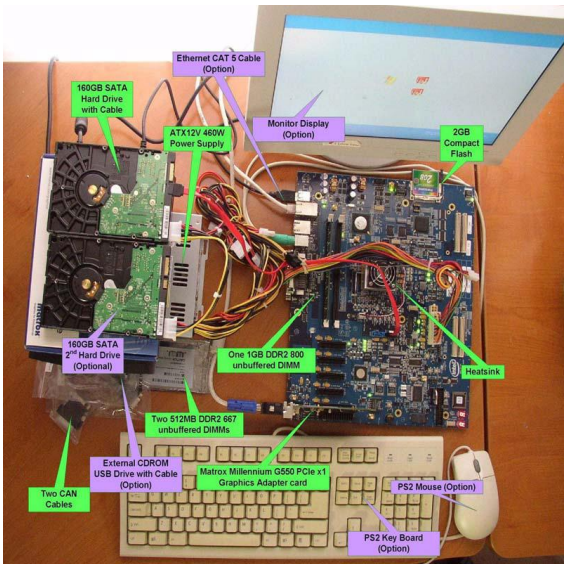

Figure1. Intel EP80579 Lab Setup [7]

V. RESULTS & DISCUSSION

The hardware implementation showed significant amount of performance gain over software. The results of our experiments are tabulated in Table 3. As can be seen, an average improvement of 26% was achieved.

Table 3. Hashing Acceleration

No. of Words	Hardware (S)	Software (S)	%Gain w.r.t Software
10000	0.0210	0.0282	25.52%
20000	0.0414	0.0563	26.45%
30000	0.0630	0.0844	25.38%
40000	0.0830	0.1126	26.33%
50000	0.1041	0.1411	26.19%
60000	0.1247	0.1702	26.74%
70000	0.1469	0.1969	25.38%
80000	0.1678	0.2253	25.52%
90000	0.1893	0.2533	25.28%
100000	0.2095	0.2817	25.61%
200000	0.4201	0.5627	25.34%
300000	0.6297	0.8483	25.77%
400000	0.8355	1.1262	25.82%
500000	1.0236	1.4082	27.31%
600000	1.2355	1.6880	26.81%
700000	1.4367	1.9716	27.13%

The hashing computation in a spam filter can be offloaded from software to achieve an overall performance improvement for Naïve Bayesian spam filters. The data shows that performance for a spam filter can be increased by 25% using the Tolapai processor. Alkabani et al. [4] made use of a dedicated chip to achieve a performance improvement of 10% for hash function of spam filters. Our approach not only achieves better performance but at the same time doesn't require a dedicated hardware co-processor or chip.

Being a generic processor, no additional chip is required to achieve this acceleration. The interface to hardware through APIs that are provided by Intel, make the acceleration completely transparent to the user. The implementation of spam filter is still flexible, thus, overcoming shortcomings of any dedicated hardware modules. Other security applications can make use of the accelerating capabilities of Tolapai without additional costs. Our conclusion is that, spam filters utilizing Tolapai will perform better to detect spam and phishing emails even when the traffic increases in real-time.

VI. FUTURE WORK

Our future work would consist of moving more functions of Naïve Bayesian spam filters to Intel EP80579. Tokenizing is one major time consuming function of such filters which if moved to hardware for speed optimization will improve overall performance of spam filters greatly. We also focus on utilizing Tolapai to implement other security applications which will be more efficient in performance than their software counterparts.

ACKNOWLEDGEMENTS

This research has been done in part through a grant from Intel Corporation. The authors would like to thank Vinodh Gopal and Madhusudhanan Chandrasekaran for the useful discussions during the course of this research.

REFERENCES

[1] S. Saroiu, S. D. Gribble and H. M. Levy, Measurement and Analysis of Spyware in a University Environment, In Proceedings of the 1st ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI), San Francisco, CA, 2004.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. In Learning for Text Categorization - Papers from the AAA1 Workshop, pp. 55-62, Madison Wisconsin. AAAI Technical Report WS-98-05, 1998.

[3] Paul Graham - A Plan for Spam. Online-http://paulgraham.com/spam.html

[4] Alkabani, Y.M.; El-Kharashi, M.W.; Bedor, H.S. Hardware/Software Partitioning of a Bayesian Spam Filter via Hardware Profiling; Industrial Electronics, 2006 IEEE

International Symposium on Volume 4, 9-13 July 2006 Page(s):3264 - 3269

[5] Ferris Research – Industries Statistic [Online] http://www.ferris.com/research-library/industry-statistics/

[6] Product Brief - Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Embedded Computing [Online]. Available: http://download.intel.com/design/intarch/ep80579/319944.pdf

[7] Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit User Guide [Online]. Available: http://download.intel.com/design/intarch/ep80579/320067.pdf

[8] Wikipedia, Email Spam, (electronic) -The free encyclopedia. Available [Online]. http://en.wikipedia.org/wiki/E-mail_spam

[9] Gartner, Inc. - Gartner Survey [Online] - http://www.gartner.com/it/page.jsp?id=565125

[10] Symantec Global Internet Security Threat Report [Online]. http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiv_04-2009.en-us.pdf

[11] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C.D. Spyropoulos, "An evaluation of naive Bayesian anti-spam filtering," in Proceedings of the Workshop on Machine Learning in the New Information Age, G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning, Barcelona, Spain, May 2000, pp. 9-17.

[12] Wikipedia, Bayesian spam filtering, (electronic) -The free encyclopedia. Online – http://en.wikipedia.org/wiki/Bayesian_spam_filtering

[13] Madhusudhanan Chandrasekaran, Krishnan Narayanan, Shambhu Upadhyaya. Phishing Email Detection based on Structural Properties, NYS Cyber Security Conference, Albany, NY, June 2006.

[14] Paypal – www.paypal.com