Battery Power Optimized Encryption

Satish Bapatla and R. Chandramouli

Multimedia System, Networking, and Communications (MSyNC) Laboratory
Department of Electrical and Computer Engineering
Stevens Institute of Technology

***Abstract*— Computational power optimization is crucial in battery power limited secure wireless mobile networks. Therefore, in this paper, we (a) introduce a hardware/software set-up to measure and model the battery power consumption of different encryption algorithms through real-life experimentation and (b) compute optimal power (number of rounds) allocation for encrypting packets such that constraints on power and security are met. We present results for three block ciphers: DES, IDEA, and GOST though the same analysis can be extended to other ciphers such as AES, RC4, etc. A new measure called "vulnerability" that quantifies the success of linear cryptanalysis attack is proposed and its relationship with the power consumption is explored. Two mathematical optimization problems are then posed: (a) compute the optimal power allocation to encrypt each packet such that vulnerability is minimized subject to a total power constraint and (b) compute the optimal number of encryption rounds for each packet such that a total power constraint is met. The differences in these two formulations are presented. A closed form solution to the the first problem is derived while the second optimization formulation is posed as an integer program and solved numerically. Several numerical results are also provided.**

I. Introduction

Security of wireless networks is a major issue currently. Battery power limited mobile wireless communications pose numerous new research and development challenges, including power efficient operation, low cost, small size, error and fault resilience, flexibility, security and privacy. Due to the persistent limitations in current battery technologies, most often the communicating mobile nodes must operate on an extremely frugal power budget. In addition to the stringent power constraint, information security is also a key performance factor in sensor and ad hoc network networks. Example applications where power and security play a major role include battlefield communications, infrastructure security and surveillance such as airports and hospitals, etc.

Battery power conservation is especially important in wireless sensor and ad hoc networks [1], [2]. The primary challenge in providing security in low power wireless networks lies in the conflicting interest between minimizing power consumption and maximizing security. In general, we can safely assume that by doing more computations one can achieve a higher amount of security. For example, the strength of encryption schemes depend on the size of the key and the number of encryption rounds. Larger key sizes/rounds produce higher levels of security at the cost of additional power consumption. As seen in Fig. 1 from our experiments for DES encryption, there is a trade-off between the security vulnerability (a measure to be defined later) and the battery power consumption. Therefore, in order to design power efficient secure protocols for wireless networks there is an inherent need to understand the relationships between power consumption and encryption parameters. Once these relationships are understood well, then it is possible to optimize power consumption w.r.t. a security requirement or vice-versa. For instance, since different data types may need different levels of security [3], we can pose the problem of security maximization w.r.t. a total power budget as a constrained optimization problem over the different data types.

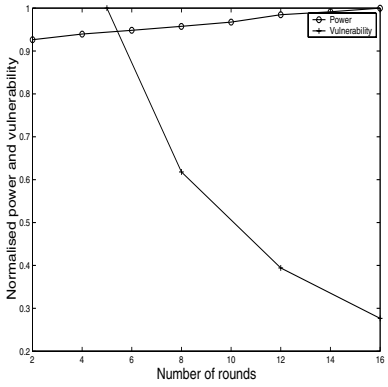

Fig. 1. Security vulnerability and battery power consumption for different DES encryption rounds.

The goals of this paper are two fold. First, we mathematically model the relationship between power consumption and encryption based security algorithms/parameters which is then followed by techniques to optimally trade them for each other. Specifically, we consider DES, IDEA, and GOST encryption algorithms and use the source code given in [4] for our analyses and experiments.

The paper is organized as follows. We present a brief overview of the different block encryption schemes in Section II, the experimental setup to collect data that captures the battery power consumption for different encryption algorithms and their parameters and the corresponding mathematical models for power consumption are discussed in Section III. Section IV describes the mathematical optimization formulations of security under power constraint and presents some

numerical results. This is followed by concluding remarks in Section V.

II. Overview of Various Block Encryption Schemes

In this section we provide only a brief overview of the various blcok encryption schemes we have considered in this paper, namely, DES, IDEA, and GOST. Further details can be found in many standard books on encryption such as [4].

A. DES

DES encrypts data in 64-bit blocks. A 64-bit block of plain text is taken as an input by the algorithm and a 64 bit block of cipher is produced as an output. DES is a symmetric algorithm [4]. Therefore, the computations performed by the encryption and decryption algorithms are nearly the same. The encryption key length is 56 bits long. The algorithm is a combination of two basic techniques of encryption: *confusion and diffusion.* The fundamental building block of DES is a single combination of basic techniques on the plain text, based on the key. This is known as a round. DES has 16 rounds.

B. IDEA

IDEA operates on 64 bit plain text blocks. The key is 128 bits long. The same algorithm is used for both encryption and decryption. The design policy behind the algorithm is one of mixing operations from different algebraic groups. Three algebraic groups are being mixed, they are XOR, addition modulo 2^{16}, and multiplication modulo 2^{16}+1. All these operations operate on 16 bit sub-blocks. In total IDEA has 8 rounds.

C. GOST

GOST is a 64-bit algorithm with a 256 bit key. The 256 bit key is divided into eight 32- bit blocks. Each round has a different sub key. The algorithm iterates a simple encryption algorithm for 32 rounds. There are eight different S-boxes in GOST [4]. Decryption is same as encryption. The GOST standard does not seem to discuss the S-box is generated. Therefore, including the secret S-box permutations, GOST has a total of about 610 bits of secret information.

III. Power vs. Security: Experimental Measurement Set-up and Modelling

The experimental set-up for measuring power consumption of encryption algorithms consists of a Sony Vaio laptop with a 700 Mhz P-III processor and 128 MB RAM running Red Hat Linux 2.4.8 chosen for its open source nature. The power consumed by the CPU in running the encryption algorithms is measured as a function of input power supply to the Laptop. A separate DC power supply is given to the laptop to permit measurements. The battery of the laptop is removed for accuracy in measurements. The current measurements are gathered using Labview software [5] from the GPIB interface of the power supply. In order to eliminate effects of the other jobs that could be running in the background the current consumption is first measured when no other tasks are running (idle amps). The difference in currents when an encryption algorithm is running and idle amps is taken as the actual current consumption during encryption. In the experiments, since voltage variation is seen to be extremely small (measured at less than 0.25%) we use a constant value given by the manufacturer. Power consumption value is then computed as the product of the voltage and the current consumption. Several experiments with different data sets were conducted and the average of these results is calculated as the final (average) power consumption value.

To monitor each the power consumption due to software component of an implementation we use OProfile [6]. OProfile is a system-wide profiler for Linux systems capable of profiling all running code at low overhead. OProfile leverages the hardware performance counters of the CPU to enable profiling of a wide variety of interesting statistics which can also be used for basic time-spent profiling. All code is profiled: hardware and software interrupt handlers, kernel modules, the kernel, shared libraries, and applications. So we have adapted OProfile to monitor the different components of an encryption algorithm in order to measure the power values for the different functions involved. Each encryption algorithm was divided into two portions: *setup functions* that initialize the key elements that would be used in encryption/decryption and *core functions* which repeatedly perform operations on data block.

For DES the function which involves both data expander function and S-box substitution function takes almost 75% of total execution time. Similarly for IDEA and GOST the core functions take more time. We know that core functions have to be carried for every round. Figure 2 shows a comparison of consumed power for encrypting using DES, IDEA and GOST. From Figure 2 we conclude that the key length and the number of rounds, though play an important role in the total power consumption, for these three block ciphers, for the key length and number of rounds fixed as given above, the difference in power consumption does not seem to be significant.

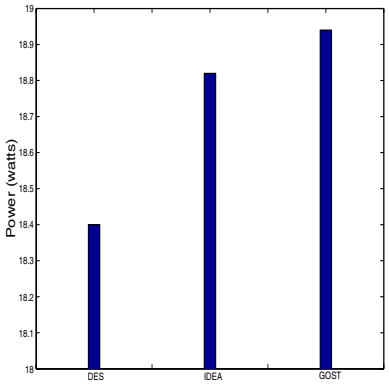

Fig. 2. Comparison of different block ciphers.

Figure 3 shows the variation of consumed power for differ-

ent rounds of DES, IDEA and GOST. We observe from these figures that power varies linearly with the number of rounds. In all these experiments, some data points were used to obtain a mathematical model (training data) and then the validity of the model was testing for other parameter values (testing data). Next we compute mathematical regression models that

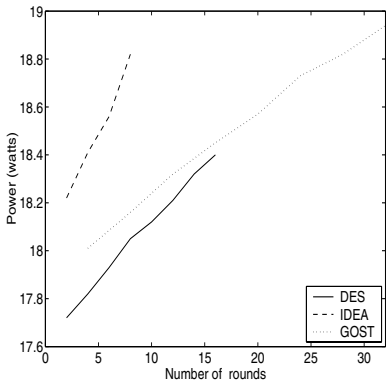

Fig. 3. Power consumption for different rounds of DES,IDEA and GOST .

capture the relationship between the power consumption and encryption parameters. Let P denote the consumed power (in Watts) and r the number of encryption rounds, respectively. Then using statistical regression, for DES we find that:

$$P(r) = 0.0486r + 17.7335 \tag{1}$$

The standard of error of this model is .0139 meaning the curve fit is a good approximation of the actual behavior. Similarly, for IDEA we find that,

$$P(r) = 0.0975r + 18.015 \tag{2}$$

with a standard of error equal to 0.03427. For GOST, it is,

$$P(r) = 0.03321r + 17.90204 \tag{3}$$

with standard of error equal to 0.0450. *From these models we see that GOST has the smallest slope implying the rate of change of power w.r.t. the number of rounds is the smallest for GOST.* Given these mathematical models for $P(r)$, the next step is to optimize the trade-off between power versus security as discussed in the next section.

IV. Security Optimization With Power Constraint

Symmetric block ciphers are popular in several applications. This popularity requires a high level of trust in their security. Unfortunately there are neither known constructions of block ciphers, which offer unconditional security nor practical constructions, which offer provable computational security. One way to measure the effectiveness of a cryptanalysis attack is to compare its complexity with the exhaustive search attack. There are mainly three types of attacks on encryption algorithms, they are: (a) brute-force attack, (b) differential cryptanalysis and (c) linear cryptanalysis. In a brute-force attack all possible encryption keys are successively tested. Linear cryptanalysis is an attempt to find linear dependency of high probability between the plain text, the cipher text and the key, by which the key may be retrieved. The DES algorithm is vulnerable to linear cryptanalysis attacks. By such an attack, the algorithm in its sixteen rounds can be broken using 2^{47} known plain texts [7]. This vulnerability raises a notable risk when encrypting bulk data that may be predictable with keys that are constant.

Note that the number of rounds and key length impact the total power consumption and the security against successful cryptanalysis attacks. Therefore, it is possible to find a mathematical relationship between power consumption and the offered security. We first consider linear cryptanalysis attack of DES for the sake of illustration. Since all the operations in DES except the S-boxes are linear it suffices to derive linear relations of the S-boxes. These relations are derived for each S-box by choosing a subset of input bits and output bits, calculating parity of these bits for each of possible inputs of S-box and counting the number of inputs whose subset parity is zero. As the number of zeros is closer to number of ones we will say that subset is more nonlinear. We have to find a statistical linear expression consisting of parity of subsets of the plain text, cipher text and the key which is derived from similar expressions of various rounds. Thus, the parity of some set of data bits in each round is known as a function of the parity of the previous set of bits in the previous round and parity of several key bits. The round linearization is based on the linearization of S-boxes.

Let $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{K}$ stand for the plain text, cipher text and the key vector, respectively. Then, following [7], the purpose of linear cryptanalysis is to find the following effective linear expression for an encryption algorithm:

$$\mathcal{P}[i_1, i_2, ..., i_a] \bigoplus \mathcal{C}[j_1, j_2, ..., j_b] = \mathcal{K}[k_1, k_2, ...k_c] \tag{4}$$

where i_1,i_2,.....,i_a, j_1,j_2,.....,j_b and k_1,k_2,.....k_c denote fixed bit locations. This equation holds with probability not equal to $1/2$ for a random plain text $\mathcal{P}$ and the corresponding cipher text $\mathcal{C}$. Then, the value of $|p - 1/2|$ represents the effectiveness of linear expression in Eq. (4). Let N be the number of given random plain texts and p be the probability that the Eq. (4) holds. If $|p - 1/2|$ is sufficiently small then the linear cryptanalysis success rate clearly increases with N or $|p - 1/2|$.For an 8-round DES, DES key is breakable with 2^{21} known plain texts, 12-round DES in 2^{33} known plain texts and a 16-round DES with 2^{47} plain texts [7].

Now we see that the success probability of a known plain text linear cryptanalysis attack can be computed as a function of the number of rounds. Therefore, we define a measure called the *vulnerability* as follows.

Definition 1: Vulnerability is defined as the ratio of maximum total number of plain texts (for a given block length) to the number of plain texts required to successfully estimate the encryption key.

For example, if we take an 8 round DES algorithm it requires a total of 2^{21} plain texts to estimate the key. If we take a 64 bit block then there are 2^{64} possible plain texts. Therefore, in this case vulnerability is given by 2^{64} /2^{21} = 2^{43}. This number indicates the reduction factor in the number of required plain texts for a successful cryptanalysis attack when compared to a brute force attack.

Figure 4 shows the vulnerability curve for different number of DES rounds. Y-axis shows the $log_2(.)$ value of the vulnerability. Clearly, the vulnerability decreases as the number of

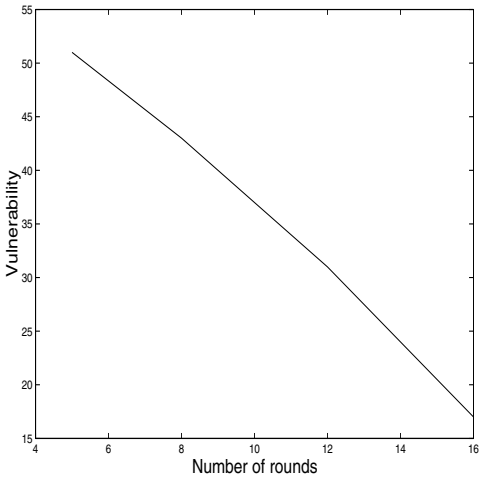

Fig. 4. Logarithm of vulnerability for different number of DES rounds.

DES rounds increases.

A. Vulnerability Minimization with Power Constraint

Suppose we have M data packets (or class of packets) and that all the packets are not equally important in terms of security (or vulnerability) requirement. This occurs in several applications. For instance, in video applications, the motion vector packets need to be more secure than the packets containing texture information. The question then is: *How do we minimize the total vulnerability subject to a total power constraint, say, P_t?* A simple strategy is to allocate power P_t/M to each packet. But, this may not be the optimal strategy if the vulnerability requirement is not equal. Therefore, we formulate the following constrained optimization problem:

1) Optimization Formulation 1: The objective is to optimally allocate the power resources to M packets with different vulnerability (security) requirement such that the total power budget is not exceeded. Mathematically, this is problem is given by,

$$\min_{\{P_1,P_2,\ldots p_M\}} \sum_{k=1}^{M} w_k V_k \quad \text{s.t.} \quad \sum_{k=1}^{M} P_k \leq P_t \tag{5}$$

where V_k stands for vulnerability of packet k, $k = 1, 2, \ldots, M$, $0 \leq w_k \leq 1$ is a weighting parameter, and P_k is the power allocated to encrypt the kth packet. Note that a higher value of w_k implies a higher security requirement for that packet. Then, the optimal power allocation is given by the following theorem.

Theorem 1: The optimal power allocation solution (to Eq. (5)) that minimizes the total weighted vulnerability for a given P_t is given by

$$P_k = 1 + \frac{(P_t - M)\prod_{i=1,i\neq k}^{M} w_i}{\sum_{i=1}^{M}\prod_{j=1,j\neq i}^{M} w_i}, k = 1, 2, \ldots M.$$

Proof sketch: Using the Langrange multiplier formulation, the cost function for the optimization formulation is,

$$J(P_1, P_2, \ldots, P_M) = \sum_{k=1}^{M} w_k V_k + \lambda(\sum_{k=1}^{M} P_k - P_t) \tag{6}$$

Then by solving the following set of equations,

$$\frac{\delta J(.)}{\delta P_k} = 0, k = 1, 2, \ldots, M; \frac{\delta J(.)}{\delta \lambda} = 0 \tag{7}$$

we get the result. The second derivative test shows that this power allocation indeed minimizes the cost function.

2) Optimization Formulation 2: In the above formulation, V_k can be computed for linear cryptanalysis of DES. But, for a general cipher, we need a more generalized formulation. Let r_k denotes the number of rounds of encryption used for packet k, $k = 1, 2, \ldots, M$. Let $0 \leq w_k \leq 1$ be a weighting parameter and P_t be the total power constraint (a higher value of w_k implies a higher security requirement). Then, P_t can be converted into an equivalent upper bound constraint on the total number of rounds R_t using the linear relationships derived in previous sections. Since the security of the three block ciphers discussed here increase with the number of rounds, the power constrained encryption formulation is now given by the following integer program,

$$\max_{\{r_1,r_2,\ldots r_M\}} \sum_{k=1}^{M} w_k r_k \quad \text{s.t.} \quad \sum_{k=1}^{M} r_k \leq R_t; l_k \leq r_k \leq u_k \tag{8}$$

where l_k and u_k are the lower and upper bounds on the number of rounds for packet k that are user fixable within the constraints of the encryption algorithm. A higher value of l_k would mean a higher security for that packet. In this formulation, we attempt to encrypt each packet with the number of rounds allowable within the total round constraint.

The integer programming problem in Eq. (8) was solved numerically using cplex [8]. Figure 5 shows the pre-fixed security weighting factor for each packet and the corresponding optimal number of rounds of DES encryption and power consumption output as a solution to the integer program formulation. It is clear from this figure that the larger the weight, the higher the number of rounds allocated to that packet. Here, the $l_k = 1$ and $u_k = 16$ for every k.

We can perform the same optimization by changing the lower bounds l_k depending on the importance of the packet. Changing l_k's would produce more significant changes in r_k's compared to keeping them fixed as shown in Figure 6.

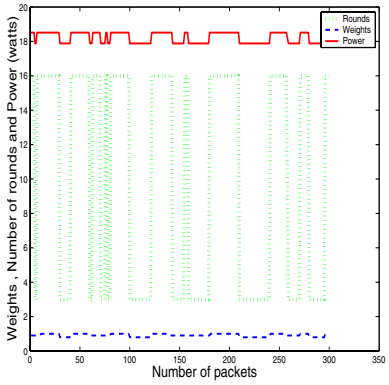

Fig. 5. Optimal allocation of number of DES encryption rounds (and power) for fixed weighting values.

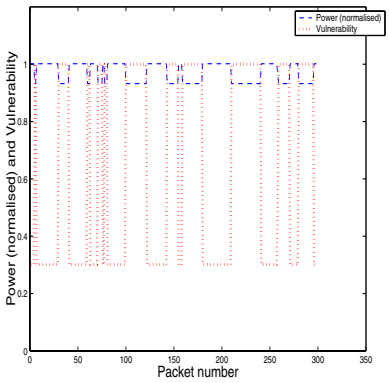

Fig. 7. Vulnerability and power for different security weights.

Similarly solutions can be obtained for IDEA and GOST as well.

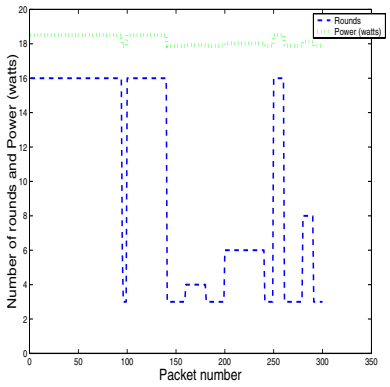

Fig. 6. Optimal allocation of number of DES encryption rounds and power for different lower bounds, l_k.

Using Eq. (5) if we solve for the vulnerability and (normalized) power using the weights used above, we get Figure 7. Clearly we can see that as the power allocation increases vulnerability decreases.

V. Conclusions

We provided an experimental set-up to measure power consumption of encryption algorithms for mobile applications. The data collected from this set-up is used to mathematically model the relationship between power consumption and security of three block ciphers: DES, IDEA and GOST. It is seen that power consumption changes linearly with the number of rounds of these encryption algorithms. GOST has the smallest rate of increase of power consumption.

A new measure called *vulnerability* is proposed based on linear cryptanalysis of DES. Minimizing vulnerability subject to a total power constraint is solved and a closed form solution to the optimal power allocation for different packets subject to different security constraints is derived. This formulation is then extended to general ciphers as an integer programming problem. Optimal number of rounds of encryption for each packet such that the total weighted number of rounds is maximized subject to a total power constraint is also solved. The solution shows that the optimal number of rounds of encryption could vary significantly depending on the power and security constraint. Two communicating parties can employ the proposed optimization algorithms by exchanging constraints and parameters during session negotiation.

Acknowledgements

This work was partially supported by NSF DAS 0242417 and NSF CAREER 0133761.

References

[1] S.Slijepcevic, M.Potkonjak, V.Tsiatsis, S.Zimbeck, and M. Srivastava, "On communication security in wireless ad-hoc sensor networks," *11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.*, pp. 139–144, 2002.

[2] J.Feng and M. Potkonjak, "Power minimization by separation of control and data radios," *IEEE CAS Workshop on Wireless Communication and Networking*, pp. 112–121, 2002.

[3] J.Goodman and A.P.Chandrakasan, "Low power scalable encryption algorithms," *Wireless Networks The Journal of Mobile Communication, Computation and Information*, vol. 4, no. 1, pp. 55–70, 1998.

[4] B.Schneier, *Applied Cryptography : protocols,algorithms,and source code in C*. John Wiley and Sons, 2002, vol. 2.

[5] "Labview," *http://www.ni.com*.

[6] "Oprofile a system profiler for linux," *http://oprofile.sourceforge.net*.

[7] M.Matsui, "Linear cryptanalysis method for des cipher," *Advances in Cryptology - Eurocrypt*, pp. 386–399, 1993.

[8] "Cplex," *http://www.cplex.com*.