Energy Consumption Analysis for Cryptographic Algorithms with Different Clocks on Smart Cards in Mobile Devices

Chu-Hsing Lin
Department of Computer Science
Tunghai University
Taichung, Taiwan, R.O.C.
chlin@go.thu.edu.tw

Guan-Han Chen
Department of Computer Science
Tunghai University
Taichung, Taiwan, R.O.C.
g98350048@thu.edu.tw

Shih-Pei Chien
Department of Computer Science
Tunghai University
Taichung, Taiwan, R.O.C.
g99350011@thu.edu.tw

***Abstract*—In this paper, energy consumption issue is investigated for cryptographic algorithms operated on smart cards in mobile devices using different clocks. We implement cryptographic algorithms on a security chip that is embedded in a smart card. When the CPU clock changes the execution time and the energy consumption will thus change. The experimental results of this paper show that to reduce CPU clock effectively reduce the energy consumption, but that will make the execution time increase. From the viewpoint of power consumption, to reduce the CPU clock does not necessarily save the energy, moreover, an algorithm will need longer execution time. From the results, programmers could select suitable cryptographic algorithms following energy consumption, execution time, and security level desired.**

Keywords- Cryptographic, LabView, Smart card, CPU clock.

I. INTRODUCTION

In the present society, Trading Behaviors on the Internet using computers or mobile devices have become a part of daily life. People often buy/sell products or pay duty [1] from the Web. To determine the identities of consumers is an important key of e-commence so that buyer and seller don't subject to fraud. For many systems, it only uses account and password to determine the identity of a consumer. However, the security level of using account and password only is limited to small transactions. For some important financial transactions is not enough, such as stock trading, internet banking operation, and pay duty. These financial transactions involves larger amount money, it needs to have higher security to confirm the identity than small transactions. In order to achieve higher security, financial transactions often use additional equipment or mechanism to enhance that.

In order to prevent data theft, in mobile commerce we directly on the phone use a smart card. For convenience, manufacturers will have smart card chips and MicroSD card combination (hereinafter referred to as Security MicroSD Card, SMC). The SMC input and output data via the MicroSD interface, and the phone can also operate a smart card chip. Due to encryption and decryption operation are done in the card, the key is also stored in the smart card chip. Therefore, the key can be tapped or the stolen with very low possibility, comparing to the cases of doing cryptographic operations in the phone. In other word, doing the cryptographic operations through SMC will have higher security level. When operating smart cards through the card reader and used on desktop computers, we do not consider the energy consumption issue. But for smart phone with battery, it has strict requirements for energy consumption. In this paper, we consider applications using smart cards [1][2] and investigate the energy consumption issues [5][6]. We implement cryptographic algorithms on a security chip that is embedded in a smart card. We are interested in the issue that how the CPU clock influences the execution time and the energy consumption. The experimental results show that to reduce CPU clock effectively reduce the energy consumption, but that will make the execution time increase. From the viewpoint of power consumption, to reduce the CPU clock does not necessarily save the energy, moreover, an algorithm will need longer execution time. From the results, programmers could select suitable cryptographic algorithms following energy consumption, execution time, and security level desired.

The rest of this paper is organized as follows: Section 2 describes background about cryptography algorithms. Section 3 describes the experimental setup used in this paper to execute the cryptography algorithms. Section 4 presents the results of our energy measurements. Section 5 summarizes the conclusions for this work.

II. BACKGROUND

We introduce the relevant symmetric encryption, asymmetric encryption, and hash functions used in our experiments.

A. Symmetric Encryption

Symmetric algorithm is faster than asymmetric algorithm in execution time. For encryption and decryption, the key length is short and low cost is required for operation. Currently symmetric algorithms are often used for the confidentiality of information security.

- Advanced Encryption Standard (AES)

AES [7] has replaced DES [8] and becomes widely used encryption algorithm after 2000 by NIST. The inventor of the AES (or called Rijndael cipher) submitted the original version, which can be designated by their flexible block length and key length, 128, 192 and 256 bits, respectively. Those have nine different possible conbinations. In the final version of the AES, the key length remains to have three possibilities, but the block length is limited to 128 bits.

978-0-7695-4532-5/11 $26.00 © 2011 IEEE
DOI 10.1109/BWCCA.2011.61

B. Hash algorithms

A hash algorithm can input any length of message and output a fixed length of hash code. Even if the message has a small change, such as 1 bit or 1 byte, which will change the output of hash value. Thus, hash algorithms are often used to test whether a message has been tampered with. Commonly used hash algorithms are such as MD5 [9], SHA1 [10] and SHA256 [11].

III. EXPERIMENTAL SETUP

In this section, we illustrate experimental environment setup and describe software and hardware that were used in this experiment. We also explain our measuring energy consumption scheme.

A. Hardware and software

We used National instruments LabVIEW[15] (NI LabVIEW) 2010 software and related hardware to build power measurement system. NI LabVIEW through intuitive graphical line and diagram to establish flow charts to develop a complete measurement, testing, and control systems. LabVIEW integrate thousands of hardware devices and support libraries to work for high-level analysis, present information, and provide a strong virtual instrumentation functions. LabVIEW platform can work on a variety of operating systems.

The security chip used in the experiment is ST33F1M [14] that is produced by STMicroelectronics. ST33F1M includes SecurCore® SC300 CPU Core [13] by ARM designing, the core is 32-bits RISC core based on Context M3 core with high performance and low dynamic power consumption. This chip supports multiple protocols, which asynchronous receiver transmitter for ISO/IEC 7816-3[13] T=0 and T=1 protocols, serial peripheral interface (SPI), and single write protocol (SWP) interface for communications for NFC router. The ST33F1M can use in mobile communications, multimedia, and banking. The PC used in this experiment is with a 2.60GHz Intel Pentium Dual-Core CPU E5300 having 4 GB of RAM running the Windows XP OS. The security software are provided from OpenSSL [12] open source project.

B. Experimental environment setup

In Figure 1, the security chip is measured by connecting a resistor 2Ω between the Vcc pin of security chip and the power supply. The voltage drop across the resistor is measured using BNC-2110[15]. BNC-2110 connect to data acquisition (DA) card and send measurement data. DA card is in the PC running the LabVIEW [15] software.

This experiment used “The Declaration of Independence of The United States of America” as encryption/decryption data. The file size is 31242 bytes, is .txt file. The file transmitted to the chip by binary data stream, the chip returns data that is processed by the same data stream.

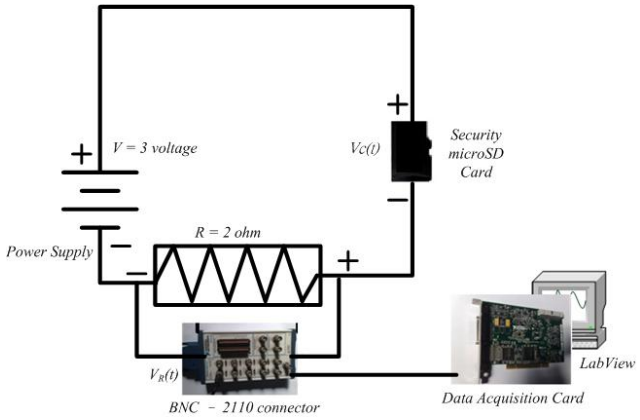

Figure 1 Experimental environment

C. Measuring energy consumption scheme

In this experiment, the measured voltage difference across the resistor, and then converted into energy. First of all, to obtain the input voltage V. The security chip execute cryptographic algorithms to obtain n sampling point, Voltage difference across the resistor is $V_R(t)$. And because the total voltage is equal to the sum of individual voltage, so we can get security chip voltage $V_C(t) = V - V_R(t)$. By Ohm's law, therefore the current per unit time is $I_t = V_C(t) / R$, getting instantaneous ampere. And energy is $J_t = I_t * V_C(t)$.

The sum of all the energy divided by sample frequency, the total energy consumption can be J, such as sampling frequency of 25 KHz, generating 25000 sample points per second. In other words, each sampling point is $1/25000$ sec. Table 1 lists the experimental parameters used in our paper.

TABLE 1. EXPERIMENT PARAMETERS

Notation	Description
V	input voltage.
$V_R(t)$	The t-th sampling point, the voltage across the resistor.
$V_C(t)$	The t-th sampling point, voltage on security chip, $V_C(t)=V-V_R(t)$
I_t	The t-th sampling point, the ampere across the resistor, $I_t = V_C(t) / R$
n	The total number of sampling points.
R	Resistor, in this thesis is 2 ohms.
J_t	The t-th sampling point, security chip energy consumption, $J_t = I_t * V_C(t)$
J	Total energy consumption , $\frac{\sum_{t=1}^{n} J_t}{rate}$, rate is sampling rates.

IV. EXPERIMENTAL RESULTS

In this section, we present an empirical analysis of the energy consumption characteristics of cryptographic algorithms using different clocks [3][4]. We analyze symmetric algorithms AES and some of the hash algorithms. For data integrity, we also used hash algorithms to validate whether the received data is equal to the original one. For data confidentiality, we used symmetric algorithms to ensure the privacy in transmission. We investigate the influence of clock speed of cryptographic algorithms on their energy consumption [6]. On the other hand, we also recorded cryptography algorithm execution timeunder different clocks. Then the energy consumption and execution time are used as performance assessment for different conbinations. The results can be used as an important reference for the security system designers [6].

A. Symmetric algorithm-AES

AES is a popular symmetric algorithm, the computation cost and safety with good competition, especially in the 32-bit system, which shows the effectiveness of AES. In

Figure 2, we can see security chip voltage changes by AES ECB mode, key length of 128 bits in the highest CPU clock. Because I/O access limits and security chip memory constraints, we divide the encrypted file into eight parts to send to security chip to conduct encryption, each transfer 4000 bytes. First of all, from 1.9 sec to 1.95 sec it shows the voltage change that generates the round keys.

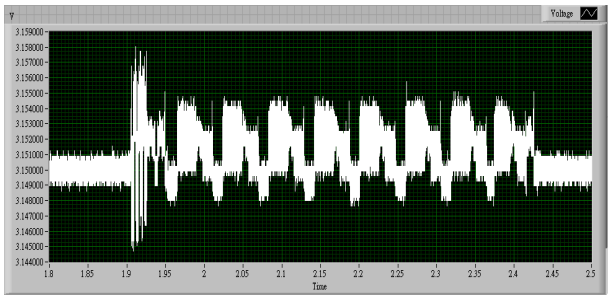

Figure 2 Aes128_ECB using the highest clock

In Table 2, we can see the execution time and energy consumption that use three kind of AES key length, when using the highest CPU clock speed with four block mode. T represents the execution time; in units of ms. P is the consumption of energy, in units of μJ. In particular, in the table energy consumption is the energy of individual bytes, rather than the total energy consumption. The total energy consumption is for the eight encryptions and does not include the round key generation. In Table 3, we can see that when using the high-speed CPU clock divided by 2, the energy consumption is reduced but the execution time increased.

TABLE 2. ENERGY COMSUMPTION AND EXECUTION TIME OF AES USING THE HIGHEST CLOCK

Highest clock								
Mode	ECB		CBC		OFB		CFB	
Length	T(ms)	P(μJ)	T(ms)	P(μJ)	T(ms)	P(μJ)	T(ms)	P(μJ)
AES-128	485	0.449	515	0.495	516	0.471	516	0.471
AES-192	531	0.483	558	0.526	546	0.508	547	0.520
AES-256	578	0.527	609	0.548	594	0.547	594	0.548

TABLE 3. ENERGY CONSUMPTION AND EXECUTION TIME OF AES USING THE HIGHEST CLOCK DIVIDE 2

Highest clock divide 2								
Mode	ECB		CBC		OFB		CFB	
Length	T(ms)	P(μJ)	T(ms)	P(μJ)	T(ms)	P(μJ)	T(ms)	P(μJ)
AES-128	641	0.384	719	0.421	687	0.405	703	0.406
AES-192	734	0.421	781	0.457	765	0.442	781	0.452
AES-256	781	0.464	859	0.503	844	0.489	843	0.489

B. Hash algorithms-MD5, SHA1, and SHA256

As previously, a file is divided into eight parts for transmission. First, it setups the initialization of SHA256, and then transmit data to the security chip. Each transmission executes operation and generates the middle value. Finally, return the result, the hash code. In Figure 4, SHA256 using the highest clock divided by 2 shows longer encryption time clearly.

In Table 4, it shows energy consumption and execution time of hash algorithms on the case of the highest CPU clock. The execution time of MD5 is 344 ms and the energy consumption of each byte is 0.305 μJ. The execution time of SHA1 is 375 ms and the energy consumption of each byte is 0.333 μJ. The execution time of SHA256 is 500 ms and the energy consumption of each byte is 0.439 μJ.

In Table 5, it shows energy consumption and execution time of hash algorithms on the case of the highest CPU clock divided by 2. The execution time of MD5 is 407 ms and the energy consumption of each byte is 0.230 μJ. The execution time of SHA1 is 484 ms and the energy consumption of each byte is 0.276 μJ. The execution time of SHA256 is 703 ms and the energy consumption of each byte is 0.400 μJ.

TABLE 4. ENERGY CONSUMPTION AND EXECUTION TIME OF HASH ALGORITHMS USING THE HIGHEST FREQUENCY

Highest clock		
Algorithm	T (ms)	P(μJ)
MD5	344	0.305
SHA1	375	0.333
SHA256	500	0.439

TABLE 5. ENERGY CONSUMPTION AND EXECUTION TIME OF HASH ALGORITHMS USING THE HIGHEST FREQUENCY DIVIDE 2

Highest clock divide 2		
Algorithm	T (ms)	P(μJ)
MD5	407	0.230
SHA1	484	0.276
SHA256	703	0.400

C. *Performance and Security analysis*

In symmetric encryption part, AES-128 only has 10 rounds, so the energy consumption is minimal, the execution time is the shortest. In the energy consumption of the three modes, AES-256 has longest execution time and energy consumption is also the largest. In Figure 3 and Figure 4, ECB mode doesn't use the initial vector, energy consumption is minimum and run fastest. So ECB mode is saving the most energy. But it lacks the initial vector such that the same plain text in message is encrypted become the same cipher text. Comparison with CBC mode that encrypt close to 32Kbytes, the execution time only increases 30 ms, but it has enhanced security.

In Figure 5 and Figure 6, CBC's execution time rate is greater than ECB. AES-128's execution time increased by 1.39 times, 15% less energy consumption. AES-192's execution time increased by 1.40 times, 11% less energy consumption. AES-256's execution time increased by 1.41 times, 8% less energy consumption. However ECB's is more energy efficient under the same conditions. The reduced energy proportion of ECB is greater than CBC. In ECB mode, AES-256 reduces 12% energy consumption using highest clock divided by 2 than highest clock. In this experiment, we can see that reducing the clock speed to reduce power consumption slightly, but it raises a lot of execution time. If the file is too large, the encryption time will be considerable, but the energy savings is limited. From the experimental results, the highest clock in CBC mode is efficient than lower clock.

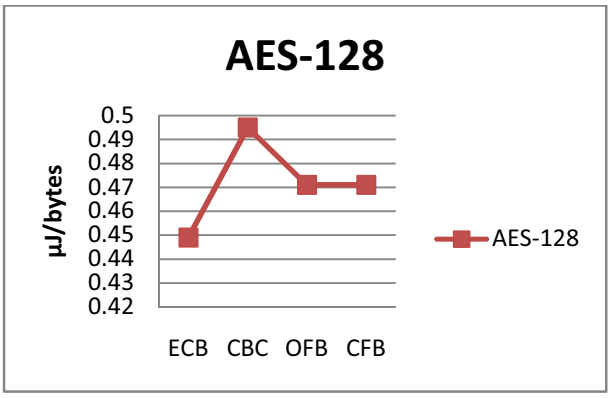

Figure 3 Energy consumption of AES-128 using highest clock

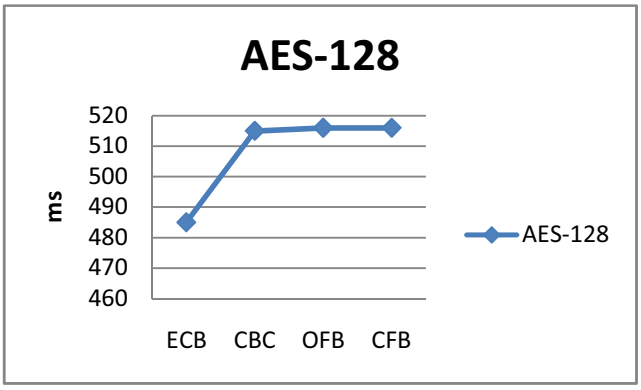

Figure 4 Execution time of AES-128 using highest clock

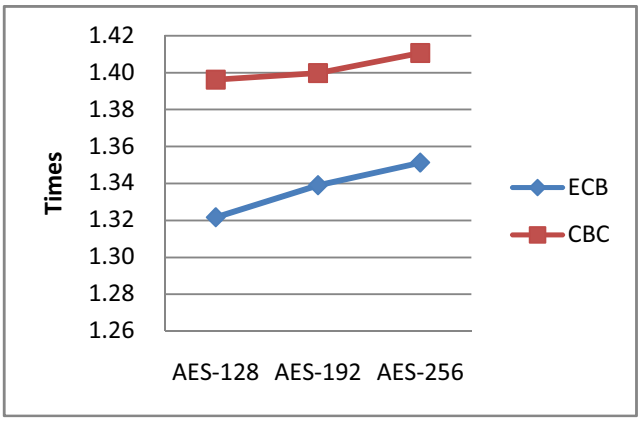

Figure 5 The growth rate of execution time using highest clock divide 2 than highest clock.

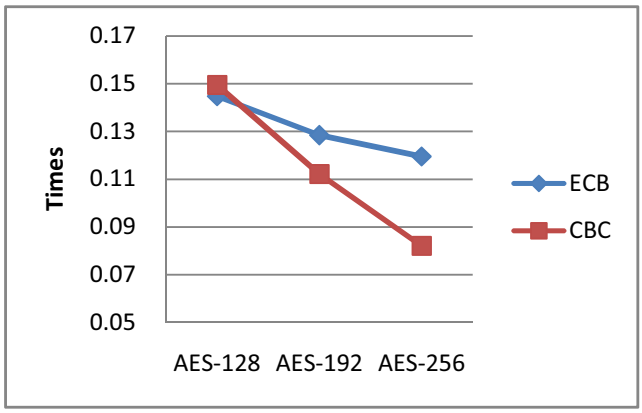

Figure 6 The reduced proportion of energy using highest clock divide 2 than highest clock.

From part of the hash algorithm in Table 4 and Table 5, it can be seen that MD5's execution time increased by 1.18 times, 24% less energy consumption. SHA1's execution time increased by 1.29 times, 17% less energy consumption. SHA256's execution time increased by 1.4 times, 8% less energy consumption. In security, SHA1 is higher than MD5, and SHA256 is higher than SHA1. Comparison of

computational complexity, SHA256 is the highest complexity, SHA1 is the second and MD5 is the lowest.

The results show that the clock speed on low complexity algorithms is more apparent than on the high complexity. But the execution time is the opposite result. That may be caused by basic energy consumption of security chip. Security chip must take fixed energy to maintain communication with the external and internal operations. When the execution of cryptographic algorithms takes too long time, the basic energy consumption of security chip will cancel out the saved energy of reducing clock. If the security system uses hash algorithms, that can select the appropriate algorithm according to demand.

V. CONCLUSIONS

There are many embedded systems, such as in mobile phones, PDAs, tablet PCs, people are accustomed to use these devices to enjoy the service. Service providers often use a smart card as a strategy to protect confidential information, including customer data and system information. Cryptographic algorithms have significantly energy consumption. In particular, the higher complexity of algorithms is more clearly. In consideration of energy saving, how to balance safety and efficacy is an important issue.

As a study in the same platform, the energy consumption of using cryptographic algorithms is not limited to execute algorithms. In order to maintain the operation of smart cards, it has the basic power consumption. So the total energy must be considered when investigating the execution time and energy consumption.

ACKNOWLEDGE

This work was supported in part by the National Science Council under the grant NSC99-2221-E029-034-MY3.

REFERENCE

[1] Yung Fu Changa, C.S. Chenb, and Hao Zhou, " Smart phone for mobile commerce," Computer Standards & Interfaces, Vol. 31, Issue 4, June 2009, pp. 740-747.

[2] H. Mahmoud, K. Alghathbar, "Novel algorithmic countermeasures for differential power analysis attacks on smart cards," Proc. Sixth Information Assurance and Security 2010 (IAS 2010), Aug. 2010, pp.52-55.

[3] Z. Karakehayov, "Dynamic clock scaling for energy-aware embedded systems," Proc. 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 2007(IDAACS 2007), Sept. 2007, pp. 96.

[4] G. Semeraro, G. Magklis, R. Balasubramonian, D.H. Albonesi, S. Dwarkadas, and M.L. Scott, " Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling," Proc. 8th International Symp. High-Performance Computer Architecture, Feb. 2002, pp. 29.

[5] E. Biham, A. Shamir,"Power analysis of the key scheduling of the AES candidates," Proc. 2nd AES andidate, March 1999, pp. 115–121.

[6] 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 Transl. Mobile Computing, vol 5, pp. 128, Feb. 2006.

[7] FIPS 197: Advanced Encryption Standard, http://csrc.nist.gov

[8] FIPS 46, Data Encryption Standard, http://csrc.nist.gov

[9] RFC 1321: The MD5 Message-Digest Algorithm, http://www.ietf.org

[10] RFC 3174: Secure Hash Algorithm 1, http://www.ietf.org

[11] FIPS 180-3: Secure Hash Standard, http://csrc.nist.gov

[12] OpenSSL Project, http://www.openssl.org

[13] ISO/IEC 7816, http://www.iso.org

[14] ST33F1M, http://www.st.com

[15] National Instruments Corp, http://www.ni.com