www.ietdl.org

Published in IET Software
Received on 6th September 2012
Revised on 24th June 2013
Accepted on 24th July 2013
doi: 10.1049/iet-sen.2012.0137

ISSN 1751-8806

Power evaluation methods for data encryption algorithms

Tingyuan Nie[1], Lijian Zhou[1], Zhe-Ming Lu[2]

[1]Communication & Electronic Engineering Institute, Qingdao Technological University, No. 11, Fushun Road, Qingdao, People's Republic of China
[2]School of Aeronautics and Astronautics, Zhejiang University, Zheda Road 38, Hangzhou 310027, People's Republic of China
E-mail: tynie@qtech.edu.cn

Abstract: With the increasingly extensive application of networking technology, security of network becomes significant than ever before. Encryption algorithm plays a key role in construction of a secure network system. However, the encryption algorithm implemented on resource-constrained device is difficult to achieve ideal performance. The issue of power consumption becomes essential to performance of data encryption algorithm. Many methods are proposed to evaluate the power consumption of encryption algorithms yet the authors do not ensure which one is effective. In this study, they give a comprehensive review for the methods of power evaluation. They then design a series of experiments to evaluate the effectiveness of three main types of methods by implementing several traditional symmetric encryption algorithms on a workstation. The experimental results show that external measurement and software profiling are more accurate than that of uninterruptible power system battery. The improvement of power consumption is 27.44 and 33.53% which implies the method of external measurement and software profiling is more effective in power consumption evaluation.

1 Introduction

Owing to the rapid development of network in the past decades, a world filled with 'ubiquitous' computational resources has become reality. Information exchange via network has become increasingly popular. Network transmits data via digital signals, which makes them vulnerable to eavesdropping and unauthorised access. The ability to encrypt messages before transmission is of fundamental importance for the security of ubiquitous computing systems. Therefore, search for the best solution to offer the necessary protection against frauds or infringements from the third party is becoming one of the most essential subjects in security system. A large amount of encryption techniques were developed to comply with specific security requirements of information applications.

Generally, most encryption algorithms can be categorised in several ways: by block size encryption algorithm operates on, there are block encryption algorithm and stream encryption algorithm. The block encryption algorithm encrypts plaintext in chunks of a certain fixed block size like 64× bits, while the stream encryption algorithm encrypts plaintext sequentially by one byte or one bit at a time [1].

By whether key is used for both encryption and decryption, there are symmetric-key algorithm and asymmetric-key algorithm. Symmetric-key algorithm uses the same key for both data encryption and data decryption. The key in symmetric-key algorithm plays a very important role and it must be distributed before data transmission. The strength of symmetric-key algorithm highly depends on the size of the key. Encryption using longer key is harder to be broken and yet needs mass computation. The classical symmetric-key algorithm includes RC2, DES, 3DES, CAST, RC6, Blowfish and advanced encryption standard (AES) [2–4]. Asymmetric-key algorithm solves the problem of key distribution. It uses two different keys at once, a public key and a private key. Public key is used for data encryption and known to the public, whereas private key is used for data decryption and known only to the message recipient. The classical asymmetric-key algorithm includes RSA, Elgamal and ECC [5]. The primary advantage of asymmetric-key algorithm is its high security and convenience. The private key never needs to be transmitted or revealed to anyone else. However, the asymmetric-key algorithm is computationally intensive because of its mathematical functions.

The execution of encryption algorithm consumes both time and power when it utilises computer resources. Applying stronger encryption algorithm on electronic device with battery may consume more power and drain battery faster. Hence, constantly using cryptographically strong algorithm may severely reduce lifetime of battery-powered devices such as cellular phone, PDA etc. Users and designers need to be aware of both benefits and costs for using encryption algorithm. The correct way to design a power-effective security protocol requires a comprehensive understanding for the power consumption of encryption schemes. The

prime work in the first step is to employ a more accurate evaluation method.

In this paper, we study several state-of-the-art power consumption evaluation methods for encryption algorithm. The main contribution of this work is the effectiveness assessment for power consumption evaluation methods through experimental comparisons. The conclusion helps software designers or hardware manufacturers to construct a more power-effective security system.

This paper is organised as follows. Section 2 introduces the related works. Section 3 assesses symmetric encryption algorithms used for evaluation. Introduction of power consumption evaluation methods are given in Section 4. Experiments and analysis are in Section 5, and the concluding remarks appear in Section 6.

2 Related works

In this section, we mainly introduce related researches for algorithm performance evaluation. The detailed power evaluation methods will be presented in Section 4.

Nadeem [6] implemented four popular secret key encryption algorithms to compare their performance by encrypting input files of varying contexts and sizes on different hardware platforms. They concluded that Blowfish is the best performing algorithm and the trade-off between performance and security of encryption algorithm should be further researched.

Prasithsangaree and Krishnamurthy [7] evaluated the performance of RC4 and AES encryption algorithm in. They used the technique described in [8] to compute power cost of encryption algorithms. Experiments show that RC4 is more suitable for large packets, whereas AES is suitable for small packets. One can save power by using a combination of RC4 and AES to provide encryption for any size packets, but the trade-off with security is not completely clear. Research in [9] used the technique in [8] and battery measurement to present a fair performance comparison for common encryption algorithms. Experimental results show performance of Blowfish is superior to other algorithms AES, DES, 3DES, RC2 and RC6.

Grobschadl *et al.* [10] evaluated both execution time and power consumption of block ciphers on StrongARM SA-1100 processor. They simulated the block ciphers by a cycle-accurate instruction set simulator Sim-Panalyser that constructed based on SimpleScalar [11]. They concluded that power consumption of block cipher primarily depends on the execution time.

Creighton *et al.* evaluated the energy cost of encryption algorithms in personal digital assistants [12]. They constructed a test system with Digit Multimeter to measure current power level under a certain workload. The measurement results indicate that all the ciphers consume similar amount of power at the same period.

Tiwari *et al.* [13] described a framework using instruction level power model to estimate energy consumption of encryption programs. The technique is able to apply to commercial microprocessors to verify whether an embedded design meets its energy constraints or guide development of embedded software.

We summarise encryption algorithm performance evaluation methods mentioned above in Table 1. It lists the used techniques and detailed evaluation items of performance. Although many performance evaluation methods are proposed, there is no literature dedicated to evaluate the effectiveness of encryption methods so far.

3 Encryption algorithms overview

In this section, we present a brief preliminary overview for the symmetric-key encryption algorithms to be evaluated. We select four types of symmetric algorithms Blowfish, CAST, RC5 and AES as examples for the evaluation. However, asymmetric algorithms may be evaluated as the same way.

Most of symmetric-key encryption algorithms using in practice has an iterative operation structure. An input block is encrypted by a few rounds of transformation applying sub-keys which are either computed in advance or computed by a key-schedule function in parallel. The round transformation and key-schedule function typically consist of a set of simple operations.

3.1 AES algorithm

The AES was originally called Rijndael, developed by Joan Daemen and Vincent Rijmen. AES is one of the variants (such as 3DES, AES etc.) of DES [14]. The performance of AES is significantly enhanced, which makes it available in many different encryption applications. AES is the first publicly accessible and open cipher approved by the NSA. It is based on a design principle known as a substitution–permutation network, which is fast in both software and hardware. AES does not use Feistel network. It has a fixed block size of 128 bits, and a key size of 128, 192 or 256 bits [15]. AES is secure so far, the only successful published attacks were side-channel attacks on some specific implementations.

3.2 Blowfish algorithm

Bruce Schneier designed Blowfish algorithm and made it freely used in the public application [3]. Blowfish combines a Feistel network, key-dependent *S*-boxes and a non-invertible *f*-function to create a secure algorithm. The only known attacks against Blowfish are based on its weak key classes [16]. Blowfish is a variable length key and 64-bit block cipher. The algorithm has not been cracked since it first introduced in 1993.

Table 1 Summary of power consumption methods

References	Technique	Performance evaluated			
		Runtime	Work load	Throughput	Power consumption
[6]	common imple.	√	—	—	—
[7–11]	soft profiling	—	—	—	—
[7, 9]	UPS battery	√	√	√	√
[12, 13]	ext. measurement	√	—	√	√

All the operations of Blowfish are XORs and additions on 32-bit data. The only additional operations are four indexed array data lookups on each round. Owing to the compact structure, Blowfish can be optimised effectively in use of both hardware and software applications.

3.3 CAST algorithm

CAST algorithm was designed in Canada by Carlisle Adams and Stafford Tavares [17, 18]. Its design is very similar to Blowfish, with key-dependent S-boxes, a non-invertible *f*-function, and a Feistel substitution–permutation network structure. David Wagner *et al.* had discovered a related-key attack on the 64-bit version of CAST. It requires approximately 217 chosen plaintexts, one related query and 248 offline computations. The strength of CAST algorithm mainly lies in the S-boxes. Many companies used CAST algorithm for their creditable security software application, but the chosen particular *S*-boxes are not published. CAST was patented by Entrust Technologies, later was generously released it for free use.

3.4 RC5 algorithm

RC5 is a group of algorithms designed by Rivest [19] and analysed by RSA Laboratories later. It can take on a variable block size, key size and number of rounds. The block size is generally dependent on the word size of used machine. David Wagner *et al.* have found weak keys in RC5 with the probability of selecting a weak key to be 2^{-10t}, where t is the number of rounds. For sufficiently large t values (greater than 10), this is not a problem as long as you are not trying to build a hash function based on RC5.

4 Power consumption evaluation methods

Power consumption is an important factor for performance of encryption algorithm, especially to applications in portable wireless devices constrained by battery. A lot of methods for power consumption evaluation have been proposed in the related literatures. It is mainly classified into software profiling, battery evaluation and external evaluation of three types of methods.

The first type of method namely software profiling is to employ software model to achieve power evaluation for encryption algorithm. Such model was originally implemented to validate correctness for a proposed design. Similarly, the power consumption of an encryption algorithm can also be evaluated via the tool constructed on the model.

Flinn and Satyanarayanan [19] proposed a tool named 'PowerScope' to profile the power consumption for applications. The tool maps power consumption to program structures, in much the same way that central processing unit (CPU) profiler maps processor cycles to specific processes and procedures. The method combines hardware instrumentation to measure current level with kernel software support to perform statistical sampling of system activities. 'PowerScope' can clearly determine the fractions of energy consumed during a certain period or that of individual procedures. They calculated the total power usage by integrating the product of instantaneous current and voltage over time. The value of power consumption was approximated by simultaneously sampling both current I_t and voltage V_t, at a regular interval of time Δt. The actual power over n samples was calculated by using a single measured voltage value V_{meas} to replace V_t. The calculation is shown in formula (1)

$$E \simeq V_{\text{meas}} \sum_{t=0}^{n} I_t \Delta t \tag{1}$$

Brooks *et al.* [21] provided 'Wattch', a power model based on cycle-level analysis. The model was integrated into an architectural simulator to estimate the power consumption of CPU. They evaluated the dynamic power consumption p_{d} as the main power consumption, as shown in formula (2). Parameters v_{dd} and $f$display supply voltage and clock frequency. The values are fixed for a certain CPU process technology. C is a load capacitance determined by circuit or transistor size that is estimated by the model. Parameter α displays an activity factor between 0 and 1 indicating how often ticks lead to switching activity on average. The activity factor is measured from the benchmark using an architectural simulator or directly assigned a value by assumption. They claimed that 'Wattch' was much faster and more precise than other existing tools

$$P_{\text{d}} = \text{CV}_{\text{dd}}^2 af \tag{2}$$

Ye *et al.* [22] presented an execution-driven and cycle-accurate RT-level power estimation tool, which is called 'SimplePower'. 'SimplePower' simulates executables providing cycle-by-cycle power estimate and switch capacitance statistics for the processor datapath, memory and on-chip bus. The total power consumed by a module is the sum of the power consumed by each bit transition.

Models of 'Wattch' and 'SimplePower' allow computer architects and designers consider the factor of power when making early-stage design decision. Nevertheless the inappropriate activity factor assignment may result in deviations of power consumption estimation.

Sinha and Chandrakasan [23] proposed a power estimation software tool called 'JouleTrack' with no need explicit instruction characterisations. They observed that the variation of current consumption for different instructions was small and that of programs was even smaller. They concluded that current consumption only depends on operating frequency and supply voltage yet independent to the program. Unfortunately, the application is restricted to the energy consumption of microprocessor. The power consumption of a subroutine executing on a microprocessor can be macroscopically represented as below

$$P_{\text{tot}} = P_{\text{dyn}} + P_{\text{stat}} = C_L V_{\text{dd}}^2 f + V_{\text{dd}} I_{\text{leak}} \tag{3}$$

where P_{tot} is the total power including both the static and the dynamic components, C_L is the total average capacitance being switched by executing program per clock cycle and f is the operating frequency.

Naik and Wei demonstrated the impact of software implementation on power saving in. They proposed several strategies for power saving. The experimental result showed that power saving achieved more than 60% by suitably choosing algorithms and applying the techniques.

The tool 'SimpleScalar' presented an infrastructure for simulation and architectural modelling [11]. The execution-driven simulation technique adopted in the model provides more powerful advantages compared with the trace-based technique. There were large differences in

doi: 10.1049/iet-sen.2012.0137

timing between its experimental results and actual measured values. The measurement method might be questioned, but it also shows that 'Simplescalar' should not be trusted as a clock-cycle accurate simulator for all types of architectures.

The second type of methods employs a strategy of power substitution of battery device. Such methods assume that the power consumption of a system in normal operating status is constant. One can measure the consumed battery percentage of an idle laptop system in a period so as to calculate the average power consumption per unit period. The average power consumption when system runs encryption algorithm can also be obtained in a similar way. The power consumed by encryption algorithm itself is the difference of the above power consumptions.

The third type of methods utilises instruments and electronic devices to construct a measurement system to achieve the goal. There are two ways, namely external measurement and onboard measurement.

Researchers use a mobile device like NOKIA N70 as reference platform to evaluate the power consumption of software [24, 25]. After finding out where the mobile phone obtains its power from, one contact is split up in such a way that they obtain two cables out of the phone. The cables are then used to measure the current taken out of the battery by using a simple multimeter or an advanced oscilloscope. One can also use a battery emulator as power supply to replace the battery. The emulator not only powers the mobile phone up, but also measures the power consumption in high accuracy. The electrical power E is calculated by multiplying voltage U, current I and time t. The most advantage of the method is that the supplied voltage is stable over all time. But it is impractical to measure the built-in component, like CPU or memory etc. Creus and Kuulusa [26] presented NOKIA S60 software profiling tools that allow developer to measure power consumption without any external equipment, namely onboard measurement. The measurement analysis is carried out either on a mobile device or on a PC.

Unfortunately, this kind of method is not suitable for power consumption evaluation of encryption algorithm. We once attempted to implement a cipher on the mobile device, yet it always led other applications or the mobile device itself down.

Creighton *et al.* [27] tried to measure the power consumption for an application on resource-limited devices. They conducted the power measurement for an HP iPAQ 4150 that connected in series between a benchtop power supply and an Agilent 3458A 8½ digit multimeter. The multimeter measured the voltage across the resistor 10 000 times per second. The resulting voltage multiplies the input voltage and divided by the resistance to calculate the power value, as shown in formula (4). Energy consumption for each encryption and decryption task (E_{task}) can be computed using (5), where n is the frequency of measurements and T is the execution time of encryption algorithm

$$P(t) = V(t) * V_{\text{input}}/R \tag{4}$$

$$E_{\text{task}} = \sum_{t=0}^{n} \left[P(t_i) - P_{\text{idle}}\right] * T \tag{5}$$

Bob *et al.* in Intel proposed a more accurate way to measure the power consumption for applications. The used data acquisition (DAQ) tool instruments some specific hardware components and logs a more granular power measurement [28]. As shown in Fig. 1, the measurement system is constructed by three components: a host PC, a NetDAQ and a target PC. The target PC has a special motherboard with built-in sensors. For each target component, all sense resistors are wired and soldered at both ends before being connected to a module attached to the NetDAQ unit. The NetDAQ measures the current and voltage drop of target PC across the sense resistors. The NetDAQ is also connected to the host PC via a cross-over network cable. It installs an IA32 system and a set of logger software on the host PC. The logger collects the measured data for the next calculation. The method significantly improved the accuracy of power evaluation because of the utilisation of precise instruments.

5 Experiment

In this section, we design a series of experiment to evaluate the power consumption of encryption algorithms by using three types of methods sequentially. Based on the experimental result, we assess the performance of power consumption evaluation methods.

For the first method, we use a workstation and a SANTAK-C3KS uninterruptible power system (UPS) supply to construct a measurement system. We calculate the power consumption by using the measured data. For the second method, we use external instruments to measure the consumed current value to calculate the power consumption. For the third method, we download an expert power profiling tool PowerScope [20] and evaluate the power consumption running on the same workstation.

We prepare all the programs using ANSI-C language. The programs are implemented on a Windows XP operating system on a workstation Fujitsu CELSIU S Series with Xeon™ CPU 3.60 GHz and 2.00 GB random access memory. In order to create a more fair comparison, all the algorithm programs are sorted into the same structure: 64-bit data input, 64-bit key size and 16-round computation. In practice, each encryption algorithm is constructed with its own characteristics for a specific purpose. In future

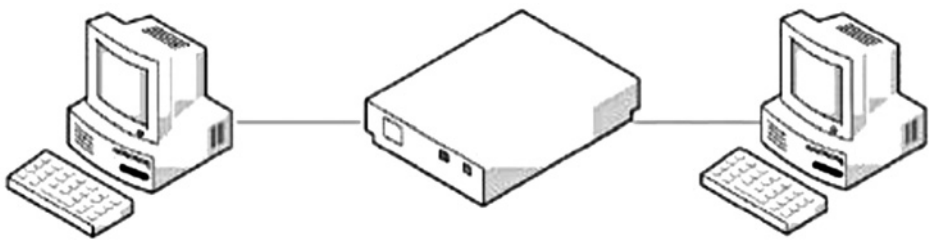

Fig. 1 *Power measurement setup (Intel)*

Table 2 Power consumption for block ciphers (method1)

Cipher	Power consumption (system idle), J/s	Total energy consumption, *J*	Total run time, s	Power consumption (cipher run), J/s	Power consumption (cipher only), J/s
CAST	6.40	10901.12	341.41	31.93	25.53
blowfish	6.40	7339.52	376.00	19.52	13.12
RC5	6.40	8791.23	341.40	25.75	19.35
AES	6.40	26102.20	703.10	37.12	30.72

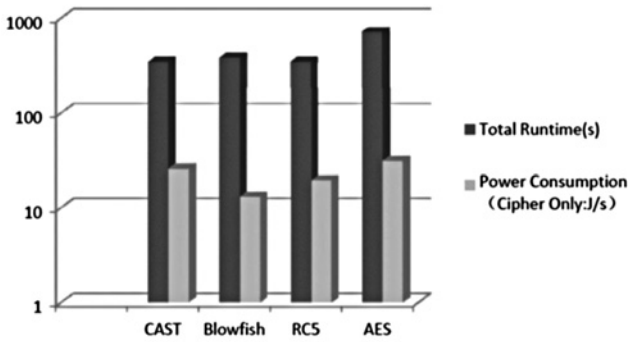

Fig. 2 *Runtime and power consumption (method 1)*

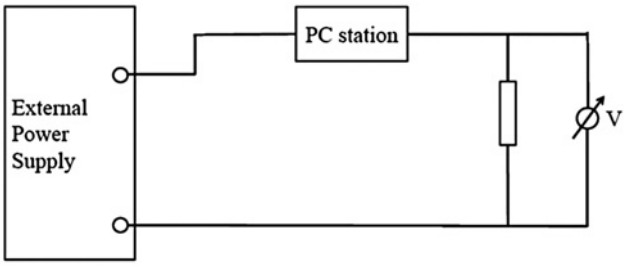

Fig. 3 *External power consumption measurement*

work, we should do more evaluation experiments for the variance of input data size and/or parameter size.

In the first method, we keep the system at idle status for a period of time without impressed power supply. We recorded the battery usage percentage of UPS and calculated the average power consumption in J/s by parameters of UPS. We charged the battery to reach 100% level, and run a cipher algorithm for one hundred million times on the workstation. We calculated the average power consumption in J/s by the same way. We repeated the procedure until the power consumption of all the algorithms was obtained.

Table 4 Power consumption for block ciphers (method 3)

Cipher	Runtime, s	Memory utilisation, kB	Power consumption, J/s
CAST	85.35	1497.00	15.63
Blowfish	94.00	937.00	8.60
RC5	85.35	976.00	14.60
AES	1732.60	1722.00	19.56

The experimental results are shown in Table 2. The second column shows the average power consumption when system is idle. The value for each algorithm is constant. The third and the fourth columns display the total power consumption and runtime for 100M-time encryption computation. The fifth column displays the average power consumption when the system runs a cipher. We calculated the average power consumed by the cipher itself, which is shown in the sixth column. From the table, we can see that the Blowfish algorithm runs fastest and consumes least power, then RC5, CAST and AES. Blowfish consumes only 13.12 J/s, account for 42.71% of about AES. It is clear that the reason that Blowfish consumes less power is because it runs much faster, so that utilises computer resource for a shorter time.

The histogram in Fig. 2 illustrates intuitively the relationship between runtime and power consumption of four block ciphers. The black histogram displays runtime, whereas the grey one displays power consumption. The *y*-coordinate uses a log scale in order to make the difference look more clearly. It shows that power consumption is roughly linear with runtime. The Blowfish cipher is more power-effective than other algorithms.

In the second method, we designed an external power consumption measurement system as shown in Fig. 3. The system is constructed by a power supply, a resistance, a PC station and a multimeter. The output voltage of the used

Table 3 Power consumption for block ciphers (Method 2)

Cipher	Measurement	Run time, s	Power consumption, J/s	Average power consumption, J/s
CAST	1st	300	16.26	16.12
	2nd	300	16.02	
	3rd	300	16.08	
Blowfish	1st	300	8.96	9.02
	2nd	300	9.08	
	3rd	300	9.01	
RC5	1st	300	15.30	15.20
	2nd	300	15.12	
	3rd	300	15.19	
AES	1st	300	20.26	20.19
	2nd	300	20.21	
	3rd	300	20.10	

doi: 10.1049/iet-sen.2012.0137

Table 5 Comparison of the methods

Cipher	UPS battery	External	Improv., %	Soft, (Powerscope)	Improv., %
CAST	25.53	16.12	36.86	15.63	38.78
Blowfish	13.12	9.02	31.25	8.60	34.45
RC5	19.35	15.20	21.45	14.60	24.55
AES	30.72	20.19	20.19	19.56	36.33
average			27.44		33.53

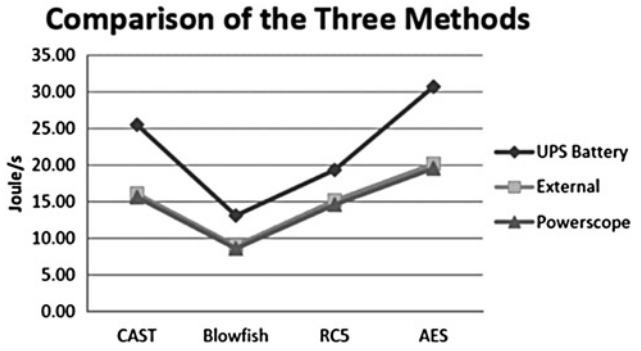

Fig. 4 *Comparison of the methods*

external power supply is constant. As we know, the power consumption usually varies along with the difference of supply impedance. To alleviate such impacts as much as possible, a very low resistance (0.02 Ω) precision resistor is placed between the external power supply and the PC station. An Agilent 3458A digit multimeter is used to measure the voltage across the resistor 10 000 times per second. The resulting voltage measurements were multiplied by input voltage and divided by resistance to calculate the power. Energy consumption of each encryption algorithm (E) is computed by using (6), where V_{ext} is the external power supply, V_t and V'_t are the varying voltages through the PC station, n is the measure times and T is the execution time of encryption algorithm

$$E = V_{\text{ext}} \sum_{t=0}^{n} \left[\left(V_t - V'_t \right) / R \right] \times T \tag{6}$$

The experimental results of the method are shown in Table 3. We evaluated each cipher for three times and each time for 300 s. The average power consumption of the ciphers is 16.12, 9.02, 15.20 and 20.19 J/s. Blowfish algorithm still acts more effectively than any other algorithms.

At last, we downloaded the expert power profiling tool PowerScope [20]. We use the workstation as the profiling platform. We evaluated the power consumption and memory utilisation for each cipher. The experimental results are shown in Table 4. Similarly, power consumption of Blowfish is the lowest, then RC5, CAST and AES. Notably, Blowfish consumes less power while utilising less memory.

We compare the performance of three evaluation methods and assess their effectiveness. The results are shown in Table 5. It indicates that the methods of external measurement and software profiling are more accurate than the method of UPS battery. The improvements are 27.44 and 33.53% separately. Fig. 4 summarises the experimental results of the three methods. We can see the curve of external measurement method is very close to Powerscope. The curve of UPS battery method is on top of the figure, which implies the method of UPS battery is rougher than other two methods.

From the experimental results, we know that the power consumed by a cipher corresponds to the average power dissipation and the total runtime. The former depends on a number of factors including supply voltage, clock frequency, and the average current drawn by the processor. By the effectiveness analysis for evaluation methods, external measurement and software profiling method are more accurate than UPS battery method. Moreover, software profiling may be more convenient than external measurement mechanism if it provides a good performance indication.

6 Conclusion

In modern information application, data encryption algorithm plays an important role for the security of a network. The performance research particularly the power consumption for encryption algorithms is becoming an important concern. In this paper, we have reviewed several popularly used symmetric encryption algorithms. We introduced three popular power consumption evaluation methods for encryption algorithm. Experiments are designed to evaluate effectiveness of the methods. The experimental result shows that Blowfish algorithm is the most power-effective encryption algorithm, then RC5, CAST and AES. We demonstrate that methods of external measurement and software profiling are more accurate than the method of UPS battery. The work can guide software designers or hardware manufacturers to reconstruct a power-effective security system.

7 Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant 61171150. The Project is sponsored by SRF for ROCS, SEM. and supported by Shandong Province Natural Science Foundation (grant no. ZR2009GL007).

8 References

1 Bruce, S.: 'Applied cryptography: protocols, algorithms, and source code in C' (Wiley, 1996)
2 Coppersmith, D.: 'The data encryption standard (DES) and its strength against attacks', *IBM J. Res. Dev.*, 1994, **38**, (3), pp. 243–250
3 The Blowfish Encryption Algorithm. Available at http://www.schneier.com/blowfish.html/, 2008
4 Diaa Salama, A.E., Hatem Mohamed, A.K., Mohie Mohamed, H.: 'Performance Evaluation of Symmetric Encryption Algorithms', *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, 2008, **8**, (12), pp. 280–286
5 Dolev, D., Yao, A.: 'On the security of public key protocols', *IEEE Trans. Inf. Theory*, 1983, **29**, (2), pp. 198–208
6 Nadeem, A.: 'A performance comparison of data encryption algorithms'. Proc. IEEE Information and Communication Technologies, Damascus, Syria, April 2006, pp. 84–89

7 Prasithsangaree, P., Krishnamurthy, P.: 'Analysis of power consumption of RC4 and AES algorithms in wireless LANs'. Proc. Globecom'03, 2003, pp. 1445–1449
8 Naik, K., Wei, D.: 'Software implementation strategies for power-conscious systems', *Mob. Netw. Appl.*, 2001, **6**, (3), pp. 291–305
9 Diaa Salama, A.E., Hatem Mohamed, A.K., Mohie, M.H.: 'Power efficiency of encryption schemes for wireless devices', *J. Comput. Theory Eng.*, 2009, **1**, (3), pp. 1793–8201
10 Grobschadl, J., Tillich, S., Rechberger, C.: 'Energy evaluation of software implementations of block ciphers under memory constraints'. Proc. Tenth Conf. Design, Automation and Test in Europe, 2007, pp. 1110–1115
11 Austin, T., Larson, E., Ernst, D.: 'SimpleScalar: an infrastructure for computer system modeling', *IEEE J. Mag.*, 2002, **35**, (2), pp. 59–67
12 Hager, C.T.R., Midkiff, S.F., Park, J.M., Martin, T.L.: 'Performance and energy efficiency of block ciphers in personal digital assistants'. Proc. Third IEEE Int. Conf. Pervasive Computing and Communications, March 2005, pp. 127–136
13 Tiwari, V., Malik, S., Wolfe, A.: 'Power analysis of embedded software: a first step toward software power minimization', *IEEE Trans. VLSI Syst.*, 1994, **2**, (4), pp. 437–445
14 William, C.B.: 'Recommendation for the triple data encryption algorithm (TDEA) Block cipher' (NIST Special Publication 800–67 Version 1.1, 2008)
15 Daemen, J., Rijmen, V.: 'The design of Rijndael: the advanced encryption standard', Springer, 2002
16 Vaudenay, S.: 'On the weak keys in blowfish, fast software encryption'. Proc. Third Int. Workshop, 1996, pp. 27–32
17 Adams, C.M., Tavares, S.E.: 'Designing *S*-boxes for ciphers resistant to differential cryptanalysis'. Proc. Third Symp. State and Progress of Research in Cryptography, Italy, February 1993, pp. 181–190
18 Adams, C.M.: 'Simple and effective key scheduling for symmetric ciphers'. Proc. Workshop on Selected Areas in Cryptography-Workshop Record, Kingston, Ontario, May 1994, pp. 129–133
19 Rivest, R.L.: 'The RC5 encryption algorithm'. Proc. First Int. Workshop on Fast Software Encryption', 1994, pp. 86–96
20 Flinn, J., Satyanarayanan, M.: 'PowerScope: a tool for profiling the power usage of mobile applications'. Proc. Second IEEE Workshop on Mobile Computer Systems and Applications, February 1999, pp. 2–10
21 Brooks, D., Tiwari, V., Martonosi, M.: 'Wattch: a framework for architectural-level power analysis and optimizations'. Proc. 27th Annual Int. Symp. Computer Architecture, June 2000, pp. 83–94
22 Ye, W., Vijaykrishna, N., Kandemir, M., Irwin, M.J.: 'The design and use of simplepower: a cycle-accurate power estimation tool'. Proc. 37th Design Automation Conf. (DAC), June 2000, pp. 340–345
23 Sinha, A., Chandrakasan, A.: 'JouleTrack a web based tool for software power profiling'. Proc. 38th Conf. Design Automation, 2001, pp. 220–225
24 Fitzek, F.H.P., Reichert, F.: 'Mobile phone programming and its application to wireless networking' (Springer, 2007)
25 Nokia Power Profiler. Available at www.forum.nokia.com/powerprofiler/, 2012
26 Creus, G., Kuulusa, M.: 'Mobile phone programming, optimizing mobile software with built-in power profiling' (Springer, Netherlands, 2007)
27 Creighton, T.R.H., Scott, F.M., Jung-Min, P., Thomas, L.M.: 'Performance and power efficiency of block ciphers in personal digital assistants'. Proc. Third IEEE Int. Conf. Pervasive Computing and Communications, 2005, pp. 127–136
28 Bob, S., Rajshree, C., Karthik, K., Jun, D.V.: Creating Power Efficient Software. Available at http://www.download.intel.com/software/pdf/CreatingPowerEfficientSoftware.pdf/, 2007

doi: 10.1049/iet-sen.2012.0137