INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

With Emphasis on the Integration of Three Technologies

IJCCC
A Quarterly Journal

Year: 2008 Volume: III Number: 4 (December)

Agora University Editing House

CCC Publications

Licensed partner: EBSCO Publishing

www.journal.univagora.ro

Publisher & Editorial Office
CCC Publications, Agora University
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel: +40 259 427 398, Fax: +40 259 434 925, E-mail: ccc@univagora.ro
Website: www.journal.univagora.ro
ISSN 1841-9836, E-ISSN 1841-9844

International Journal of Computers, Communications and Control (IJCCC) is published from 2006 and has 4 issues/year (March, June, September, December), print & online.

Founders of IJCCC: I. Dziţac, F.G. Filip and M.J. Manolescu (2006)

This publication is subsidized by:

1. Agora University
2. The Romanian Ministry of Education and Research / The National Authority for Scientific Research

CCC Publications, powered by Agora University Publishing House, currently publishes the "International Journal of Computers, Communications & Control" and its scope is to publish scientific literature (journals, books, monographs and conference proceedings) in the field of Computers, Communications and Control.

IJCCC is indexed and abstracted in a number of databases and services including:

1. ISI Thomson Reuters - Science Citation Index Expanded (also known as SciSearch®)
2. ISI Thomson Reuters - Journal Citation Reports/Science Edition
3. ISI Thomson Reuters - Master Journal List
4. SCOPUS
5. Computer & Applied Science Complete (EBSCO)
6. Vocational Studies Complete (EBSCO)
7. Current Abstracts (EBSCO)
8. Collection of Computer Science Bibliographies(CCSB)
9. Informatics portal io-port.net (FIZ KARLSRUHE)
10. MathSciNet
11. Open J-Gate
12. Google Scholar
13. Romanian Journals CNCSIS (cat.A, cod 849)
14. Information Systems Journals (ISJ)
15. Urlich's Periodicals Directory
16. Genamics JournalSeek
17. ISJ- Journal Popularity
18. Magistri et Scholares
19. SCIRUS
20. DOAJ

Contents

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 324-335

On DPA-Resistive Implementation of FSR-based Stream Ciphers using SABL Logic Styles

Reza Ebrahimi Atani, Sattar Mirzakuchaki, Shahabaddin Ebrahimi Atani, Willi Meier

Abstract:
The threat of DPA attacks is of crucial importance when designing cryptographic hardware. This contribution discusses the DPA-resistant implementation of two eSTREAM finalists using SABL logic styles. Particularly, two Feedback Shift Register (FSR) based stream ciphers, Grain v.1 and Trivium are designed in both BSim3 130nm and typical 350nm technologies and simulated by HSpice software. Circuit simulations and statistical power analysis show that DPA resistivity of SABL implementation of both stream ciphers has a major improvement. The paper presents the tradeoffs involved in the circuit design and the design for performance issues.

Keywords: DPA attack, Stream cipher, Grain v.1, Trivium, SABL, Standard CMOS.

1 Introduction

The term of security for a cryptographic primitive can be considered from two points of view: mathematical security (resistance against classical cryptanalysis) and the second one is physical security. Physical attacks on cryptographic devices take advantage of implementation-specific characteristics to recover the secret parameters. They are therefore much less general since they are specific to a given implementation but often much more powerful than classical cryptanalysis, and are considered very seriously by cryptographic devices implementors. A side-channel attack occurs when an attacker is able to use some additional information leaked from the implementation of a cryptographic function to cryptanalyze the function. Clearly, given enough side-channel information, it is trivial to break a cipher. One side channel attack in particular, namely the differential power analysis (DPA) is of great concern. It was first reported by Kocher et al. in 1998 that the power consumption of a smart card could reveal the secret key of the cryptographic algorithm [1]. DPA is a well-known and thoroughly studied threat for implementations of block ciphers (DES and AES), public key algorithms (RSA) and recently stream ciphers (Grain and Trivium [4]).

Stream ciphers as part of the symmetric key cryptography family, have always had the reputation of efficiency in hardware and speed. They have attracted much attention since the beginning of the eSTREAM project in 2004. Although there is vast literature about DPA on implementations of block ciphers and public key algorithms, only few publications can be found about DPA attacks on stream ciphers ([2], [3], [4], [8], [13], [14]).

In power analysis attacks, it is assumed that the power consumption of a circuit is correlated to the data handled. An attacker can therefore recover secret information by simply monitoring the power signals of a running device.

Stream ciphers require frequent synchronization to prevent synchronization loss between sender and receiver. Normally the initialization will be done with the same secret key and with a different initial value IV. So an attacker can disrupt the synchronization and apply a new known IV and measure the power traces in the initialization phase to apply a DPA on the embedded system of the stream cipher. So far, there is only one report on a practical DPA targeting hardware implementations of stream ciphers [4]. In that paper, a chosen IV DPA attack on Grain and Trivium stream ciphers has been described and executed. Protecting implementations against DPA attacks is usually difficult and expensive. The goal of countermeasures against DPA attacks is to make the power consumption independent of intermediate values of the stream cipher. In general, there are three basic groups into which these countermeasures can be

characterized: protocol countermeasures, algorithmic countermeasures, and hardware countermeasures [11].

The principles of the countermeasures can be implemented at different levels in a cryptographic device. In general, these techniques are theoretical countermeasures and only reduce the side channel leakage and do not fundamentally prevent a DPA. But the advantage of these countermeasures is to make the attack significantly harder. In this article, we provide a brief overview of hiding and masking logic styles (hardware countermeasures) and particularly we will use sense amplifier base logic (SABL) for secure implementation of stream ciphers. SABL is a logic style that uses a fixed amount of charge for every transition, including the degenerated events in which a gate does not change state. In every cycle, a SABL gate charges a total capacitance with a constant value.

So far, there has not been a unified architecture which can be used as a test bench for applicability of logic styles on stream ciphers. Regarding this, two FSR-based stream ciphers - Grain v.1 and Trivium stream ciphers - are implemented in cell level to find out the tradeoffs involved in designing the architecture and performance issues. Power traces of the resulting circuits exhibit that SABL significantly reduces signal to noise ratio (SNR). The rest of the paper is structured as follows: a general model of power analysis attack on stream ciphers is given in Section 2. Section 3 describes an overview of DPA Countermeasures on cell level. In sections 4 and 5 the descriptions of Grain v.1 and Trivium are explained. Design and simulation issues are described in section 6 and finally, conclusions are drawn in Section 7.

2 Differential Power Analysis of Stream Ciphers

DPA is based on the fact that CMOS logic and application specific details cause logic operations to have power characteristics that depend on the input data. It relies further on statistical analysis and error correction to extract the information from the power consumption that is correlated to the secret key [1]. In a DPA a hypothetical model of the device under attack is used to predict the power consumption. The classical setup for a DPA on stream ciphers is illustrated in Fig. 1. Output power traces are determined by the input data, IV, private key, output of the device and by many other parameters. An attacker to some extent has the potential knowledge of some of them (e.g. IV, input data and output data) while others are unknown. Regarding a DPA attack, multiple measurements of the power consumption of a cryptographic device are made. For each measurement, different chosen IV's are sent to the device. Since the cryptographic algorithm is known, a hypothesis on intermediate values can be used to calculate the targeted data values based on the random input values. If the correct hypothesis is used, the targeted data values are calculated correctly for all measurements. According to (1), the total power consumption of an embedded device depends on 3 factors:

$$P_{Total} = P_{Cons.} + P_{Noise} + P_{DD}. \tag{1}$$

With the help of statistical methods (calculation of correlations, mean values, etc.), the randomness of the data values that are not targeted ($P_{Const.}$: leakage currents and data independent power consumption and P_{Noise}: which comes from electrical noise) is exploited to reduce their effects on the power consumption traces. P_{DD} is the data dependent power consumption and is targeted in statistical analysis. After all, the result of the statistical operation indicates which key hypothesis is correct. Normally, a hamming distance power model is used to map the transitions that occur at the outputs of cells of a netlist to power consumption values. In CMOS gates, it is reasonable to assume that the main component of the data dependent power consumption is the dynamic power consumption which is the power dissipation of charging and discharging of output capacitance nodes ($P_{0\to1}$ or $P_{1\to0}$). In a CMOS gate, we can express dynamic power consumption by:

$$P_{Dynamic} = N \cdot C_L \cdot f \cdot V_{DD}^2 \tag{2}$$

where C_L is the gate load capacitance and N is the probability of a $0 \rightarrow 1$ or $1 \rightarrow 0$ output transition and f is the clock frequency. This equation shows that the power consumption of CMOS circuits is data dependent. Note that N is the most important factor in the hypothetical model. There are different techniques for calculation of it. For example, a variable gate delay model can be used for measuring the number of transitions and glitches of a circuit [7]. This technique can be easily applied to circuits by using a VHDL simulator in Register Transfer Level.

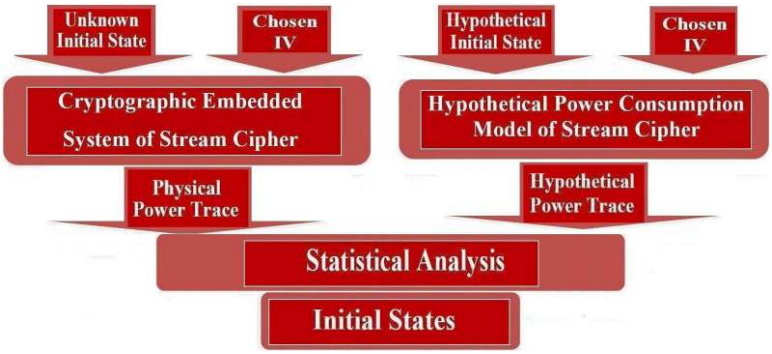


Figure 1: Differential power analysis model of stream ciphers.

3 DPA Countermeasures on Cell Level

So far, several methods in different ways have been proposed to counteract DPA attacks. In this section different known DPA countermeasures on cell level (hiding and masking techniques) are briefly presented and then their merits and disadvantages will be discussed.

The first structured approach to counteract DPA attacks at the cell level was the use of hiding logic styles. These styles try to break the correlation between an algorithm's intermediate results and the power consumption of the cryptographic device that executes this algorithm by making the instantaneous power consumption of the cells either random or the same in each clock cycle. The three major types of hiding logic styles are: Dual-Rail Precharge (DRP), Asynchronous, and Current Mode Logic (CML). DRP logic styles are the most popular types. For instance, SABL [10] and Wave Dynamic Differential Logic (WDDL) [15] are dual rail precharge logic styles whose logic gates are driven by a precharge signal to prevent glitches, and each logic signal is represented by two complementary wires. Other examples of DRP logic styles are Dual Spacer Dual Rail logic (DSDR), Three-phase Dual-rail Precharge Logic (TDPL) [16], and Three State Dynamic Logic (3SDL). Data dependent time of evaluation of the WDDL and its memory effect made it vulnerable to DPA attacks. One of the major drawbacks of hiding logic styles is the balancing of the cells and interconnect layouts to achieve constant power consumption. Since the charge and discharge of output nodes of dynamic and differential styles follow simple RC charge and discharge, cells and wires must mainly be balanced in a capacitive and resistive manner. But due to process variations, complex cross-coupling effects, and area limitations it is a hard task.

Besides hiding, masking at the cell level has become popular during the past few years. Using a masked logic style, designers also break the correlation between an algorithm's intermediate values and the power consumption of the cryptographic device that executes this algorithm. All intermediate values are masked by a random value. The cells then process only the masked intermediate values and their corresponding mask. Because the unmasked values and the masked value are uncorrelated, power consumption of the cell also remains uncorrelated to intermediate values. Generally there are two types of masking operation: boolean masking or arithmetic masking. If the masked cells are not activated in a data or operation dependent manner, masked logic styles counteract DPA attacks. There are two different

possible masking schemes: one mask per circuit (single masking) or one mask per signal. These masked bits are normally prepared by some random number/sequence generators.

Before, masking was mainly used at the architecture level. As a result, only a few practical results are available for this type of cell level countermeasure. For examples Masked Dual rail Precharge Logic (MDPL) [17] and Dual rail Random Switching Logic (DRSL) [18] were introduced by combining the masking scheme and dual rail precharge logic in order to use semi custom design tools without routing constrains. Designers can implement MDPL cells using commonly available conventional single rail standard cells. Only sequential cells are connected to the clock signal, and combinational cells precharge their outputs when their inputs have been set to the precharge value. The memory effect can reduce the DPA resistance of masked logic styles. Practical evaluations of the manufactured chips have also shown that early propagation is also a major threat to the DPA resistancy of masked logic styles.

Although all these efforts, it has been shown ([19], [20], [21]) that MDPL leaks information. For example in [19], it has been shown that MDPL is susceptible to the early propagation effect. In order to combat the early propagation issues, the designers of MDPL introduced a so called improved MDPL (iMDPL). In each iMDPL gate there is an evaluation precharge detection unit, which consists of three (CMOS) AND gates and two (CMOS) OR gates. Hence it is not surprising that the area requirements for iMDPL gates increased significantly compared to MDPL gates. Another threat to masked circuits is the detection of the mask value, which lets attackers completely cancel out the effect of masking in a DPA attack. In particular, such an attack is dangerous for single masked circuits, where only one mask value is used for all signals in the circuit. Increasing the number of mask values per circuit is an option but it is impractical regarding its high complexity and area utilization.

4 Sense Amplifier Based Logic

In this paper we will concentrate on SABL [10] for DPA resistive implementation of stream ciphers. SABL is part of the DRP logic styles. Fig. 2 shows the transistor schematic of standard SABL gate library used for implementation of ciphers. Equation (3) illustrates the power consumption of a SABL gate,

$$P = C_L \cdot f \cdot V_{DD}^2 + C_{Clk} \cdot f \cdot V_{DD}^2 \tag{3}$$

where C_L represents the total output capacitance of the gate and C_{Clk} is the clock propagation circuitry capacitance. As can be seen in the Fig.2, SABL gates can be designed using Differential Pull Down Networks (DPDN) or Differential Pull Up Networks (DPUN), controlled respectively by clk and $\overline{clk}$. This allows two modes for cascading SABL gates: domino connection (by connecting the outputs of the gate to the inputs of the next gate through inverters) or NP-connection (N-gates followed by P-gates like in NP-logic).

In SABL, the concepts of dual rail and precharge logic are combined to achieve constant power consumption. Precharging breaks a signal's sequence of values by splitting each clock cycle into precharge and evaluation phases. In the precharge phase, the complementary wires encoding a signal are set to a predefined precharge value, such as 1. In the subsequent evaluation phase, one of the two complementary wires is set to 1 according to the actual value that is processed. As a result, for each signal in a circuit, exactly one $0 \rightarrow 1$ transition and one $1 \rightarrow 0$ transition occur in a clock cycle. By ensuring a balance between the complementary wires between cells on the one hand and a balance of the internal structure of the cells on the other hand, designers can achieve constant power consumption. The price is high power consumption and high current spikes of these gates which appear at the beginning of the precharge phase. By the use of delayed clock mechanism introduced in [13] and [14] we can reduce the peak of these spikes.

But in practice the throughput is highly dependent on layout design of the chip to have balanced complementary wires. Since the charge and discharge of output nodes of differential styles follow simple

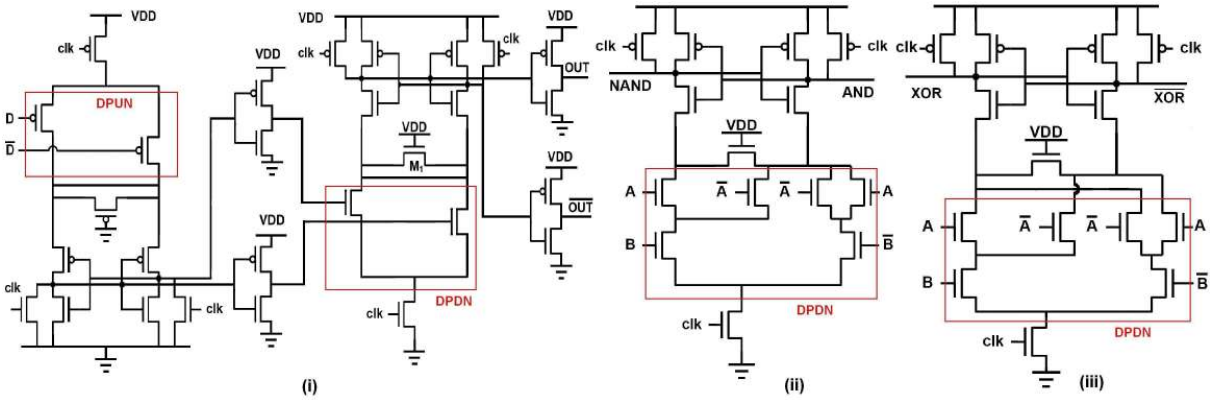


Figure 2: (i) SABL D-flip flop (ii) SABL Nand2 gate (iii) SABL Xor2 gate

RC charge and discharge, cells and wires must mainly be balanced in a capacitive and resistive manner. But due to process variations, complex cross-coupling effects, and area limitations this is hard to achieve. Avoiding these effects often requires custom cell design, which involves considerably more design effort than using available standard cells.

5 Grain Stream Cipher

Grain v.1 [5] is a stream cipher introduced in 2005 as a candidate for the hardware profile of eSTREAM project. Grain v.1 is a binary additive synchronous stream cipher with an internal state of 160 bits s_i, s_{i+1},..., s_{i+79} and b_i, b_{i+1},..., b_{i+79} residing in a linear feedback shift register (LFSR) and a nonlinear feedback shift register (NLFSR), respectively. The design of the algorithm mainly targets hardware environments where gate count, power consumption and memory is very limited. The key size of Grain is 80 bits (k_i, $0 \leq i \leq 79$). Additionally, an initial value of 64 bits (IV_i, $0 \leq i \leq 63$) is required. In initialization phase, all 80 NLFSR elements are loaded with the key bits, ($b_i = k_i$, $0 \leq i \leq 79$), then the first 64 LFSR elements are loaded with the IV bits, ($s_i = IV_i$, $0 \leq i \leq 63$). The last 16 bits of the LFSR are filled with ones. $f(x)$ and $g(x)$ are two polynomials used as feedback function for the LFSR and NLFSR.

$$f : s_{i+80} = s_{i+62} \oplus s_{i+51} \oplus s_{i+38} \oplus s_{i+23} \oplus s_{i+13} \oplus s_i \tag{4}$$

$$\begin{aligned} g : b_{i+80} = {} & s_i \oplus b_i \oplus b_{i+9} \oplus b_{i+14} \oplus b_{i+21} \oplus b_{i+28} \oplus b_{i+33} \oplus b_{i+37} \oplus b_{i+45} \oplus b_{i+52} \oplus b_{i+60} \oplus \\ & b_{i+62} \oplus b_{i+63} \cdot b_{i+60} \oplus b_{i+37} \cdot b_{i+33} \oplus b_{i+15} \cdot b_{i+9} \oplus b_{i+60} \cdot b_{i+52} \cdot b_{i+45} \oplus \\ & b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \oplus b_{i+63} \cdot b_{i+45} \cdot b_{i+28} \cdot b_{i+9} \oplus b_{i+60} \cdot b_{i+52} \cdot b_{i+37} \cdot b_{i+33} \oplus \\ & b_{i+63} \cdot b_{i+60} \cdot b_{i+21} \cdot b_{i+15} \oplus b_{i+63} \cdot b_{i+60} \cdot b_{i+52} \cdot b_{i+45} \cdot b_{i+37} \oplus \\ & b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \cdot b_{i+15} \cdot b_{i+9} \oplus b_{i+52} \cdot b_{i+45} \cdot b_{i+37} \cdot b_{i+33} \cdot b_{i+28} \cdot b_{i+21} \end{aligned} \tag{5}$$

The output function $h(x)$ uses as input selected bits from both feedback shift registers:

$$\begin{aligned} h(x) = {} & x_1 \oplus x_4 \oplus x_0 \cdot x_3 \oplus x_2 \cdot x_3 \oplus x_3 \cdot x_4 \oplus x_0 \cdot x_1 \cdot x_2 \oplus \\ + {} & x_0 \cdot x_2 \cdot x_3 \oplus x_0 \cdot x_2 \cdot x_4 \oplus x_1 \cdot x_2 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_4 \end{aligned} \tag{6}$$

where the variables x_0, x_1, x_2, x_3, and x_4 corresponds to the tap positions s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64} and b_{i+63} respectively. The output of the filter function is masked with the some state bits from the NFSR to produce the keystream z_i:

$$z_i = b_{i+1} \oplus b_{i+2} \oplus b_{i+4} \oplus b_{i+10} \oplus b_{i+31} \oplus b_{i+43} \oplus b_{i+56} \oplus h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$$

This output is used during the initialization phase as additional feedback to LFSR and NLFSR. During normal operation this value is used as key stream output. The generated output bits per clock cycle is called *Radix*. By implementing the small feedback functions, $f(x)$ and $g(x)$, and the output function several times the speed of Grain can easily reach up to Radix-32.

6 Trivium Stream Cipher

Trivium [6] is a stream cipher introduced in 2005 as a candidate for the hardware profile of the eSTREAM project. Trivium has an internal state of 288 bits $a_i, a_{i+1}, \ldots, a_{i+92}$, $b_i, b_{i+1}, \ldots, b_{i+83}$ and $c_i, c_{i+1}, \ldots, c_{i+110}$ - residing in three coupled NLFSRs A, B, and C of 93, 84, and 111 bits respectively. Trivium has a key $k = (k_0, \ldots, k_{79})$ of 80 bits as well as an initial value $IV = (IV_0, ..., IV_{79})$ of 80 bits. The initialization of the key and IV is done as follows:

$$\begin{cases} (a_0, \ldots, a_{92}) = (0, \ldots, 0, k_{79}, \ldots, k_0) \\ (b_0, \ldots, b_{83}) = (0, 0, 0, 0, IV_{79}, \ldots, IV_0) \\ (c_0, \ldots, c_{110}) = (1, 1, 1, 0, 0, \ldots, 0, 0) \end{cases} \tag{7}$$

Then, the state is updated over 4 full cycles, according to (3), but without generating key stream bits. After 1152 clocking it outputs a key stream bit z_i.

$$\begin{cases} a_{i+93} = a_{i+24} \oplus c_i \oplus (c_{i+1} \cdot c_{i+2}) \oplus c_{i+45} \\ b_{i+84} = b_{i+6} \oplus a_i \oplus (a_{i+1} \cdot a_{i+2}) \oplus a_{i+27} \\ c_{i+111} = c_{i+24} \oplus b_i \oplus (b_{i+1} \cdot b_{i+2}) \oplus b_{i+15} \end{cases} \tag{8}$$

$$z_i = a_i \oplus b_i \oplus c_i \oplus a_{i+27} \oplus b_{i+15} \oplus c_{i+45} \tag{9}$$

Trivium has a very simple structure that is well suited for different Radix implementations from Radix-1 to Radix-64 without noticeable hardware penalties.

The basic structure of the Grain v.1 and Trivium stream ciphers are shown in Fig. 3. In April 15, 2008, the eSTREAM competition was finished and according to the final report [12] both ciphers were selected among the four finalists of the H/W profile.

7 Design and Simulation Results

Both eSTREAM candidates are modeled at transistor level using a spice netlist. Circuit design of Grain v.1 and Trivium are mainly based on the techniques presented in [13] and [14]. In order to specify the impact of minimum feature size on the design, ciphers are designed using two technologies: typical BSIM3 $0.13\mu m$ CMOS SOI technology and typical $0.35\mu m$ CMOS SOI technology. Spice simulations were run to test the circuits by test vectors provided by the inventors of the ciphers using Hspice circuit simulator and C compiler. Domino cascading scheme is used for all SABL gate connections to make sure having a $0 \rightarrow 1$ transition in the input of all cascaded gates to prevent possible glitches. First a new standard gate library based on SABL logic is designed. Minimum possible sized transistors are used to lower the total capacitance to get lower dynamic power in (2). This will also minimize the charging time

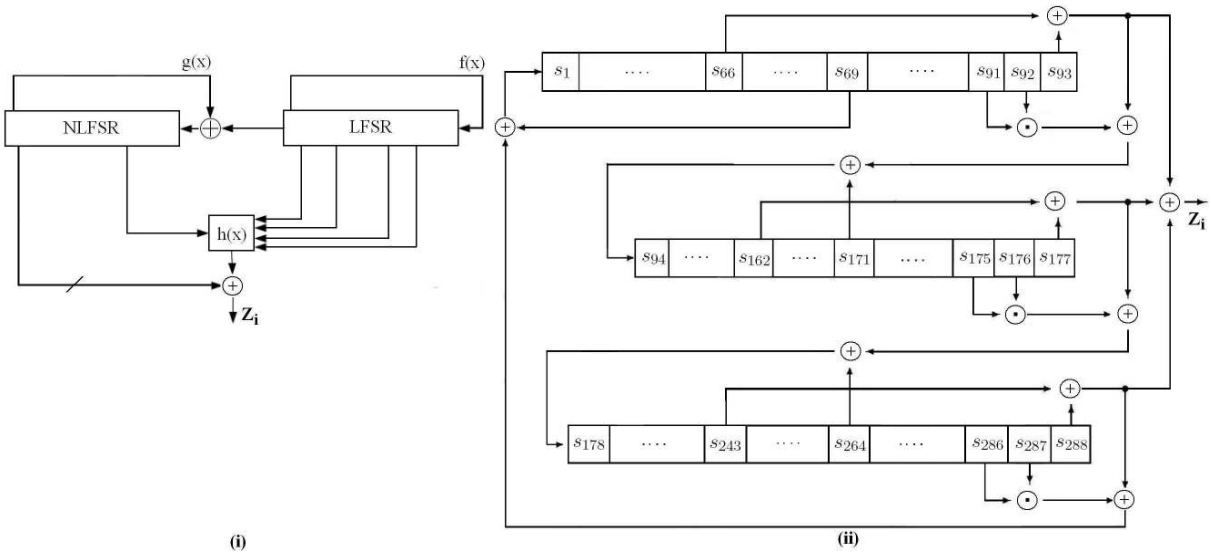


Figure 3: (i) Grain stream cipher (ii) Trivium stream cipher.

during precharge phase. Besides, this will help to cut the current spikes in the beginning of the precharge phase of each cycle. In order to get rid of the spikes, a delayed clocking mechanism is used [13], [14].

In order to increase security and speed of the initialization phase, a parallel data loading scheme is used, since in case of serial bit loading a straight forward simple power analysis attack is very likely to be successful in recovering all key bits. In parallel loading, key and IV will be loaded in the state bits after the first rising edge of the *CLK* signal. The gate level architecture of the parallel data loading scheme for standard CMOS is shown in Fig. 4. Note that in case of SABL all the wires and gates are dual rail. The area overhead is three Nand2 gates for each FlipFlop of the FSRs in the ciphers. Since all the components in the architecture need a *CLK* signal for switching from precharge phase into evaluation and vice versa, a chained buffer clock signal is needed. For the standard CMOS implementation of a the stream ciphers, standard two input Nand gates (4 Transistors), 8 transistor two input Xor gates, and the former 24 transistor, edge triggered D-FlipFlops (using eight Nand2), are used. In order to monitor all current variations, one sample has been taken every 50ps. Both simulations were run for four different 80-bit keys and IV's (64 bit IV for Grain v.1) in both SABL and standard CMOS designs. All power simulations are observed by $5MHz$ clock signal. The average power consumption per cycle was extracted by averaging the power consumption on 100 consecutive clock cycles. Then, the Mean Power Consumption (*MPC*), the Power Consumption Standard Deviation (*PCSD*), the Normalized Energy Deviation (*NED*) and Normalized Standard Deviation (*NSD*) were extracted for each simulated logic style (10). For example Supply current traces for standard CMOS design of Grain v.1 for the choice of K_2, IV_3 (in Table 1) in initialization phase is shown in Fig. 5.

$$NED = \frac{max(energy/cycle) - min(energy/cycle)}{max(energy/cycle)}, NSD = \frac{PCSD}{MPC} \tag{10}$$

In terms of transistor cost, the complete Trivium (including parallel data loading and clock buffering circuitary) required $\approx$ 23000 transistors for the SABL and $\approx$ 8500 transistors for the standard CMOS. In case of Grain v.1, $\approx$ 13500 transistors for the SABL and $\approx$ 6000 transistors for the standard CMOS are needed, confirming more than two times higher hardware cost for SABL styles. Table 1 shows the summary of final statistical power analysis results. For example in $0.13\mu m$ technology, and for K_1, IV_1, for Grain v.1, $\frac{PCSD_{SABL}}{PCSD_{SCMOS}} = 0.016$ which shows that the power consumption fluctuations of SABL implementation is nearly 1.6% of standard CMOS ($Power = Current \times Costant\ Supply\ Voltage$). This is a

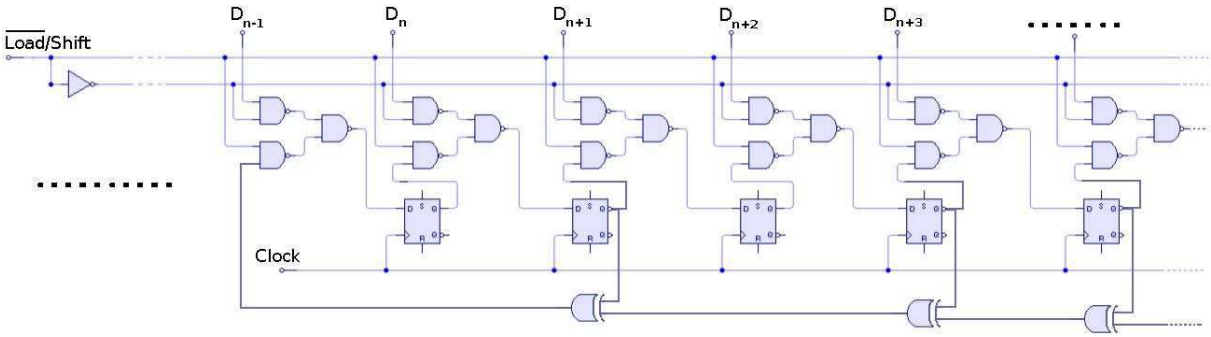


Figure 4: Parallel data loading scheme in FSRs (standard CMOS)

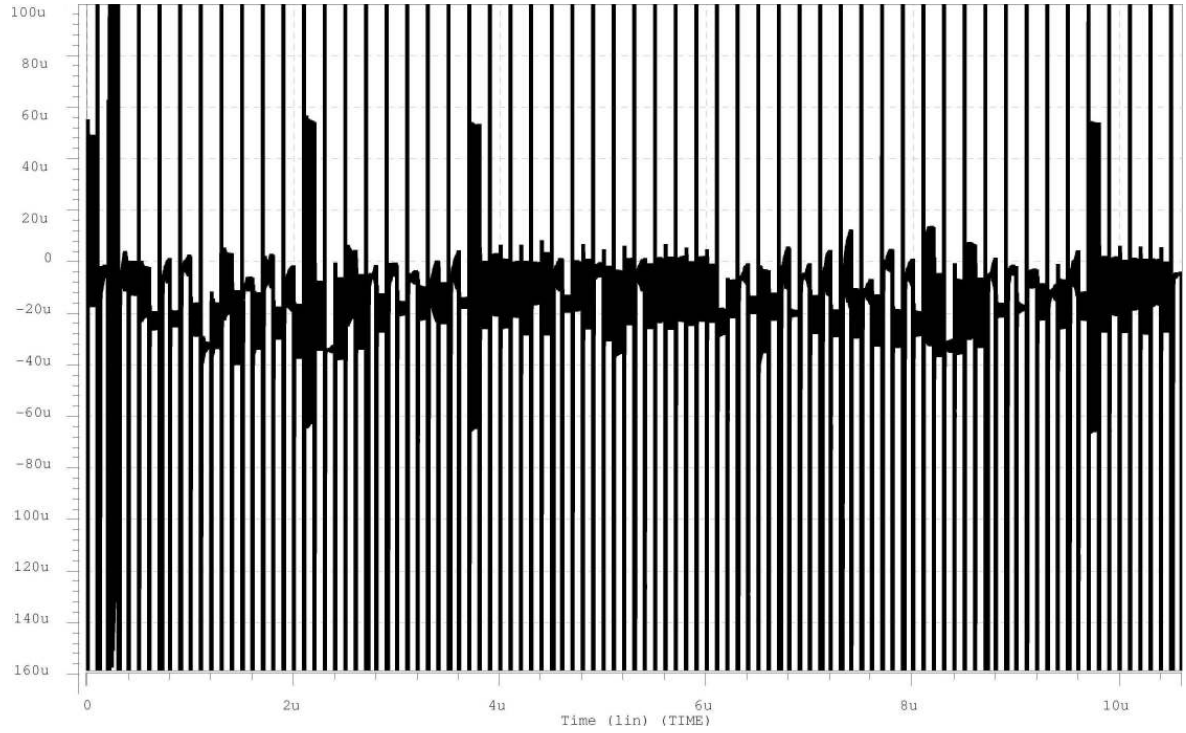


Figure 5: supply current variation of Standard CMOS design of Grain v.1 in 350nm technology

Table 1: Statistical power analysis of Trivium and Grain v.1 for different 80 bit hexadecimal key and IV's ($K_1 = AA\ldots A,\ K_2 = 80\ldots0, IV_1 = 55\ldots5, IV_2 = FF\ldots F, IV_3 = 00\ldots0, IV_4 = 11\ldots1$). [Note that in case of Grain v.1 the IV's are 64 bits]

Stream Cipher	Trivium				Grain			
Trivium	MPC [μW]	PCSD [μW]	NED	NSD	MPC [μW]	PCSD [μW]	NED	NSD
SABL	$0.35\mu m, V_{dd} = 3.3V, V_{TN} = 0.6V, V_{TP} = -0.85V$							
K_1, IV_1	949	1.2632	0.0091	0.00133	616	0.9497	0.0136	0.00154
K_1, IV_2	940	1.2469	0.0106	0.00132	605	0.9375	0.0111	0.00155
K_2, IV_3	938	1.2403	0.0089	0.00131	601	0.9318	0.0122	0.00155
K_2, IV_4	943	1.2531	0.0117	0.00133	611	0.9393	0.0120	0.00153
S-CMOS	$0.35\mu m, V_{dd} = 3.3V, V_{TN} = 0.6V, V_{TP} = -0.85V$							
K_1, IV_1	641	49.1	0.3292	0.0766	421	26.8	0.2784	0.0636
K_1, IV_2	637	19.3	0.2351	0.0303	402	18.5	0.1905	0.0460
K_2, IV_3	629	23.5	0.3139	0.0374	397	17.1	0.2187	0.0430
K_2, IV_4	635	41.7	0.2842	0.0657	415	29.6	0.3882	0.0713
SABL	$0.13\mu m, V_{dd} = 1.2V, V_{TN} = 0.4V, V_{TP} = -0.39V$							
K_1, IV_1	545	0.8435	0.0061	0.0015	378	0.7610	0.0124	0.0020
K_1, IV_2	537	0.7927	0.0054	0.0014	371	0.7424	0.0082	0.0020
K_2, IV_3	536	0.7831	0.0050	0.0014	369	0.7291	0.0079	0.0019
K_2, IV_4	541	0.8237	0.0059	0.0015	374	0.7482	0.0101	0.0020
S-CMOS	$0.13\mu m, V_{dd} = 1.2V, V_{TN} = 0.4V, V_{TP} = -0.39V$							
K_1, IV_1	337	31.20	0.8945	0.0926	258	22.50	0.7889	0.0872
K_1, IV_2	321	16.12	0.8191	0.0190	246	14.14	0.8026	0.0545
K_2, IV_3	319	17.14	0.8614	0.0537	242	12.31	0.8402	0.0509
K_2, IV_4	326	29.94	0.9218	0.0918	252	21.72	0.7924	0.0862

major improvement but still $PCSD_{SABL} \neq 0$ and very small current variations are detectable. The overall comparison between SABL and standard CMOS design for 4 different key and IV choices (Table 1) is shown in Table 2. DPA resistivity factor is calculated in (11):

$$DPA\ Resistivity \propto \frac{1}{PCSD} \tag{11}$$

One of the fundamental parameters of a cryptographic algorithm is the amount of data it can process within a given period. The total throughput of the algorithm is expressed as $Mbits/s$ and can be calculated from $T = f \times Radix$ where f is the clock frequency of the design (e.g. $5MHz$). Since Trivium throughput rate for SABL and S-CMOS designs are equal, in order to make a fair comparison, a new normalized

Table 2: Overall comparison for SABL and S-CMOS design of Trivium (All data are normalized)

Cipher	Trivium				Grain			
Logic Style	SABL	S-CMOS	SABL	S-CMOS	SABL	S-CMOS	SABL	S-CMOS
Technology	$0.13\mu m$		$0.35\mu m$		$0.13\mu m$		$0.35\mu m$	
Transistor Cost (A)	1	0.37	1	0.37	1	0.44	1	0.44
Power Consumption (P)	1	0.60	1	0.67	1	0.66	1	0.67
DPA Resistancy (DR)	1	0.0374	1	0.0435	1	0.045	1	0.043
Qualifying Factor (QF)	1	0.062	1	0.065	1	0.068	1	0.064

Qualifying Factor (QF) is defined in (12):

$$QF = \frac{DR \times T}{P \times A} \tag{12}$$

Where, A, P, and DR corresponds to transistor cost, power consumption, and DPA resistancy respectively. At the end of the simulations and data analysis we exhibited that SABL logic styles allow to significantly decrease the supply current variations of both eSTREAM circuits. But still in both designs very small current variations are detectable. As a disadvantage this could be a start of a DPA attack since the predictability of the energy variations is more critical than their amplitude. It is clear that decreasing the power consumption variations will affect all the stream cipher design components in exactly the same way, and therefore not affect the SNR. Since DPA efficiency depends on the possibility to predict the power consumption of a device in function of its input data and the value of the correlation coefficient, the attack is still theoretically feasible against SABL circuits. But these current differences are due to the presence of parasitic capacitances in the design and therefore, they cannot be predicted without a precise transistor level knowledge of the circuit. As a consequence, an attacker can only target one specific implementation and preliminarily needs to build a table containing the power consumption differences in function of the circuit input data. These informations are not usually made available to the users in full custom design. Moreover, under the assumption that we can perfectly predict and measure the power consumption, a circuit resistance is equal for any logic style. Nevertheless, in practice, measurements are not perfect and induce noise, independently of the logic style considered. This will cause a reduction of the correlation values, depending on the power consumption variances, although it is hard to evaluate and highly depends on the attacker measurement setup.

As can be seen in the Table 1 and Table 2 the DPA resistancy is improved for smaller minimum feature sized designs. Although, current variations do not follow the scaling rules. This is mainly because of clock feedthrough effect and also the former subthreshhold leakages which play a big roll in deep submicron designs.

Stream ciphers always had the reputation of efficiency in hardware. Their smaller architecture helps to use full custom design flow in order to have balanced routing of component wires. So simpler stream cipher designs would have lower design costs. Regarding design flow, Trivium has lower hardware complexity and circuit design is easier. Although Trivium has bigger architecture, timing constraints and clock distribution of Trivium are the same as Grain. Comparing resistance against DPA attacks of the two eSTREAM candidates, simulations show Grain has lower current spikes and smaller current variations. This is thanks to the higher circuit complexity of Grain which combines different current variation of gates to achieve a semi random supply current variation. Current spikes in Trivium are due to the higher number of flip flops. Another disadvantage of Trivium is its large number of iterations in initialization phase (1152 rounds) which let attackers to have more power traces.

8 Summary and Conclusions

This paper investigated the use of SABL logic to counteract power analysis attacks. In particular, an efficient DPA resistive circuit for Grain v.1 and Trivium stream ciphers have been designed and compared with their standard CMOS implementations. First we exhibited that SABL allow to significantly decrease the circuit energy variations. This is due to equal amounts of power consumption in each clock cycle of SABL gates. All implementations have been done on transistor level but in practice the cipher itself is part of a system on chip with lots of other circuits which can increase $P_{Cons.} + P_{Noise}$ in (1) to achieve lower SNR. Although SABL cannot be completely tamper resistant, this logic probably presents acceptable security margins for general applications of stream ciphers. For future work interested researchers can investigate some circuit changes in SABL styles to counteract other side channel attacks such as fault attacks to obtain more security.

Acknowledgment

Reza Ebrahimi Atani wishes to thank the Iran Telecommunication Research Center (ITRC) for their financial support (www.itrc.ac.ir).

Bibliography

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology - CRYPTO'99*, Springer-Verlag, LNCS Vol. 1666, pp. 388–397, 1999.

[2] Ch. Rechberger and E. Oswald, "Stream Ciphers and Side-Channel Analysis" *In SASC 2004 - The State of the Art of Stream Ciphers*, Brugge, Belgium, Workshop Record, pp. 320–326, Oct. 14-15, 2004.

[3] J. Lano, N. Mentens, B Preneel, and I. Verbauwhede, "Power Analysis of Synchronous Stream Ciphers with Resynchronization Mechanism" *In SASC 2004 - The State of the Art of Stream Ciphers*, Brugge, Belgium, Workshop Record, pp. 327–333, Oct. 14-15, 2004.

[4] W. Fischer, B. M. Gammel, O. Kniffler, J. Velton, "Differential Power Analysis of Stream Ciphers," *Topics in Cryptology - CT-RSA 2007*, Springer-Verlag, LNCS, Vol. 4377, pp. 257–270, 2007.

[5] M. Hell, Th. Johansson, A. Maximov, and W. Meier, "Grain - A Stream Cipher for Constrained Environments," 2006, eSTREAM project website.

[6] C. De Canniere, and B. Preneel, "Trivium Specifications," 2005, eSTREAM project website.

[7] T. Seko, A. Nakamura, and T. Kikuno, "Measurement of glitches based on variable gate delay model using VHDL simulator," *Asia-Pacific Conference on Circuits and Systems,* Nov. 1998, PP. 767 – 770.

[8] B. Gierlichs et. al., "Susceptibility of eSTREAM Candidates towards Side Channel Analysis," *SASC 2008,* Switzerland, Feb. 13-14, 2008, Workshop Record, pp. 320 – 326.

[9] K. Tiri, and I. Verbauwhede, "Charge recycling sense amplifier based logic: securing low power security ICs against DPA" *30th European Conference on Solid-State Circuits,* 21-23 Sept. 2004, pp. 179 – 182.

[10] K. Tiri, M. Akmal, and I. Verbauwhede, "A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards," *28th European Solid State Circuits Conference*, IEEE Press, pp. 403 – 406, , 24-26 Sep. 2002.

[11] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards,* Springer, 2007.

[12] S. Babbage et. al., *The eSTREAM Portfolio,* April 2008, eSTREAM project website.

[13] R.E. Atani, W. Meier, S. Mirzakuchaki, and S.E.Atani, "Design and Implementation of DPA Resistive Grain-128 Stream Cipher Based on SABL Logic", *International Journal of Computers, Communications & Control*, Vol. III (supl. issue), pp. 293 – 298, 2008.

[14] R.E. Atani, W. Meier, S. Mirzakuchaki, and S.E.Atani, "Design and simulation of a DPA resistive circuit for Trivium stream cipher based on SABL styles" *Mixdes 2008,* 19-21 June. 2008, pp. 203 – 208.

[15] K. Tiri, and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation" *DATE 2004,* 2004, pp. 246–251.

[16] M. Bucci, L. Giancane, R.o Luzzi, and A. Trifiletti, "Three-Phase Dual-Rail Precharge Logic" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 232–241.

[17] T. Popp, and S. Mangard, "Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints" *In Cryptographic Hardware and Embedded Systems CHES 2005,* Vol. 3659 of LNCS, Springer, 2005, pp. 172–186.

[18] Z. Chen, and Y. Zhou, "Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage," *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 242–254.

[19] D. Suzuki, and M. Saeki, "Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 255–269.

[20] P. Schaumont, and K. Tiri, "Masking and Dual-Rail Logic Dont Add Up," *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 95–106.

[21] B. Gierlichs, "DPA-Resistance Without Routing Constraints?" *In Cryptographic Hardware and Embedded Systems CHES 2006,* Vol. 4249 of LNCS, Springer-Verlag, 2006, pp. 107–120.

Reza Ebrahimi Atani
Electrical Engineering Department
Iran University of Science and Technology (IUST),
Narmak,16846, Tehran, Iran.
E-mail: rebrahimi@iust.ac.ir

Sattar Mirzakuchaki
Electrical Engineering Department
Iran University of Science and Technology (IUST),
Narmak,16846, Tehran, Iran.
E-mail: m-kuchaki@iust.ac.ir

Shahabaddin Ebrahimi Atani
Mathematics Department
University of Guilan
P.O.Box 1914, Rasht, Iran.
E-mail: ebrahimi@guilan.ac.ir

Willi Meier
IAST Institute
FHNW
CH 5210, Windisch, Switzerland.
E-mail: willi.meier@fhnw.ch

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 336-342

Fuzzy Szpilrajn Theorem through Indicators

Irina Georgescu

Abstract: In this paper there are studied some numerical indicators which measure the degree to which a fuzzy relation verifies some properties (reflexivity, transitivity, etc.).The main result is a fuzzy generalization of the Szpilrajn theorem in terms of such numerical indicators and is applied to any fuzzy relation.
Keywords: fuzzy relation, Szpilrajn theorem, similarity

1 Introduction

The classical Szpilrajn theorem [10] asserts that any partial order can be extended to a total order. This result has been followed by refinements and generalizations and has been used in applications too (e. g. consumer theory [8]).

The first fuzzy version of the Szpilrajn theorem has been established by Zadeh [11]. Other fuzzy versions of this theorem can be found in [4], [6]. In [3] the topic is systematically studied in the framework of the fuzzy orders with respect to a left–continuous t–norm $*$ and a $*$–similarity relation Ω.

The idea of this paper is the following: instead of studying a property P of a fuzzy relation R (e. g. reflexivity, transitivity, etc.) to define numerical indicators which should express "the degree to which the fuzzy relation R verifies the property P". In this way, instead of considering a fuzzy order R on a set X we will have a number $Ord(R)$ which should measure "the degree to which R is a fuzzy order".

The main result of the paper is a generalization of the Szpilrajn theorem expressed in terms of such numerical indicators. It is a refinement of Theorem 6.2 in [3] and it is applied to any fuzzy relation.

2 Preliminaries

In this section we shall recall some basic facts on the residuum associated with a left–continuous t–norm and on fuzzy relations ([1], [2], [4], [5], [7], [9]).

For any $a,b \in [0,1]$ we denote $a \vee b = \max(a,b)$ and $a \wedge b = \min(a,b)$. More generally, for any set $\{a_i\}_{i\in I} \subseteq [0,1]$ we denote $\bigvee_{i\in I} a_i = \sup\{a_i | i \in I\}$ and $\bigwedge_{i\in I} a_i = \inf\{a_i | i \in I\}$.

Let $*$ be a left–continuous t–norm [7], [5]. The *residuum* $\rightarrow$ associated with $*$ is introduced by $a \rightarrow b = \bigvee\{c \in [0,1] | a * c \leq b\}$.

The *biresiduum* $\leftrightarrow$ is denoted by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

We fix a left–continuous t–norm $*$.

Lemma 1. *[1], [5] For any $a,b,c \in [0,1]$ the following properties hold:*

(1) $a * b \leq c$ *iff* $a \leq b \rightarrow c$;

(2) $a \wedge b = a * (a \rightarrow b)$;

(3) $a \leq b$ *iff* $a \rightarrow b = 1$;

(4) $a = 1 \rightarrow a$;

(5) $1 = a \rightarrow a$;

(6) $a \leq a * (a \leftrightarrow b)$.

Lemma 2. *[1], [5] For any* $\{a_i\}_{i\in I} \subseteq [0,1]$ *and* $a \in [0,1]$ *the following properties hold:*

(1) $(\bigvee_{i\in I} a_i) * a = \bigvee_{i\in I}(a_i * a)$;

(2) $a \to (\bigwedge_{i\in I} a_i) = \bigwedge_{i\in I}(a \to a_i)$;

(3) $(\bigvee_{i\in I} a_i) \to a = \bigwedge_{i\in I}(a_i \to a)$.

Let X be a non–empty subset. A *fuzzy subset* of X is a function $A : X \to [0,1]$. Denote by $\mathcal{F}(X)$ the family of the fuzzy subsets of X. For any $A, B \in \mathcal{F}(X)$ denote $A \subseteq B$ if $A(x) \leq B(x)$ for any $x \in X$.

A *fuzzy relation* on X is a function $R : X^2 \to [0,1]$. R is said to be

- reflexive if $R(x,x) = 1$ for any $x \in X$;
- symmetric if $R(x,y) = R(y,x)$ for any $x, y \in X$;
- $*$–transitive if $R(x,y) * R(y,z) \leq R(x,z)$ for all $x, y, z \in X$;
- strongly complete if $R(x,y) \vee R(y,x) = 1$ for any $x, y \in X$.

A reflexive, symmetric and $*$–transitive fuzzy relation Ω on X will be called $*$–similarity relation.

Let Ω be a $*$–similarity relation on X and R a fuzzy relation on X. R is said to be:

- Ω–reflexive if $\Omega(x,y) \leq R(x,y)$ for any $x, y \in X$;
- $(*,\Omega)$–antisymmetric if $R(x,y) * R(y,x) \leq \Omega(x,y)$ for any $x, y \in X$.

A $(*,\Omega)$–order is $*$–transitive, Ω–reflexive and $(*,\Omega)$–antisymmetric fuzzy relation R on X. Let R, Q be two $(*,\Omega)$–orders on X. We say that Q is an *extension* of R if $R \subseteq Q$, i. e. $R(x,y) \leq Q(x,y)$ for all $x, y \in X$.

The following fuzzy generalization of the Szpilrajn theorem was proved in [3]:

Theorem 3. *Let* Ω *be a* $*$*–similarity relation on* X*. Then any* $(*,\Omega)$*–order on* X *has a strongly complete extension.*

3 Some indicators

Let $*$ be a left–continuous t–norm and Ω be a $*$–similarity relation on X.

Definition 4. For any fuzzy relation R on X let us define:

$Ref(R) = \bigwedge_{x\in X} R(x,x)$;

$Trans(R) = \bigwedge_{x,y,z\in X} [R(x,y) * R(y,z) \to R(x,z)]$;

$Ref_\Omega(R) = \bigwedge_{x,y\in X} (\Omega(x,y) \to R(x,y)]$;

$Ant_\Omega(R) = \bigwedge_{x,y\in X} [R(x,y) * R(y,x) \to \Omega(x,y)]$;

$SC(R) = \bigwedge_{x,y\in X} (R(x,y) \vee R(y,x))$;

$Ord_\Omega(R) = Ref_\Omega(R) \wedge Ant_\Omega(R) \wedge Trans(R)$.

Lemma 5. *For any fuzzy relation R the following equivalences hold:*

(1) $Ref(R) = 1$ *iff R is reflexive;*

(2) $Trans(R) = 1$ *iff R is* $*$*–reflexive;*

(3) $Ref_\Omega(R) = 1$ *iff* R *is* Ω*–reflexive;*

(4) $Ant_\Omega(R) = 1$ *iff* R *is* $(*, \Omega)$*–antisymmetric;*

(5) $SC(R) = 1$ *iff* R *is strongly complete;*

(6) $Ord_\Omega(R) = 1$ *iff* R *is a* $(*, \Omega)$*–order.*

$Ref(R)$ will be called the *d*egree of reflexivity of R, $Trans(R)$ the *d*egree of $*$–transitivity of R, etc. The indicators introduced above refine the properties of reflexivity, transitivity, etc. of fuzzy relations. Thus, instead of saying that the fuzzy relation R is reflexive, the real number $Ref(R)$ will measure "the degree to which R is reflexive".

Proposition 6. *Let* R *be a fuzzy relation on* X *and* $x, y, z \in X$*. Then*

(1) $Ref(R) \leq R(x,x)$*;*

(2) $Trans(R) * R(x,y) * R(y,z) \leq R(x,z)$*;*

(3) $Ref_\Omega(R) * \Omega(x,y) \leq R(x,y)$*;*

(4) $Ant_\Omega(R) * R(x,y) * R(y,x) \leq \Omega(x,y)$*;*

(5) $Ord_\Omega(R) \leq \Omega(x,y) \leftrightarrow (R(x,y) * R(y,x))$*;*

(6) $Ord_\Omega(R) \leq R(x,x)$.

4 Main result

In this section we shall prove a generalization of the theorem of Szpilrajn formulated in terms of the indicators introduced in the previous paragraph. The result will be valid for any fuzzy relations and in particular one will obtain Theorem 3.

Let $*$ be a left–continuous t–norm and Ω a $*$–similarity relation on X. If R and Q are two fuzzy relations on X then we denote

$R \preccurlyeq Q$ iff $R \subseteq Q$ and $Ord_\Omega(R) \leq Ord_\Omega(Q)$.

It is easy to see that $\preccurlyeq$ is a partial order on the set of the fuzzy relations defined on X. If Q is a fuzzy relation on X then we denote by $Ext(Q)$ the set of all fuzzy relations R on X with the property that $Q \preccurlyeq R$.

Lemma 7. *If* Q *is a fuzzy relation on* X *then the partially ordered set* $(Ext(Q), \preccurlyeq)$ *admits a maximal element.*

Proof. We prove that $(Ext(Q), \preccurlyeq)$ is inductive. We consider a chain $(R_i)_{i \in I}$ in $Ext(Q)$: for any $i, j \in I$ we have $R_i \preccurlyeq R_j$ or $R_j \preccurlyeq R_i$. Of course $Q \preccurlyeq R_i$ for any $i \in I$. We will denote $R = \bigcup_{i \in I} R_i$. It suffices to prove that $R \in Ext(Q)$. It is obvious that $Q \subseteq R$ therefore we have to prove that $Ord_\Omega(Q) \leq Ord_\Omega(R)$. We show first that

(a) $Ord_\Omega(Q) \leq Ref_\Omega(R)$.

Let $x, y \in X$ and $i \in I$. Since $Q \subseteq R_i$ it follows immediately

$Ord_\Omega(Q) \leq Ref_\Omega(Q) \leq Ref_\Omega(R_i) \leq \Omega(x,y) \to R_i(x,y)$.

By applying Lemma 1 (2) and the previous inequality

$Ord_\Omega(Q) * \Omega(x,y) \leq \Omega(x,y) * (\Omega(x,y) \to R_i(x,y)) =$

$= \Omega(x,y) \wedge R_i(x,y) \leq R_i(x,y) \leq R(x,y)$

from where according to Lemma 1 (1), $Ord_\Omega(Q) \leq \Omega(x,y) \to R(x,y)$. It follows

$Ord_\Omega(Q) \leq \bigwedge_{x,y\in X} (\Omega(x,y) \to R(x,y)) = Ref_\Omega(R)$.

We intend to prove that

(b) $Ord_\Omega(Q) \leq Ant_\Omega(R)$

Let $x,y \in X$. Then

$Ord_\Omega(Q) * R(x,y) * R(y,x) = Ord_\Omega(Q) * [\bigvee_{i\in I} R_i(x,y)] * [\bigvee_{j\in I} R_j(y,x)] =$

$= \bigvee_{i,j\in I} Ord_\Omega(Q) * R_i(x,y) * R_j(y,x)$.

Let $i,j \in I$. Assume that $R_i \preccurlyeq R_j$ therefore $R_i \subseteq R_j$ and $Ord_\Omega(R_i) \leq Ord_\Omega(R_j)$. Then, according to $Q \preccurlyeq R_j$ and Proposition 6 (4):

$Ord_\Omega(Q) * R_i(x,y) * R_j(y,x) \leq Ord_\Omega(R_j) * R_j(x,y) * R_j(y,x) \leq$

$\leq Ant_\Omega(R_j) * R_j(x,y) * R_j(y,x) \leq \Omega(x,y)$.

Since this inequality is valid for any $i,j \in I$ it follows $Ord_\Omega(Q) * R(x,y) * R(y,x) \leq \Omega(x,y)$, therefore, according to Lemma 1 (1), $Ord_\Omega(Q) \leq R(x,y) * R(y,x) \to \Omega(x,y)$. From here we deduce

$Ord_\Omega(Q) \leq \bigwedge_{x,y\in X} [R(x,y) * R(y,x) \to \Omega(x,y)] = Ant_\Omega(R)$.

We still have to prove

(c) $Ord_\Omega(Q) \leq Trans(R)$.

Let $x,y \in X$. Then

$Ord_\Omega(Q) * R(x,y) * R(y,z) = \bigvee_{i,j\in I} Ord_\Omega(Q) * R_i(x,y) * R_j(y,x)$

Let $i,j \in I$. Assume $R_i \preccurlyeq R_j$ therefore $R_i \subseteq R_j$ and $Ord_\Omega(R_i) \leq Ord_\Omega(R_j)$. According to Proposition 6 (2):

$Ord_\Omega(Q) * R_i(x,y) * R_j(y,z) \leq Ord_\Omega(R_j) * R_j(x,y) * R_j(y,z) \leq$

$\leq Trans(R_j) * R_j(x,y) * R_j(y,z) \leq R_j(x,z) \leq R(x,z)$

from where, $Ord_\Omega(Q) * R(x,y) * R(y,z) \leq R(x,z)$. By applying Lemma 1 (1) it follows that for any $x,y,z \in X$ we have $Ord_\Omega(Q) \leq R(x,y) * R(y,z) \to R(x,z)$, from where

$Ord_\Omega(Q) \leq \bigwedge_{x,y,z\in X} [R(x,y) * R(y,z) \to R(x,z)] = Trans(R)$.

From (a), (b) and (c) one obtains $Ord_\Omega(Q) \leq Ord_\Omega(R)$. We have shown that $(Ext(Q), \preccurlyeq)$ is inductive. According to Zorn's axiom a maximal element exists in $Ext(Q)$. □

In the following we will situate ourselves in the case of the Gödel t–norm $\wedge$.

Theorem 8. *Let Q be a fuzzy relation on X. Then there exists a fuzzy relation R on X such that $Q \preccurlyeq R$ and $Ord_\Omega(Q) \leq SC(R)$.*

Proof. According to Lemma 7 there exists a fuzzy relation R on X maximal in $(Ext(Q), \preccurlyeq)$. Then $Q \preccurlyeq R$. It remains to prove that $Ord_\Omega(Q) \leq SC(R)$. We assume by absurdum that

$Ord_\Omega(Q) \not\leq SC(R) = \bigwedge_{x,y\in X} (R(x,y) \vee R(y,x))$.

therefore there exist $a,b \in X$ such that $Ord_\Omega(Q) \not\leq R(a,b) \vee R(b,a)$ from where $Ord_\Omega(Q) \not\leq R(a,b)$ and $Ord_\Omega(Q) \not\leq R(b,a)$. Assume $R(a,b) \leq R(b,a)$. According to the lines above, $R(b,a) < Ord_\Omega(Q)$. We define a new fuzzy relation R' on X by

$R'(x,y) = R(x,y) \vee (R(x,b) \wedge R(a,y))$

for any $x,y \in X$. We intend to prove that $R \preccurlyeq R'$. It is obvious that $R \subseteq R'$ therefore it remains to prove that $Ord_\Omega(R) \leq Ord_\Omega(R')$. From $R \subseteq R'$ it follows immediately

(1) $Ord_\Omega(R) \leq Ref_\Omega(R')$

We establish now the inequality:

(2) $Ord_\Omega(R) \leq Ant_\Omega(R')$

Let $x,y \in X$. Then

$Ord_\Omega(R) \wedge R'(x,y) \wedge R'(y,x) =$
$= Ord_\Omega(R) \wedge [R(x,y) \vee (R(x,b) \wedge R(a,y))] \wedge [R(y,x) \vee (R(y,b) \wedge R(a,x))] =$
$= [Ord_\Omega(R) \wedge R(x,y) \wedge R(y,x)] \vee [Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,x)] \vee$
$\vee [Ord_\Omega(R) \wedge R(y,x) \wedge R(x,b) \wedge R(a,y)] \vee [Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \wedge R(y,b) \wedge R(a,x)]$
We will establish the following inequalities:
(a) $Ord_\Omega(R) \wedge R(x,y) \wedge R(y,x) \leq \Omega(x,y)$;
(b) $Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,x) \leq \Omega(x,y)$;
(c) $Ord_\Omega(R) \wedge R(y,x) \wedge R(x,b) \wedge R(a,y) \leq \Omega(x,y)$;
(d) $Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \wedge R(y,b) \wedge R(a,x) \leq \Omega(x,y)$.
In order to obtain (a) we use Proposition 6 (4):
$Ord_\Omega(R) \wedge R(x,y) \wedge R(y,x) \leq Ant_\Omega(R) \wedge R(x,y) \wedge R(y,x) \leq \Omega(a,b)$.
We treat now the other three cases. According to Proposition 6 (5) we have
$Ord_\Omega(R) \leq \Omega(a,b) \leftrightarrow (R(a,b) \wedge R(b,a)) = \Omega(a,b) \leftrightarrow R(a,b)$
We consider first the case $R(x,y) \leq R(y,x)$. Then $Ord_\Omega(R) \leq \Omega(x,y) \leftrightarrow R(x,y)$ with the same argument as above. (b) results like this:
$Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,x) \leq R(x,y) \wedge [\Omega(x,y) \leftrightarrow R(x,y)] \leq \Omega(x,y)$
Now we treat cases (c) and (d). First we notice that according to Proposition 6 (2):
$Ord_\Omega(R) \wedge R(y,x) \wedge R(x,b) \wedge R(a,y) \leq Trans(R) \wedge R(a,y) \wedge R(y,x) \wedge R(x,b) \leq R(a,b)$
therefore
$Ord_\Omega(R) \wedge R(y,x) \wedge R(a,y) \leq R(a,b) \wedge [R(a,b) \leftrightarrow \Omega(a,b)] \leq \Omega(a,b)$.
Analogously we obtain:
$Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \wedge R(y,b) \wedge R(a,x) \leq \Omega(a,b)$.
We consider the possible subcases:
(i) $\Omega(a,b) \leq R(x,y)$;
(ii) $\Omega(a,b) > R(x,y)$.
According to the proof above, in case (i) the inequalities (c) and (d) are immediate. We are situated now in case (ii). One notices that
$Ord_\Omega(R) \wedge R(x,b) \wedge \Omega(a,b) \wedge R(a,y) \leq \Omega(a,b) \wedge [\Omega(a,b) \leftrightarrow R(a,b)] \leq R(a,b) \leq R(b,a)$
therefore
$Ord_\Omega(R) \wedge R(x,b) \wedge \Omega(a,b) \wedge R(a,y) \leq Ord_\Omega(R) \wedge R(x,b) \wedge R(b,a) \wedge R(a,y) \leq Trans(R) \wedge R(x,b) \wedge R(b,a) \wedge R(a,y) \leq R(x,y)$
From $Ord_\Omega(R) \wedge R(x,b) \wedge \Omega(a,b) \wedge R(a,y) \leq R(x,y)$ and $\Omega(a,b) > R(x,y)$ it follows
$Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \leq R(x,y)$.
By using this last inequality we have
$Ord_\Omega(R) \wedge R(y,x) \wedge R(x,b) \wedge R(a,y) \leq Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \leq R(x,y)$
from where
$Ord_\Omega(R) \wedge R(y,x) \wedge R(x,b) \wedge R(a,y) \leq R(x,y) \wedge [R(x,y) \leftrightarrow \Omega(x,y)] \leq \Omega(x,y)$
Thus (c) was proved and (d) follows analogously.
The case $R(y,x) \leq R(x,y)$ is treated analogously. Therefore the inequalities (a)–(d) are true, so $Ord_\Omega(R) \wedge R'(x,y) \wedge R'(y,x) \leq \Omega(x,y)$. Cf. Lemma 1 (1) for any $x,y,z \in X$ we have $Ord_\Omega(R) \leq (R'(x,y) \wedge R'(y,x)) \rightarrow \Omega(x,y)$ therefore
$Ord_\Omega(R) \leq \bigwedge_{x,y \in X} [(R'(x,y) \wedge R'(y,x)) \rightarrow \Omega(x,y)] = Ant_\Omega(R')$
Now we establish the inequality
(3) $Ord_\Omega(R) \leq Trans(R')$.
Let $x,y,z \in X$. We prove
(4) $Ord_\Omega(R) \wedge R'(x,y) \wedge R'(y,z) \leq R'(x,z)$.
We notice that
$Ord_\Omega(R) \wedge R'(x,y) \wedge R'(y,z) =$

$= Ord_\Omega(R) \wedge [R(x,y) \vee (R(x,b) \wedge R(a,y))] \wedge [R(y,z) \vee (R(y,b) \wedge R(a,z))] =$
$= [Ord_\Omega(R) \wedge R(x,y) \wedge R(y,z)] \vee [Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,z)] \vee$
$\vee [Ord_\Omega(R) \wedge R(y,z) \wedge R(x,b) \wedge R(a,y)] \vee [Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \wedge R(y,b) \wedge R(a,z)].$

Then to prove (4) is equivalent with establishing the following inequalities:

(e) $Ord_\Omega(R) \wedge R(x,y) \wedge R(y,z) \leq R'(x,z)$;

(f) $Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,z) \leq R'(x,z)$;

(g) $Ord_\Omega(R) \wedge R(y,z) \wedge R(x,b) \wedge R(a,y) \leq R'(x,z)$;

(h) $Ord_\Omega(R) \wedge R(x,b) \wedge R(a,y) \wedge R(y,b) \wedge R(a,z) \leq R'(x,z)$.

(e) follows by applying Proposition 6 (2):

$Ord_\Omega(R) \wedge R(x,y) \wedge R(y,z) \leq Trans(R) \wedge R(x,y) \wedge R(y,z) \leq R(x,z) \leq R'(x,z)$.

(f) and (g) follow like this:

$Ord_\Omega(R) \wedge R(x,y) \wedge R(y,b) \wedge R(a,z) \leq (Trans(R) \wedge R(x,y) \wedge R(y,b)) \wedge R(a,z) \leq R(x,b) \wedge R(a,z) \leq R'(x,z)$;

$Ord_\Omega(R) \wedge R(y,z) \wedge R(x,b) \wedge R(a,y) \leq (Trans(R) \wedge R(a,y) \wedge R(y,z)) \wedge R(x,b) \leq R(x,b) \wedge R(a,z) \leq R'(x,z)$.

(h) follows similarly. We established (e)–(h), therefore (4) is true. Cf. Lemma 1 (1) for any $x,y,z \in X$ we have $Ord_\Omega(R) \leq (R'(x,y) \wedge R(y,z)) \to R'(x,z)$, from where

$$Ord_\Omega(R) \leq \bigwedge_{x,y,z \in X} [(R'(x,y) \wedge R'(y,z)) \to R'(x,z)] = Trans(R')$$

From (1), (2) and (3) we deduce $Ord_\Omega(R) \leq Ord_\Omega(R')$ therefore $R \preccurlyeq R'$. We can see that $R'(a,b) = R(b,a) \vee (R(b,b) \wedge R(a,a))$ and $R(b,a) < Ord_\Omega(Q) \leq Ord_\Omega(R)$ (since $Q \preccurlyeq R$). Then, by applying Proposition 6 (6):

$R(b,a) < Ord_\Omega(R) \leq R(b,b) \wedge R(a,a) \leq R'(a,b)$.

It follows $R \neq R'$, contradicting the maximality of R. We conclude $Ord_\Omega(Q) \leq SC(R)$, therefore the theorem is proved. □

Remark 9. By Applying Lemma 5 we see that Theorem 3 is a particular case of Theorem 8.

Bibliography

[1] R. Bělohlávek, *Fuzzy Relational Systems. Foundations and Principles*, Kluwer, 2002.

[2] U. Bodenhofer, *Similarity–based Generalizations of Fuzzy Orderings Preserving the Classical Axioms*, International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, Vol. 3, pp. 593–610, 2000.

[3] U. Bodenhofer, F. Klawonn, *A Formal Study of Linearity Axioms for Fuzzy Orderings*, Fuzzy Sets and Systems, Vol. 145, pp. 323–354, 2004.

[4] S. Gottwald, *Fuzzy Sets and Fuzzy Logic*, Vieweg, Braunschweig, 1993.

[5] P. Hájek, *Methamathematics of Fuzzy Logic*, Kluwer, 1998.

[6] U. Höhle, N. Blanchard, *Partial Ordering in L–undeterminate sets*, Information Sciences, Vol. 35, pp. 135–144, 1985.

[7] E. P. Klement, R. Mesiar, E. Pap, *Triangular Norms* , Kluwer, 2000.

[8] M. Richter, *Revealed Preference Theory*, Econometrica, Vol. 34, pp. 635–645, 1966.

[9] I. J. Rudas, J. Fodor, *Information Aggregation in Intelligent Systems using Generalized Operators*, International Journal of Computers, Communications and Control, Vol. 1, pp. 47–57, 2006.

[10] E. Szpilrajn, *Sur l'Extension de l'Ordre Partiel*, Fundamenta Mathematicae, Vol. 16, pp. 386–389, 1930.

[11] L. A. Zadeh, *Similarity Relations and Fuzzy Orderings*, Information Sciences, Vol. 3, pp. 177–200, 1971.

Irina Georgescu
Academy of Economic Studies
Department of Economic Cybernetics
Piata Romana No 6, R 70167, Oficiul Postal 22
Bucharest, Romania
E-mail: irina.georgescu@csie.ase.ro

Irina Georgescu received her PhD in Economics (Information Systems) from Abo Akademi University, Turku, Finland in 2005. Since then she is a teaching assistant at the Department of Economic Cybernetics, Academy of Economic Studies, Bucharest, Romania. Between 2007–2008 she was a postdoctoral researcher at Abo Akademi University, Turku, Finland. She is the author of about 30 scientific publications and of a monograph issued by Springer. Her research interests are mainly in the area of soft computing techniques, consumer theory and social choice and welfare economics.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 343-352

Analysis and Design on Key Updating Policies for Satellite Networks

Yuxuan Ji, Hengtai Ma, Gang Zheng

Abstract: Satellite networks are becoming increasingly important because of the exciting global communication services they provide. Key management policies have been successfully deployed in terrestrial networks to guarantee the information security. However, long propagation, storage and computation constraints bring new challenges in designing efficient and cost-effective key updating policies for satellite networks. Based on the structure and communication features of satellite networks, a dynamic key management model for satellite networks (DKM-SN) is presented, which includes certificates owned by each satellite, primary keys and session keys both of which are shared between two satellites. Furthermore, a protocol is designed for updating certificates for satellites; different policies for updating primary and session keys are studied and their efficiency and security are analyzed and compared. In addition, simulation environment for satellite networks is built and the key updating processes are implemented in Walker constellation. From the simulation results, further contrasts on key updating time and storage costs between the applications of IBM hybrid key management model (HKMM) and DKM-SN in satellite networks are presented. Finally, important suggestions in designing key updating policies are given.
Keywords: key updating, satellite networks, model, protocol, simulation

1 Introduction

Satellite networks are composed of various kinds of communication satellites, vehicles and constellations. It contains both satellite-to-satellite and satellite-to-ground links. Satellite networks integrate terrestrial systems and all sorts of satellites which are deployed in different orbits with diverse tasks. Nowadays, satellite networks are increasingly used in the long-distance information transmission services. In order to ensure the message confidentiality, integrity and nonrepudiation, as well as efficiency of communication, a key management mechanism should be used to provide data encryption, authentication and key distribution and updating services for satellite communication. Key management model defines the entities in the services, the categories and relationships of the keys, and the key updating protocols and algorithms. In contrast with terrestrial networks, satellite networks are subject to dynamic network topology, long propagation delay, as well as low computing and storage capabilities of satellites. Due to these constraints, efforts should be made to decrease the key updating time in order to reduce the communication cost. Besides, for those keys used to encrypt large amount of information, the key updating protocols should be based on symmetric encryption techniques for the sake of computation cost. The storage cost on satellites should also be reduced by efficiently lowering the number of the keys.

Currently, key management policies for terrestrial networks are comparatively sophisticated. There are generally three kinds of key management policies, including those that use symmetric key encryption techniques, public key encryption techniques and combination of the two. For example, Kerberos [1] uses symmetric key encryption techniques and a KDC (Key Distribution Center); both symmetric and public key encryption techniques are adopted in IBM hybrid key management model (HKMM) in which three kinds of keys including session key, primary key and public key are used. Based on different network features and environments, the number of entities and encryption techniques involved in updating a certain type of key may be very different. For instance, though both for updating the session key, Neuman-Stubblebine protocol [3] includes three parties while Janson-Tsudik protocol [4] involves only

two. Besides, the encryption technique used to update session key can be either symmetric or public. However, all above key management policies are only applicable in terrestrial networks.

Some problems concerning key management policies in satellite networks have been studied. After studying a series of possible security threats, CCSDS showed the urgency and necessity to implement security policies in satellite networks, such as key management, authentication, access control, etc [5]. Ayan Roy-Chowdhury analyzed some problems that occur during encryption and key distribution processes when applying IPSec and SSL into satellite networks [6]. Their discussion was based on a hybrid satellite network with a single satellite component and several ground terminals. Cruickshank designed an authentication and key establishment protocol for two satellite users who need to encrypt data and voice information [7]. Since the public encryption technique was used, the method was applicable in situations where the satellites are only used to relay messages rather than implement public encrypting operations which entail a large amount of computation cost. Tanya Vladimirova et al. introduced some security services required on satellites, proposed an on-board security architecture and AES fault-tolerant mechanism [8]. However, current literature seldom studies the key management policy for satellite networks with communication links between satellites. We focus on this issue and mainly discuss the categories of keys, the protocols for updating keys and the key updating efficiencies of different policies.

HKMM uses session key, primary key and public key. However, the session key updating in HKMM may cause unendurably long propagation if directly applied in satellite networks. In order to solve this problem, a dynamic key management model applicable in satellite networks (DKM-SN)is presented, which also includes session keys, primary keys and public keys. Based on DKM-SN, a protocol is designed for updating public keys in satellite networks. By further studying the protocols such as Neuman-Stubblebine, Janson-Tsudik and improved Beller-Yacobi protocols [9], the efficiency and security issues are analyzed in designing policies for updating primary keys and session keys in satellite networks. Finally, the differences on time and storage costs between HKMM and DKM-SN are shown through simulations under the satellite network environment. The rest of this paper is organized as follows. In section 2, we discuss DKM-SN, detailing the designing principles of all three parts including public key, primary key and session key updates. Section 3 presents the simulation results under satellite network environment. We conclude with a short summary and extensions on future work in Section 4.

2 Key Management Model

In satellite networks, communication process should be kept secret in order to ensure the message confidentiality. Besides, robust and secure protocols are indispensable so that the communication process can resist well-known attacks, such as arrogating, playback, modification and so on. Therefore, we design a DKM-SN to ensure the secrecy, authenticity and integrity of messages, and at the same time decrease the time, storage and computation costs with best efforts when providing key updating services. DKM-SN consists of public, primary and session keys, which are divided based on their functions. Firstly, every satellite has a pair of public and private keys, and a certificate issued by CA (Certificate Authority), all of which are updated through communication between a certain satellite and CA. Secondly, there is a primary key shared between a pair of satellites and it is updated with their public key information (including public and private keys and certificates). Besides, two satellites also need to share a session key when they communicate with each other and the session key is updated by the primary key shared between them. The differences between DKM-SN and HKMM are: (1) HKMM uses KDC while DKM-SN does not; (2) each primary key in HKMM is shared between satellite and KDC while it is shared between two satellites in DKM-SN; (3) in order to lower storage cost, dynamic primary key updating policy is adopted in DKM-SN, which means that for some pairs of satellites their primary keys exist only when they need to establish session keys to communicate.

Figure 1 shows HKMM on left and DKM-SN on right. P denotes primary key, S for session key, T for terminal in terrestrial networks, and V for satellite. We can see that in HKMM the primary keys

are shared between each terminal and KDC while the session keys are shared between every pair of terminals. For instance, in HKMM, session key S1 can be established by terminal 3, terminal 4 and KDC using primary keys P3 and P4. In DKM-SN, both the primary and session keys are shared between two satellites. For instance, session key S21 needs to be established and shared only by two satellites V2 and V3 using their primary key P2.

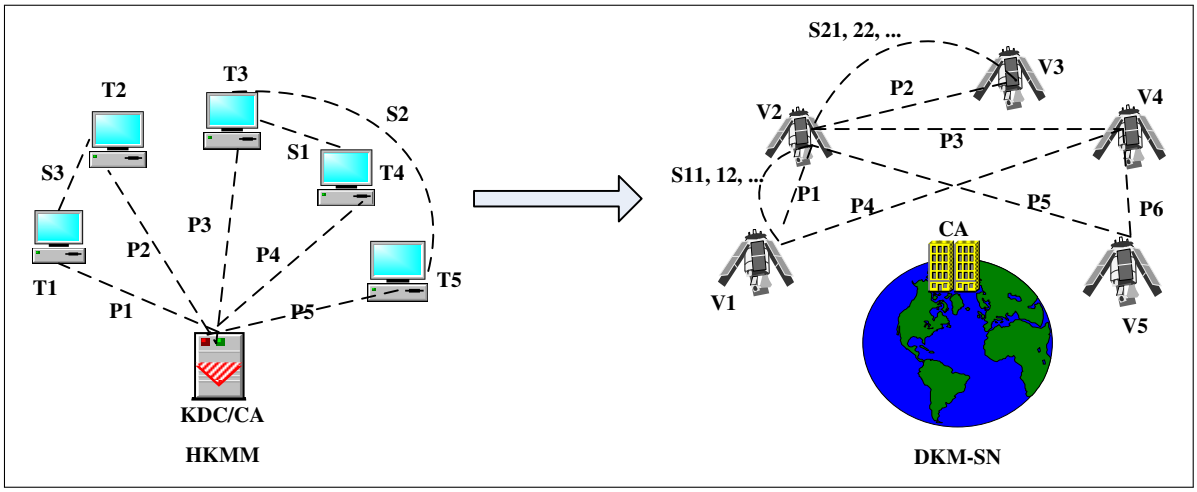


Figure 1: Key management models: HKMM and DKM-SN

2.1 Certificate Updating

Generally, an entity's public key information is updated by itself before it expires. Based on PKIX standards [10], we design a proper protocol for updating public key information.

1. $B \rightarrow M : B, PK_B, Cert(B)$
2. $M \rightarrow B : M, PK_M, [Cert(M)], PK'_M, E(PK_M^{-1}, h(PK'_M, M, B))$
3. $B \rightarrow M : Cert(M)'$

B, M - CA and satellite M; PK_X, PK_X^{-1}, $Cert(X)$ - old public key, private key and certificate of satellite X; PK'_X, $(PK'_X)^{-1}$, $Cert(X)'$ - new public key, private key and certificate of satellite X; $E(K,Y)$ - public key encryption with key K and plaintext Y; $h(Y)$ - hush function; $[Y]$ - Y is optional.

The execution of this protocol between CA and satellite proceeds as follows:

1. CA sends its own identifier, public key and certificate to satellite and informs the satellite to start updating its public key information.
2. Upon receiving message from CA, the satellite does the followings: (1) check validity of CA's certificate; (2) generate a new pair of public and private keys for itself; (3) get a signature with its old private key and send this signature, its identifier, certificate and both old and new public keys to CA.
3. Upon receiving messages from satellite, CA does the followings: (1) check the validity of satellite's old certificate; (2) check satellite's signature; (3) generate a new certificate for the satellite's new public key.
4. Upon receiving new certificate from CA, satellite can check its validity with CA's public key.

Note that the execution can continue only when the checks in the last step are successful. Now we consider some possible attacks on CA and satellite when running the protocol.

1. Attacks on CA

   (1) In the second step of our protocol, it is possible that adversary may arrogate satellite M by sending M's certificate and a public key the adversary generated. However, the adversary has to send a signature at the same time. Since we assume that the adversary doesn't know the satellite's old private key before it expires, then the adversary could not get a correct signature and would be detected by CA.

   (2) Also in the second step, the adversary may try to replay a signature signed by the satellite's private key which has expired. Obviously, only if both satellite and CA use their valid public key information that hasn't expired, our protocol could resist this playback attack.

2. Attacks on satellite

   (1) Since satellite M's new public key is sent without encryption, the adversary may generate a certificate for M by signing M's identifier and the new public key with its own private key, and then sends this false certificate to M in the third step. However, this attack would fail after satellite has received the certificate and checks its validity using CA's public key.

   (2) As an alternative, in the third step the adversary could also replay a certificate that has expired. However, the fact is that in order to keep the freshness of every run of the protocol we require that new public and private keys are different from before. For this reason, the expired certificate would be proved invalid when the satellite checks it.

The above discussion is suitable for a satellite. When updating the public key information for CA, we should first update it before it expires and then update certificates for all the satellites using CA's updated certificate.

2.2 Primary Key Updating

Although all satellites have their own public key information, we avoid using it to encrypt data due to the high complexity and low efficiency of public key encryption algorithms. In most systems, public key information is used to establish symmetric keys so that encryption on data could be faster. We consider two strategies for establishing symmetric keys with public key information: (1) using public key information, two satellites directly establish a shared session key to encrypt data; (2) they first establish a primary key between them with their public key information and then use this primary key to establish their session keys. Both primary key and session keys are symmetric keys shared between the two satellites. In the first strategy, though the cost of data encryption is low, the cost of frequent session key updates remains high. As for the second strategy, since both of frequent data encryption and session key updates adopt symmetric key encryption techniques, it is more efficient than the first one. Note that the primary keys are updated much less frequently than session keys.

HKMM includes primary keys and a KDC in its system. There is a unique primary key shared between KDC and each terminal (or entity). Hence, the number of primary keys in HKMM is n.

In DKM-SN, the second strategy is adopted, but no KDC is deployed in our network and each unique primary key is shared between two satellites. We build DKM-SN in this way for the following reasons: (1) KDC may become a bottleneck since all the session key updates have to pass KDC; (2) satellite networks are typical of long propagation. Since the session key updating process involving three parties (KDC and two satellites) has to be finished within at least 4 steps while that involving two parties (two satellites) in DKM-SN can be achieved within 3 steps, the former would be much slower than the latter. In

satellite networks, even one trip may lead to unendurably long propagation due to the multi-hop routing and long space distance between two satellites.

As for the communication protocol when updating primary keys, improved Beller-Yacobi protocol [9] is suggested. This sophisticated protocol is qualified for updating a symmetric key with public key information and engages only two entities. Besides, this protocol is suitable for the satellites that have limited computing power.

2.3 Session Key Updating

Session keys are distributed in the network to encrypt the large amount of data, such as images, languages, commands and so on. According to the previous analysis, session key in DKM-SN is shared between two satellites and updated by the primary key between the same pair of satellites. Session keys must be updated more frequently than certificate and primary keys because they are used more often. In this way, the possibility of ciphertext-only attacks can be decreased. Obtaining an old session key is not once and for all, since the attacker must intercept and analyze new ciphertexts in order to get new session keys.

As for the protocols for updating session keys, security and efficiency are of most importance. First of all, the protocol should be able to resist well-known attacks, such as replay, modification, typing, reflection and so on. Besides, it should not cost too much time and storage. Based on satellite network features, such as long propagation and limited resources, Janson-Tsudik 2PKDP [4] is suggested for updating session keys. This protocol is illustrated as follows.

1. $A \rightarrow B : A, N_{ab}$
2. $B \rightarrow A : AUTH_{K_{ab}}(N_{ab}, K_{ba}, B), E_{K_{ab}}(N'_{ba}) \oplus K_{ba}$
3. $A \rightarrow B : ACK_{K_{ab}}(N_{ab}, K_{ba}, A)$

This key establishment protocol contains the minimum cost of computation and numbers of messages and steps as proved in [4].

A detailed comparison of the time and computation costs between session key updating protocol with KDC and that without KDC is presented as follows. As for session key updates with KDC, we use Neuman-Stubblebine protocol [3] which also has the minimum number of steps involved.

According to Janson-Tsudik 2PKDP, we get the calculation time C_{jt} in a single entity and overall session key establishment time T_{jt}:

$$\begin{aligned} T_{jt} &= 3 * T_{ab} + 4 * T_{mac} + 2 * T_{es} + 2 * T_{\oplus} \\ C_{jt} &= 2 * T_{mac} + T_{es} + T_{\oplus} \end{aligned} \tag{1}$$

T_{mac} - calculation time of $MAC()$ function; T_{es} - calculation time of symmetric encryption; $T_{\oplus}$ - calculation time of XOR operation; T_{ab} - propagation delay between satellites A and B.

Since $T_{\oplus} \leq$ any other item, so we get:

$$\begin{aligned} T_{jt} &\approx 3 * T_{ab} + 4 * T_{mac} + 2 * T_{es} \\ C_{jt} &\approx 2 * T_{mac} + T_{es} \end{aligned} \tag{2}$$

As for Neuman-Stubblebine protocol, we get the calculation time C_{ns} for a single node and overall session key establishment time T_{ns} similarly:

$$\begin{aligned} T_{ns} &= (2 * T_{ab} + T_{ac} + T_{bc}) + 8 * T_{es} \\ C_{ns} &= 8 * T_{es} \end{aligned} \tag{3}$$

T_{ac} - propagation delay between KDC and satellite A; T_{bc} - propagation delay between KDC and satellite B.

Because propagation delay is much larger than the calculation time of either encryption or $MAC()$ in satellite networks, we conclude from the above estimation that both the calculation time cost of a single node and overall session key updating time cost of Janson-Tsudik 2PKDP are much less than those of Neuman-Stubblebine.

For the three-party session key establishment policy, KDC may become a bottleneck in the system. Fortunately, the two-party session key establishment policy could avoid this drawback.

3 Simulations

In order to verify the effectiveness and suitability of DKM-SN in satellite network environment, a proper simulation environment for satellite networks is very important and necessary. Based on the simulation system for distributed satellite networks [11], we design a simulation scenario and implement the experiments. We set up a Walker constellation consisting of 20 MEO satellites. There are 4 orbital planes on each of which 5 satellites are equally deployed. Each plane is of an inclination of 75° and orbit height 14163km.

In certificate updating simulation, the protocol designed in section 2.1 is applied and 160-bit ECDSA [12] is implemented as signature algorithm. In primary key updating, improved Beller-Yacobi protocol is applied. ECDSA and ECIES [13] are used as signature and encryption algorithms respectively. In session key updating, Janson-Tsudik 2PKDP and 128-bit AES are implemented.

3.1 Certificate Updating Simulation

Dynamic Topology

In this part, we examine the influences of topological changes on the certificate updates for a certain satellite.

Dynamic changes of satellite network topology lead to the variations of routing tables in satellites. Therefore, the time cost of certificate updating for a certain satellite varies with time. Figure 2 shows both the overall and computation time costs of certificate updates for a certain satellite in different periods of time.

We update a satellite's certificate every 10 seconds. Figure 2 shows the time cost during 500 seconds. The longest updating time for the satellite is 834.200ms. Routing information shows that in the longest updating data transmission from the satellite to CA entails 5 hops and thus the three-step interaction between them entails 15 hops all together. The minimum time cost is 167.45ms when there is only 1 hop between satellite and CA and so it costs 3-hop effort to finish the certificate updating process. Based on the result, we suggest update certificates when communication between satellite and CA needs the minimum number of hops.

Static Topology

In this part, we examine the certificate updates of different satellites in Walker constellation during a fixed period of time.

Figure 3 shows both the overall and computation time costs of certificate updates for different satellites during a fixed period of time in Walker constellation. We can see that there are huge gaps of time costs among the satellites since the number of hops between CA and the satellites are very different. In the simulation, for example, there are 5 hops between CA and NO.6 satellite while 1 hop between CA and NO.4.

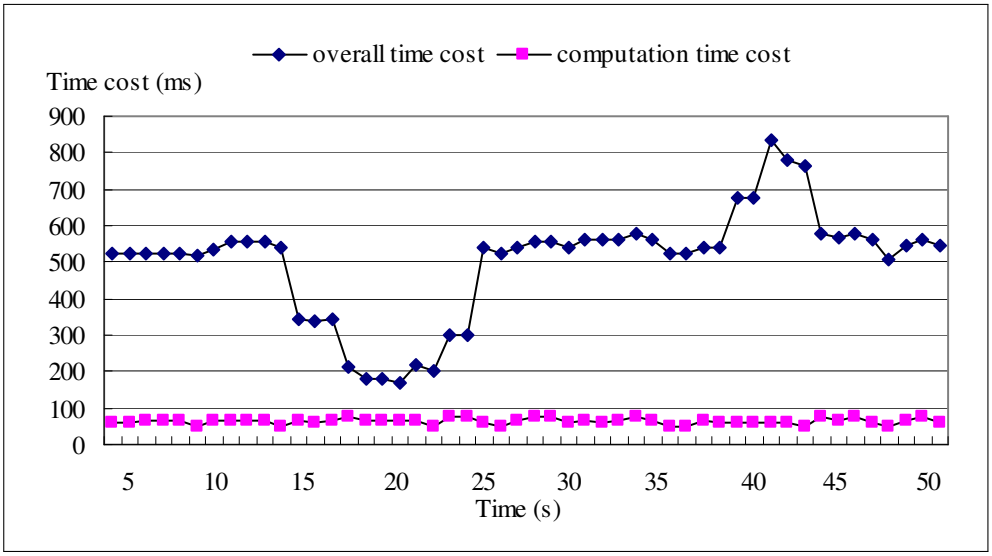


Figure 2: Time cost of certificate updating of a certain satellite in different periods of time

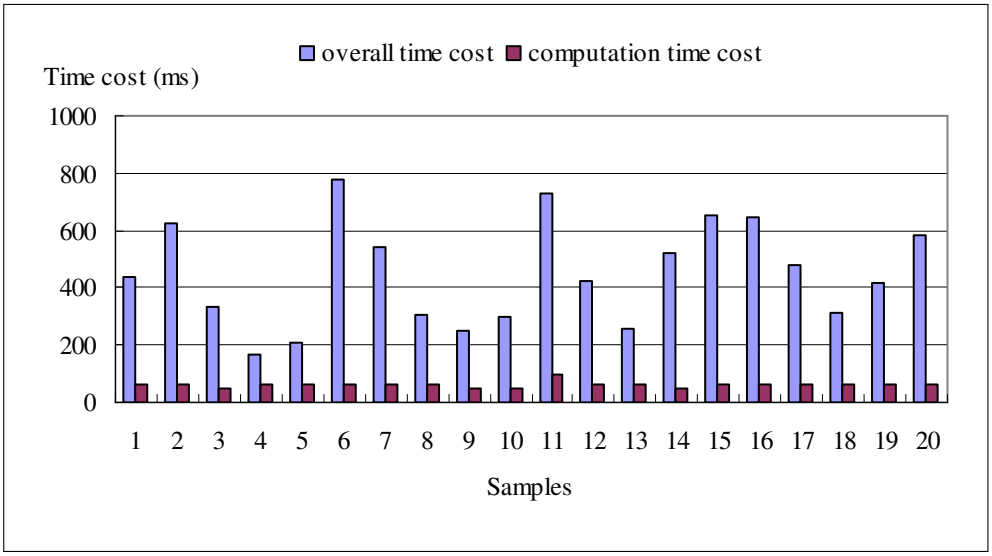


Figure 3: Time costs of certificate updates of all satellites in the same period of time

Besides, from Figure 2 and Figure 3 we can see that the computation cost for different satellites varies little. It is a small fraction of the overall time cost of certificate updates, ranging from 5% to 40%.

3.2 Comparison of Computation Time Costs

Figure 4 shows the computation costs of all three kinds of key updates in Walker Constellation. 20 samples are presented for each kind of updates.

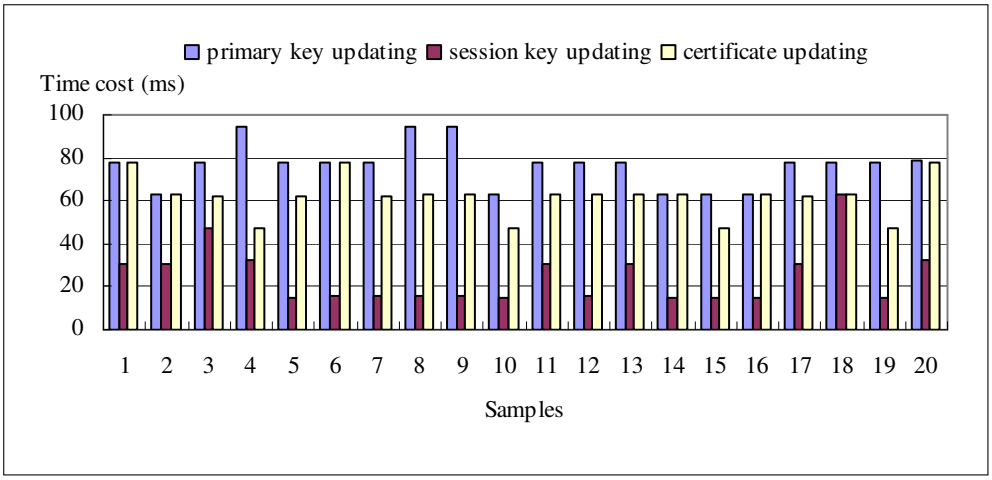


Figure 4: Computation costs of three kinds of key updates

We see from Figure 4 that session key update costs least, which is suitable for the most frequently updated session keys. The average time cost of certificate updates is less than that of primary key updates mainly due to the extra public key encryption in the improved Beller-Yacobi protocol.

3.3 Comparison between Key Management Models

Comparing DKM-SN with HKMM, we mainly have the following conclusions. Firstly, the time costs for session key updates in DKM-SN are much less than that in HKMM. Secondly, DKM-SN will contain more primary keys than HKMM if all primary keys are pre-distributed between all pairs of satellites. Since the storage capacity is limited in satellite, we recommend that for some pairs of satellites their primary keys exist only when necessary. Therefore, if we want to establish a session key between two satellites while they share no primary key, we need to first establish a primary key with their public key information and then establish the session key with their primary key. A much detailed comparison between HKMM and DKM-SN is presented as follows.

Suppose the total number of satellites is n; the average time cost of primary key updates is z.

In DKM-SN, let a denote the maximum number of primary keys, namely $a = \frac{n\times(n-1)}{2}$; h denotes the average time cost of session key updates when there is a shared primary key between every pair of satellites; x denotes the practical number of primary keys; y denotes the average time cost of session key updates when x pairs of satellites share primary keys.

In HKMM, suppose the average time cost of session key updates is $h+\delta$, $\delta > 0$.

Based on DKM-SN, we have,

$$y = \frac{x\times h+(a-x)\times(z+h)}{a} = (z+h) - \frac{z}{a}\times x \tag{4}$$

Obviously, given $x = 0$, we get $y = z+h$. This means there is no primary key pre-shared in DKM-SN and for every two satellites before establishing a session key we should first set up a primary key. Similarly, given $x = a$, we get $y = h$. Under this condition, each pair of satellites owns a primary key.

If $y > h + \delta$, we get $x < \frac{a \times (z-\delta)}{z}$.

So we conclude that when $n < x < \frac{a \times (z-\delta)}{z}$, HKMM is superior to DKM-SN in both average time cost of session key updates and storage cost.

Specifically, based on the simulation data, we have $z \approx 400ms$, $h \approx 360ms$, $\delta \approx 120ms$, $n = 20$.

Under this condition, $y = (z+h) - \frac{z}{a} \times x \approx 760 - 2 \times x$. Figure 5 shows the relationship between average time cost of session key updates and the number of primary keys in DKM-SN.

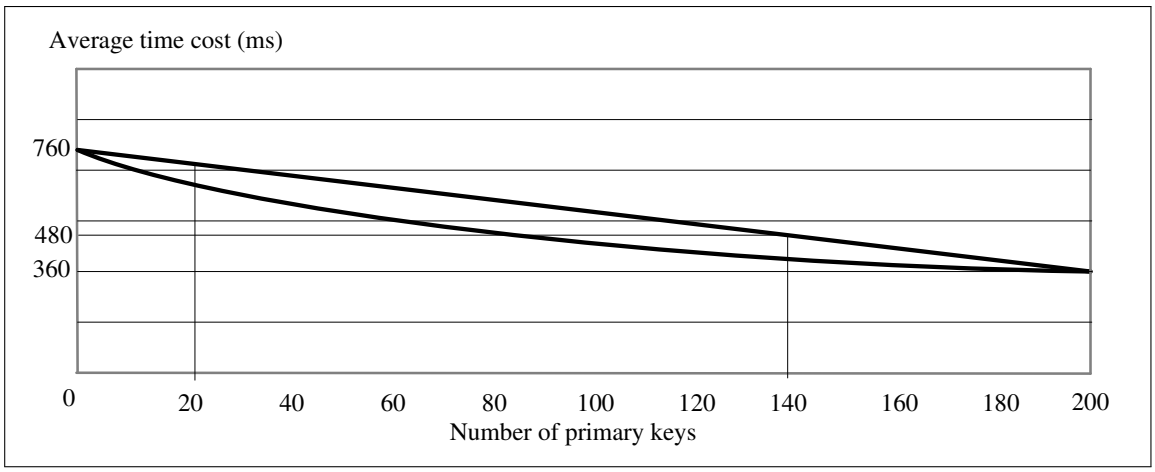


Figure 5: Average time cost of session key updates in DKM-SN

Given $y > h + \delta$, namely $760 - 2 \times x > 480$, we get $x < 140$. We can see from the linear function shown in Figure 5 that if the number of primary keys ranges from 20 to 140, HKMM is better than DKM-SN in both time and storage costs. When $140 < x \leq 200$, average time cost of session key updates in DKM-SN is less than that in HKMM while the number of keys is more than that in HKMM. Furthermore, if we can deploy the primary keys for those satellites that communicate most frequently, the average time cost function of the number of primary keys will be a concave function shown in Figure 5. We can make estimation and decision given real on-board data in satellite networks. For examples, if satellites could store the maximum number of primary keys, pre-allocating a primary key for every pair of satellites is the best choice, which is especially suitable for a network with a small number of satellites.

4 Summary and Conclusions

In this paper, a new key management model called DKM-SN is designed for satellite networks. A protocol for updating satellite certificate is designed and the efficiency and security of different policies for updating primary and session keys are analyzed. The performance is shown when updating three kinds of keys and a further contrast between DKM-SN and HKMM is given in both time cost and storage cost. The efficiency and applicability of different key updating policies are discussed under different on-board storage constraints and time requirements. In the future, we will further discuss the problems in designing efficient, low-cost and secure key management models for satellite networks.

Bibliography

[1] J. Kohl, C. Neuman, The Kerberos Network Authentication Service (V5), *http://www.ietf.org/rfc/rfc1510.txt*, RFC 1510, 1993.

[2] V. Le, S. M. Matyas, D. B. Johnson and J. D. Wilkins, A Public Key Extension to the Common Cryptographic Architecture, *IBM System Journal*, Vol. 32, pp. 461-485, 1993.

[3] B. C. Neuman and S. G. Stubblebine, A Note on the Use of Timestamps as Nonces, *ACM Operating Systems Reviews*, Vol. 27, pp. 10-14, 1993.

[4] Philippe Janson and Gene Tsudik, Secure and Minimal Protocols for Authenticated Key Distribution, *Computer Communications*, Vol. 18, pp. 645-653, 1995.

[5] CCSDS, Security Threats Against Space Missions, *Washington: Informational Report*, CCSDS 350.1-G-1, Green Book, Issue 1, 2006.

[6] A. Roy-Chowdhury *et al.*, Security Issues in Hybrid Networks with a Satellite Component, *IEEE Wireless Communications*, Vol. 12, pp. 50-61, 2005.

[7] H S Cruickshank, A Security System for Satellite Networks, *Fifth International Conference on Satellite Systems for Mobile Communications and Navigation*, London: IEE, pp. 187-190, 1996.

[8] Tanya Vladimirova, Roohi Banu and Martin N. Sweeting, On-Board Security Services in Small Satellites, *MAPLD International Conference*, Washington: NASA Office of Logic Design, 2005.

[9] C. Boyd and A. Mathuria, Key Establishment Protocols for Secure Mobile Communication: a Selective Survey, *Lecture Notes in Computer Science*, Vol. 1438, pp. 344-355, 1998.

[10] J. Schaad, M. Myers, Public-Key Infrastructure (X.509), *www.ietf.org/html.charters/pkix-charter.html*, IETF, PKIX 2797.

[11] X. Ying, Z. Gang, Modeling and Distributed Simulation for Satellite Networks, *Computer Simulation*, Vol. 25, pp. 65-69, 2008.

[12] ANSI X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standards Institute, 1999.

[13] ANSI. X9.63, Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, American National Standards Institute, 2001.

Yuxuan Ji
Institute of Software, Chinese Academy of Sciences
National Key Laboratory of Integrated Information System Technology
4# South Fourth Street, Zhong Guan Cun, Beijing 100190, P.R. CHINA
E-mail: jiyuxuan06@gmail.com

Hengtai Ma
Institute of Software, Chinese Academy of Sciences
National Key Laboratory of Integrated Information System Technology
4# South Fourth Street, Zhong Guan Cun, Beijing 100190, P.R. CHINA
E-mail: htma@ios.cn

Gang Zheng
Institute of Software, Chinese Academy of Sciences
National Key Laboratory of Integrated Information System Technology
4# South Fourth Street, Zhong Guan Cun, Beijing 100190, P.R. CHINA
E-mail: gangzhengcn@yahoo.com.cn

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 353-365

Quality of Service Scheduling in Real-Time Systems

Audrey Marchand, Maryline Chetto

Abstract: In this paper, we deal with dynamic scheduling components integrating new Quality of Service (QoS) functionalities into a Linux-based real-time operating system. In our approach, periodic tasks allow occasional deadline violations within given bounds specified according to the Skip-Over task model. Hence, every task has a minimal QoS guarantee which is expressed by the ratio of periodic task instances which must complete before their deadline. The work stated here provides two on-line scheduling algorithms, namely RLP and RLP/T, which enhance the existing Skip-Over algorithms. More specifically, the proposed algorithms aim at improving the actual QoS observed for periodic tasks (which is always greater or equal to the QoS guarantee). These novel scheduling techniques rely on the EDL (Earliest Deadline as Late as possible) scheduling strategy. Simulation results show the performance of RLP and RLP/T with respect to basic Skip-Over algorithms. Finally, we present the integration of these QoS scheduling services into CLEOPATRE open-source component library, a patch to Linux/RTAI.
Keywords: real-time, dynamic scheduling, quality of service, periodic tasks, component-based systems, Linux-based systems

1 Introduction

Software environments, and more precisely operating systems have still difficulties to meet the special demands of multimedia applications. In particular, multimedia applications have real-time constraints which are not handled properly by general-purpose operating systems. In order to meet the requirements imposed by multimedia applications on processor scheduling, we have to turn to the temporal stringency of real-time systems. Real-time systems are those in which the time at which the results are produced is important. The correctness of the result of a task is not only related to its logic correctness, but also to when the results occur.

Traditional classification of real-time systems stands for three classes to characterize the real-time requirement of such systems : hard, soft and firm. In hard real-time systems, all instances must be guaranteed to complete within their deadlines. In those critical control applications, missing a deadline may cause catastrophic consequences on the controlled system. For soft systems, it is acceptable to miss some of the deadlines occasionally. It is still valuable for the system to finish the task, even if it is late. In firm systems, tasks are also allowed to miss some of their deadlines, but, there is no associated value if they finish after the deadline. Typical illustrating examples of systems with firm real-time requirements are multimedia systems in which it is not necessary to meet all the task deadlines as long as the deadline violations are adequately spaced.

A prominent strategy for performing resource management for multimedia systems is QoS-driven management, in which quality requirements such as resolution and frame rate are translated into resource requirements such as computation burst frequencies and durations. This resource information is then used for admission testing and resource reservation. The motivation for this translation from application level requirements to resource requirements is to guarantee a given QoS. The complexity of both the QoS space and the resource space suggests that perfect characterization is hard to achieve, so it would be desirable to have a scheduling policy that would adapt to changes in user QoS requirements. Such policy should strive to achieve the desired QoS, in an environment with variable resources, as well as complex and variable application demands.

In this paper, we address the problem of the dynamic scheduling of periodic tasks with firm constraints. The scope of the paper is to maximize the actual QoS of periodic tasks by maximizing the number of instances which complete before their deadline. The remainder of this paper is organized in the following manner. Next section introduces existing approaches for scheduling firm real-time systems. Then we present relevant background material about both the Skip-Over model and the EDL scheduling algorithm. More particularly, we give the definition of RTO and BWP scheduling algorithms, which are based on the Skip-Over model. The functioning and optimality of the EDL algorithm is also outlined. Further, we describe the proposed algorithms, namely RLP and RLP/T, as an enhancement of the BWP algorithm, based on the EDL scheduling mechanism. Moreover, we present a model of a real-world problem to show the practical interest of our work. The performance analysis of both RLP and RLP/T, in terms of task completions, is reported after. Then, we describe the integration of these QoS components into Linux/RTAI. Finally, in section 8, we summarize our contribution.

2 Related work

There have been some previous approaches to the specification and design of real-time systems that tolerate occasional losses of deadlines. Hamdaoui and Ramanathan in [7] introduced the concept of *(m,k)-firm* deadlines to model tasks that have to meet m deadlines every k consecutive invocations. Their algorithm uses a *distance-based priority (DBP)* scheme to increase the priority of a job in danger of missing more than m deadlines over a sliding window of k requests for service. Moreover, algorithms such as VDS [17] and DWCS [19] are provably superior to DBP in meeting (m,k) service requirements for a number of specific and non-trivial situations.

Similar to (m,k)*-firm* scheduling is the work introduced by Koren and Shasha [8] with the notion of *skip factor*. If a task has a skip factor of s, it will have one invocation skipped out of s. It is a particular case of the *(m,k)-firm* model where $m = k - 1$. They reduce the overload by skipping some task invocations, thus exploiting skips to increase the feasible periodic load. This approach gives a solution to the scheduling problem of overloaded systems, while representing a system Quality of Service requirement for real-time applications. Broadly speaking, the Skip-Over scheduling algorithms guarantee the timing correctness of the real-time application. One interesting result is that making optimal use of skips is a NP-hard problem. There are also examples of (m,k)-hard schedulers [1], but most such approaches require off-line feasibility tests, to ensure predictable service.

In [3, 4], Caccamo and Buttazzo follow this work by scheduling hybrid task sets consisting of skippable periodic and soft aperiodic tasks. They propose and analyze an algorithm, based on a variant of Earliest Deadline First (EDF) scheduling, in order to exploit skips under the Total Bandwith Server (TBS). In previous works [10, 12], we have considered the same approach but using the Earliest Deadline as Late as possible server (EDL). These results have led us to propose a raw version of the RLP algorithm (idle time schedule based on red tasks only) [11].

West and Poellabauer in [16] proposed a *windowed lost rate*, that specifies a task can tolerate x deadlines missed over a finite range or window, among consecutive y instances. In [2], Bernat *et al.* introduce a general framework for specifying tolerance of missed deadlines under the definition of *weakly hard* constraints.

3 Theoretical Background

3.1 The Skip-Over model

We are here interested in the problem of scheduling periodic tasks which allow occasional deadline violations (*i.e.*, skippable periodic tasks), on a uniprocessor system. We assume that tasks can be preempted and that they do not have precedence constraints. A task T_i is characterized by a worst-case

Task	T_0	T_1	T_2	T_3	T_4
c_i	3	4	1	7	2
p_i	30	20	15	12	10

Table 1: A basic periodic task set

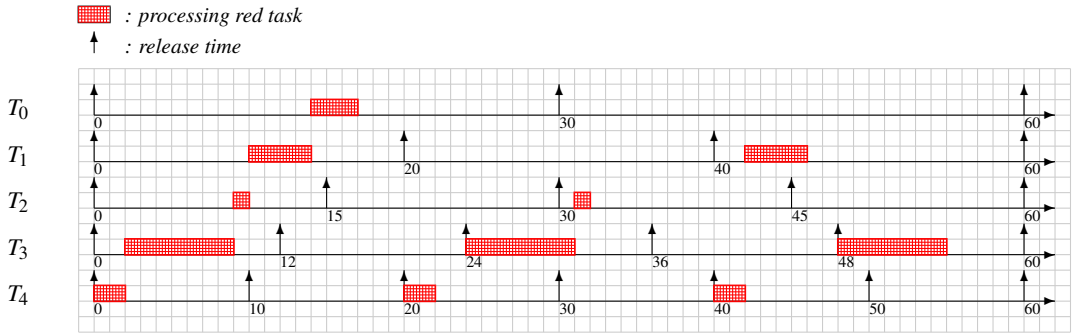


Figure 1: RTO scheduling algorithm ($s_i = 2$)

computation time c_i, a period p_i, a relative deadline equal to its period, and a skip parameter s_i, which gives the tolerance of this task to missing deadlines. The distance between two consecutive skips must be at least s_i periods. When s_i equals to infinity, no skips are allowed and T_i is equivalent to a hard periodic task. One can view the skip parameter as a QoS metric (the higher s_i, the better the quality of service).

A task T_i is divided into instances where each instance occurs during a single period of the task. Every instance of a task can be red or blue [8]. A red task instance must complete before its deadline; a blue task instance can be aborted at any time. However, if a blue instance completes successfully, the next task instance is still blue.

Red Tasks Only (RTO) algorithm

The first algorithm proposed by Koren and Shasha is the Red Tasks Only (RTO) algorithm. Red instances are scheduled as soon as possible according to Earliest Deadline First (EDF) algorithm, while blue ones are always rejected. Deadline ties are broken in favor of the task with the earliest release time. In the deeply red model where all tasks are synchronously activated and the first $s_i - 1$ instances of every task T_i are red, this algorithm is optimal. RTO is illustrated in Figure 1 using the task set $\mathcal{T} = \{T_0, T_1, T_2, T_3, T_4\}$ of five periodic tasks whose parameters are described in Table 1. Tasks have uniform skip parameter $s_i = 2$ and the total processor utilization factor $U_p = \sum \frac{c_i}{p_i}$ is equal to 1.15.

As we can see, the distance between every two skips is exactly s_i periods, thus offering only the minimal guaranteed QoS level for periodic tasks.

Blue When Possible (BWP) algorithm

The second algorithm studied is the Blue When Possible (BWP) algorithm which is an improvement of the first one. Indeed, BWP schedules blue instances whenever their execution does not prevent the red ones from completing within their deadlines. In that sense, it operates in a more flexible way. Deadline ties are still broken in favor of the task with the earliest release time. Figure 2 shows an illustrative example of BWP scheduling using the task set previously described in Table 1.

Compared with RTO, more task instances complete successfully with BWP. We observe that five violations of deadline relative to blue task instances occur at time instants $t = 24$ (task T_3), $t = 30$ (tasks T_2 and T_4) and $t = 60$ (tasks T_3 and T_4), thus reducing the QoS.

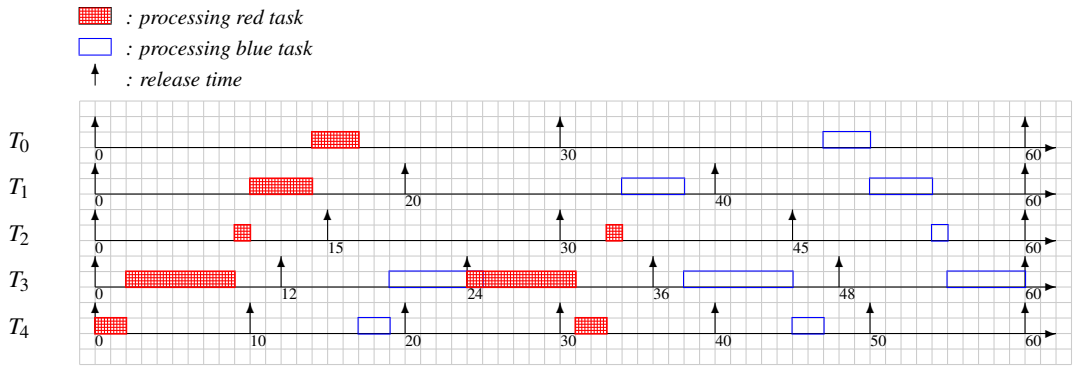


Figure 2: BWP scheduling algorithm ($s_i = 2$)

3.2 The EDL algorithm

The definition of the Earliest Deadline as Late as possible (EDL) algorithm makes use of some results presented by Chetto and Chetto in [5]. Under EDL, periodic tasks are scheduled as late as possible. An accurate characterization of the idle times during which the processor is not occupied is necessary. The authors introduced an *availability function* f_Y^X defined with respect to a task set Y and a scheduling algorithm X. $f_Y^X(t) = 1$ if the processor is idle at t, 0 otherwise.

So, for any instants t_1 and t_2, value of $\int_{t_1}^{t_2} f_Y^X(t)dt$ denoted by $\Omega_Y^X(t_1,t_2)$ gives the total idle time in $[t_1,t_2]$. f_Y^{EDL} can be described by means of the following two vectors:

- $\mathcal{K}$, called *static deadline vector*, represents the times at which idle times occur and is constructed from the distinct deadlines of periodic tasks.
- $\mathcal{D}$, called *static idle time vector*, represents the lengths of the idle times relating to time instants of vector $\mathcal{K}$.

The complexity of the EDL algorithm is $O(Kn)$ where n is the number of periodic tasks, and K is equal to $\lfloor \frac{R}{p} \rfloor$, where R is the longest deadline, and p is the shortest period [13]. We also recall the fundamental property relative to the optimality of EDL [5]:

Theorem 1. *Let X be any preemptive scheduling algorithm and $\mathcal{A}$ a set of independent aperiodic tasks. For any instant t,*

$$\Omega_{\mathcal{A}}^X(0,t) \leq \Omega_{\mathcal{A}}^{EDL}(0,t) \tag{1}$$

We give now an illustrative example of the computation of the idle times performed by EDL. Consider the periodic task set $\mathcal{T} = \{T_1, T_2\}$ consisting of two periodic tasks $T_1(3,10)$ and $T_2(3,6)$. The $f_{\mathcal{T}}^{EDL}$ computation produced at time zero is described in Figure 3.

The authors in [5] described how the EDL algorithm can be applied, first to the decision problem that arises when a sporadic time critical task occurs and requires to be run at an unpredictable time and secondly, to the scheduling problem that arises in a fault tolerant system using the Deadline Mechanism [**?**] for which each task implements primary and backup copies (the processor time reserved for the execution of the backup copies is realized with EDL and is reclaimed as soon as the primary task executes successfully).

In next sections, we are interested in using EDL first to simulate a schedule (RLP implementation) and then to derive a measure required for deciding whether a blue task can be accepted (RLP/T implementation).

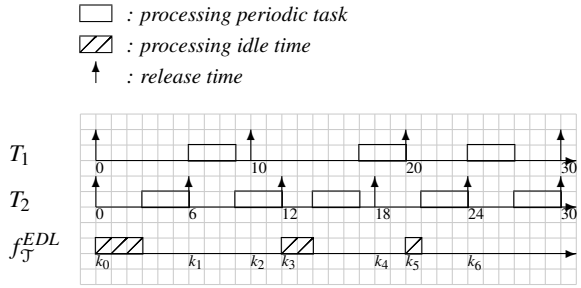


Figure 3: $f_{\mathcal{T}}^{EDL}$ computation produced at time zero

4 The Proposed Algorithms

4.1 Red tasks as Late as Possible (RLP)

The main drawback of BWP relies on the fact that blue task instances are executed as background tasks. This leads to abort partially or almost completely executed blue task instances, thus wasting processor time.

Algorithm outline

The objective of RLP algorithm is to bring forward the execution of blue task instances so as to minimize the ratio of aborted blue instances, thus enhancing the actual QoS (*i.e.,* the total number of task completions) of periodic tasks. From this perspective, RLP scheduling algorithm, which is a dynamic scheduling algorithm, is specified by the following behaviour:

1. if there are no blue task instances in the system, red task instances are scheduled as soon as possible according to the EDF (Earliest Deadline First) algorithm.

2. if blue task instances are present in the system, these ones are scheduled as soon as possible according to the EDF algorithm (note that it could be according to any other heuristic), while red task instances are processed as late as possible according to the EDL algorithm.

Deadline ties are always broken in favor of the task with the earliest release time. The main idea of this approach is to take advantage of the slack of red periodic task instances. Determination of the latest start time for every red request of the periodic task set requires preliminary construction of the schedule by a variant of the EDL algorithm taking skips into account [12]. In the EDL schedule established at time τ, we assume that the instance following immediately a blue instance which is part of the current periodic instance set at time τ, is red. Indeed, none of the blue task instances is guaranteed to complete within its deadline. Moreover, Silly-Chetto in [13] proved that the online computation of the slack time is required only at time instants corresponding to the arrival of a request while no other is already present on the machine. In our case, the EDL sequence is constructed not only when a blue task is released (and no other was already present) but also after a blue task completion if blue tasks remain in the system (the next task instance of the completed blue task has then to be considered as a blue one). Note that blue tasks are executed in the idle times computed by EDL and are of same importance beside red tasks (contrary to BWP which always assigns higher priority to red tasks).

Illustrative example

Consider once again the periodic task set $\mathcal{T}$ defined in Table 1. The relating RLP scheduling is illustrated in Figure 4. In this example, we can see that, thanks to RLP scheduling, the number of

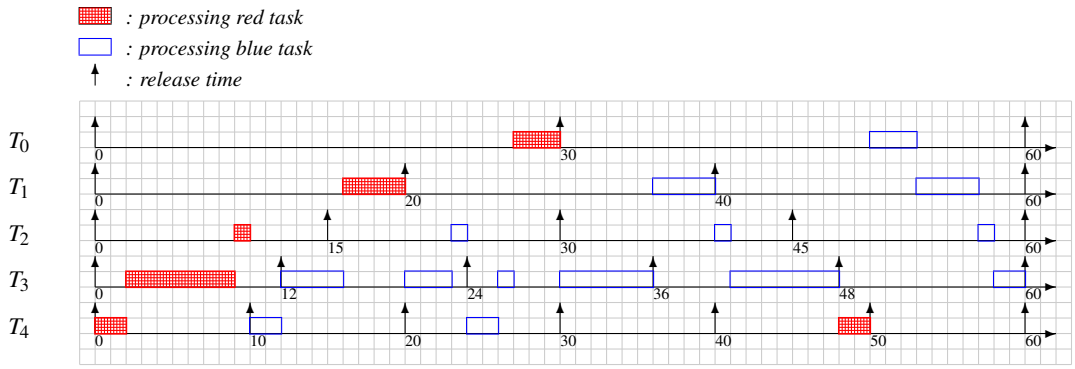


Figure 4: RLP scheduling algorithm ($s_i = 2$)

violations of deadline relative to blue task instances has been reduced to three. They occur at time instants $t = 40$ (task T_4), and $t = 60$ (tasks T_3 and T_4). Observe that T_3 first blue task instance which failed to complete within its deadline in the BWP case (see Figure 2), has enough time to succeed in the RLP case, since the execution of T_1 and T_0 first red task instances is postponed. Until time $t = 10$, red task instances are scheduled as soon as possible. From time $t = 10$ to the end of the hyperperiod (defined as the least common multiple of task periods), red task instances do execute as late as possible in the presence of blue task instances, thus enhancing the actual QoS of periodic tasks.

4.2 Red tasks as Late as Possible with blue acceptance test (RLP/T)

The main drawback of RLP relies on the fact that this algorithm attempts to execute blue task instances as soon as possible, at the risk of aborting them before their completion, thus generating a processor time wasting. This assessment led us to propose a novel algorithm named RLP/T (Red tasks as Late as Possible with blue acceptance Test).

Algorithm outline

Red tasks as Late as Possible with blue acceptance test (RLP/T) algorithm is designed to maximize the actual QoS of periodic task sets defined under skip constraints.

It acts as follows: red tasks enter straight the system at their arrival time whereas blue tasks integrate the system upon acceptance. Once they have been accepted, blue tasks are scheduled as soon as possible together with red tasks. Upon acceptance, blue tasks are again of same importance beside red tasks. Deadline ties are always broken in favor of the task with the earliest release time.

Whenever a new blue task enters the system, the idle times are computed using the EDL scheduler. In the EDL schedule established at time τ, we assume that the instance following immediately a blue instance which is part of the current periodic instance set at time τ, is also blue. Indeed, all blue task instances previously accepted at τ are guaranteed by the schedulability test they passed successfully. This one checks whether there are enough idle times to accommodate the new blue task within its deadline, as described in the following section.

Acceptance test of blue tasks under RLP/T

Now, we are ready to present the new feasibility test algorithm for the RLP scheduling scheme which, given any occurring blue task B is capable of answering the question "Can B be accepted ?". Notice that B will be accepted if and only if there exists a valid schedule, *i.e.*, a schedule in which B will execute by its deadline while red periodic tasks and blue tasks previously accepted, will still meet their deadlines. Let τ be the current time which coincides with the arrival of a blue task B. Upon arrival, task $B(r,c,d)$

is characterized by its release time r, its execution time c, and its deadline d, with $r+c \leq d$. We assume that the system supports several blue tasks at time τ. Each of them has been accepted before τ and has not completed its execution at time τ. Let denote by $\mathcal{B}(\tau) = \{B_i(c_i(\tau), d_i), i\text{=}1 \text{ to } blue(\tau)\}$ the blue task set supported by the machine at τ. Value $c_i(\tau)$ is called dynamic execution time and represents the remaining execution time of B_i at τ. A deadline occurs at d_i. We assume that $\mathcal{B}(\tau)$ is ordered such that $i < j$ implies $d_i \leq d_j$.

The acceptance test of blue tasks within a system involving RLP skippable tasks presented below in Theorem 2, is based on the one established by Silly-Chetto and al. [14] for the acceptance of sporadic requests occurring in a system consisting of basic periodic tasks (*i.e.*, without skips).

Theorem 2. *Task B is accepted if and only if, for every task* $B_i \in \mathcal{B}(\tau) \cup \{B\}$ *such that* $d_i \geq d$, *we have* $\delta_i(\tau) \geq 0$, *with* $\delta_i(\tau)$ *defined as:*

$$\delta_i(\tau) = \Omega^{EDL}_{\mathcal{T}(\tau)}(\tau, d_i) - \sum_{j=1}^{i} c_j(\tau) \tag{2}$$

$\delta_i(\tau)$ is called slack of task B_i at time τ which represents the maximum units of time during which the task could not be served by the processor without missing its deadline. $\Omega^{EDL}_{\mathcal{T}(\tau)}(\tau, d_i)$ denotes the total units of time that the processor is idle in the time interval $[\tau, d_i]$. The total computation time required by blue tasks within $[\tau, d_i]$ is given by $\sum_{j=1}^{i} c_j(\tau)$

The procedure that implements the acceptance test calls for the EDL algorithm for the computation of the total idle times which will be used to compute the slack of blue tasks. Then, this slack is compared to zero. Thus, the acceptance test proposed in this paper runs in $O(\lfloor \frac{R}{p} \rfloor n + blue(\tau))$ in the worst-case, where n is the number of periodic tasks, R is the longest deadline, p is the shortest period, and $blue(\tau)$ denotes the number of active blue tasks at time τ, whose deadline is greater or equal to the deadline of the occurring task. Note that this acceptance test could be implemented in $O(n + blue(\tau))$ by considering and maintaining to update additional data structures using slack tables, as proved in [15].

Illustrative example

RLP/T scheduling is illustrated in Figure 5 with the periodic task set $\mathcal{T}$ defined in Table 1. It is easy to see that RLP/T improves on both RLP and BWP. Only two violations of deadline relative to blue task instances are observed: at time instants $t = 40$ (task T_4) and $t = 60$ (task T_3). The acceptance test contributes to compensate for the time wasted in starting the execution of blue tasks which are not able to complete within their deadline. As we can observe, in the RLP case (see Figure 4), T_3 blue instance released at time $t = 48$ is aborted at time $t = 60$ (2 units of time were indeed wasted). Note that the rejection of this blue task instance, performed with RLP/T, contributes to save time used for the successful completion of T_4 blue instance released at time $t = 50$.

In section 6, we quantify more precisely the gain of performance of RLP/T upon RLP, BWP and RTO.

5 Applying theory to real-world problems

5.1 Multimedia applications

In order to understand the importance of CPU scheduling in multimedia real-time applications, it is useful to place the issue in context. Multimedia implies a certain amount of data to be handled within a specified time frame and requires a tremendous amount of resources to accommodate. For multimedia applications to function correctly, there must be a steady stream of data for the output devices to process. For the viewer to perceive continuous media such as movies or music, the output devices have to output

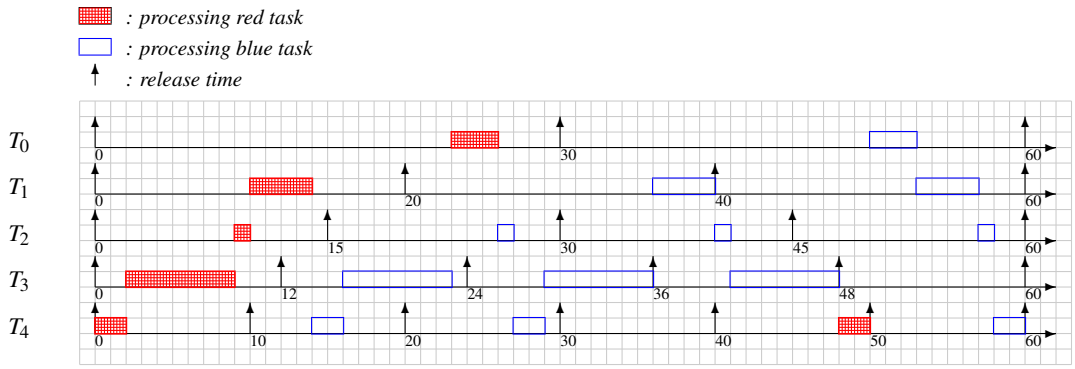


Figure 5: RLP/T scheduling algorithm ($s_i = 2$)

new media within strict time constraints (e.g. 30 frames per second for video applications). Given this observation, one gets a better understanding of the crucial role of CPU scheduling in such applications.

Consider a simplified model of a real-time telesurveillance application, as represented in Figure 6. The relevant tasks are the *acquiring tasks* and the *display task*. Video data are first captured and digitized through video capture devices such as video cameras. Then, each video capture task "Acq_i" reads the input video buffer relative to its camera, thus periodically acquiring incoming frames. Downstream from the chain, another task named "*Display*" is in charge of continuously consuming frames from an output frame buffer and sending the acquired video frames to a final display device composed of various telesurveillance screens.

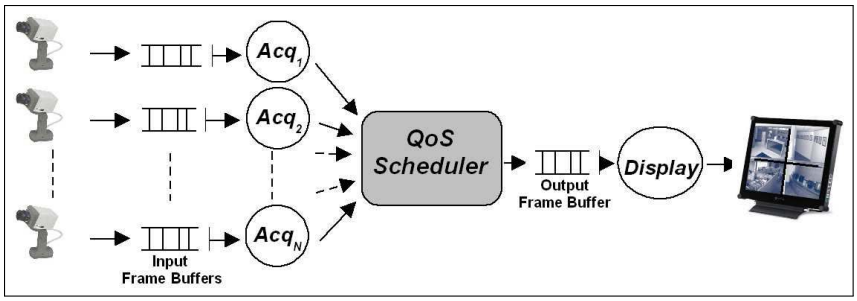


Figure 6: Simplified architecture of a real-time telesurveillance application

One important problem part of the management of telesurveillance systems is the data refreshing rate on the display device. Indeed, by definition, such a system must provide pictures as recent as possible to be useful. If there is no data for the output devices to process, there is buffer underflow. The media application will stall and wait for new data to be provided. Naturally, buffer underflow should be avoided whenever possible.

Scheduling implies multiplexing a resource among several tasks to ensure all throughput requirements are met. In the present case, the problem consists in ensuring an acceptable refreshing rate (*i.e.* a guaranteed QoS level).

5.2 Dynamic QoS scheduling

For multimedia applications, the CPU scheduler determines the resulting quality of service. The more CPU cycles allocated to a task, the more data can be produced, thus providing a better quality output. In the previous example, more CPU cycles produce more frames, thus allowing more frames per second to be displayed on the workstation monitor. However, if several videos are executing at the same time there may not be enough CPU cycles available to produce all the video frames requested. In this

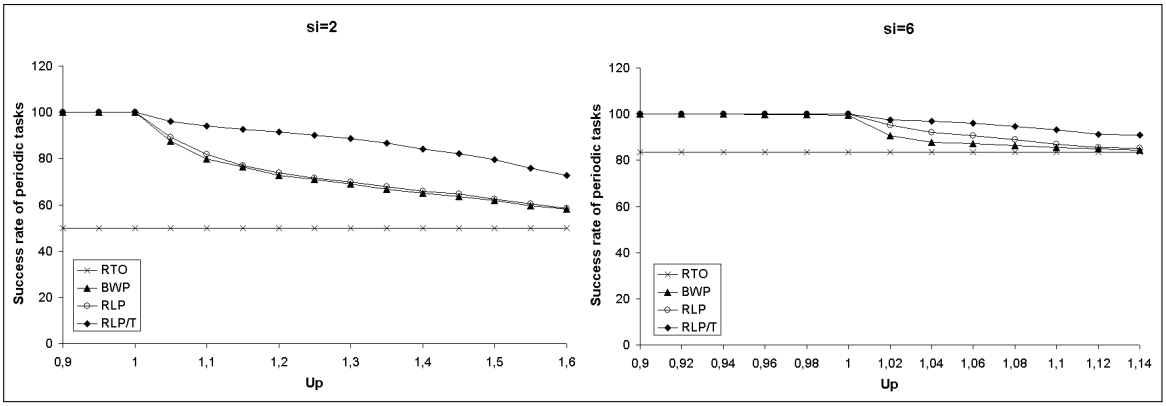


Figure 7: QoS of periodic tasks with low and high skip parameters ($s_i = 2$ and $s_i = 6$)

overloaded situation, the quality of service (*i.e.* frame rate) is reduced to a lower acceptable level, which results for instance to display an image at 15 frames per second instead of 30 frames per second. The resulting video just appears less smooth during the transient overload period.

Now, let us consider RLP or RLP/T algorithms for scheduling the application described in Figure 6. At initialization time, the application specifies a desired average rate of execution by appropriately setting the skip parameter s_i of each "Acq_i" task. RLP (or RLP/T) scheduler can be viewed as an EDF priority-based scheduler coupled with a skip-over rate regulator. That ensures every task not to be executed below a specified rate, whatever is the CPU workload. RLP and RLP/T superiority over RTO and BWP effectively results in a higher and guaranteed frame rate on the workstation monitors.

6 Simulation Study

In this section, we summarize the results of a simulation study which compares the performance of the different QoS scheduling algorithms. The objective is to maximize the actual QoS level of periodic tasks, i.e., the ratio of periodic tasks which complete before their deadline.

Experiments also evaluate the impact of the skip value for each algorithm, namely RTO, BWP, RLP and RLP/T.

6.1 Simulation context

The simulation context includes 50 periodic task sets, each consisting of 10 tasks with a least common multiple equal to 3360. Tasks are defined under QoS guarantees specified by uniform s_i. Their worst-case execution time is randomly generated and depends on the input setting of the periodic load U_p. Deadlines are equal to the periods and greater than or equal to the computation times. Simulations have been processed over 10 hyperperiods.

6.2 Varying the periodic load

Measurements rely on the ratio of periodic tasks which complete before their deadline. The evaluation is done varying the periodic load U_p. The results obtained for $s_i = 2$ (one instance every two can be aborted) and $s_i = 6$ (one instance every six can be aborted) are described on Figure 7.

From the graphs, we can say that BWP, RLP and RLP/T outperform the RTO model in which the QoS level is still constant whatever is the periodic load applied. For $s_i = 6$, the actual QoS (which corresponds to the QoS guarantee) remains constant at a rate of 5/6=83%. The advantage of RLP over BWP is slight for low skip parameters, and more significant for high skip parameters. We note that the performance of

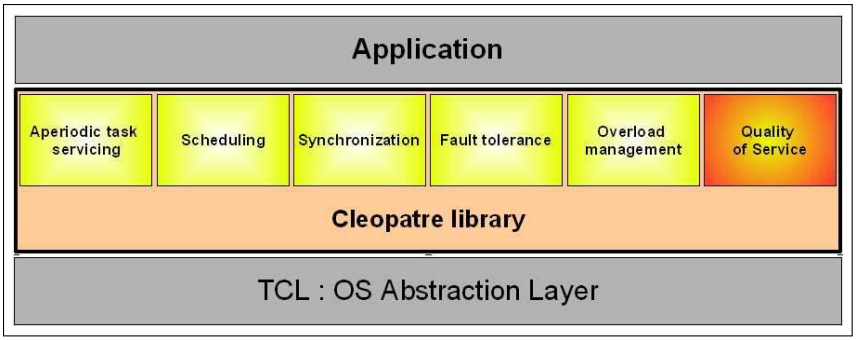


Figure 8: The CLEOPATRE framework

BWP and RLP is dramatically worse than the one achieved by RLP/T. This result was expected because both BWP and RLP attempt to schedule blue instances that have not enough time for completing within their deadlines. This wasted time is not saved for executing other blue instances with closer deadline. In contrast, RLP/T finds a way of saving this CPU time by implementing an acceptance test for blue instances. We can observe that this gain of performance is all the more significant as the periodic load U_p is higher. For instance, in Figure 7, for $U_p \geq 120\%$, RLP/T enjoys more than factor $\frac{1}{4}$ success rate advantage over BWP. Moreover, we observe a very low gradient for the RLP/T curve which is not the case for other models. For $U_p = 150\%$ and $s_i = 2$, actual QoS levels for RTO and RLP/T are respectively equal to 50% and 84%, which figures the great predominance of RLP/T over RTO.

The variation of the skip parameter value shows that, for wide loads, the actual QoS of periodic tasks is all the more improved with RLP/T that the QoS constraint is smaller. For instance, for $U_p = 110\%$, RLP/T applied to periodic tasks with $s_i = 2$ will successfully process twice as many periodic instances over BWP, as with periodic tasks with $s_i = 6$. As we can see, the major difference in the performance between RLP/T and BWP appears not only for heavy loads but also for small value of s_i.

7 Integration into a Linux-based system

7.1 CLEOPATRE library

RTO, BWP, RLP and RLP/T algorithms have been integrated into a library of free software components called CLEOPATRE (Software Open Components on the Shelf for Embedded Real-Time Applications) [6]. This library, part of a French National Project [1], was designed to provide more efficient and better service to real-time applications. The purpose was to enrich the real-time facilities of real-time Linux versions, such as RTLinux [18] or RTAI [9]. RTAI was the solution adopted for this project because we wanted the CLEOPATRE components to be distributed under the LGPL[2] license which is also the one used in the RTAI project.

The CLEOPATRE library whose framework is shown in Figure 8 offers selectable COTS (Commercial-Off-The-Shelf) components dedicated to dynamic scheduling, aperiodic task service, resource control access and fault-tolerance. Components are totally independent from the kernel and the hardware. Reusability of the components with another hardware and OS is made possible by just adapting the OS abstraction layer in the TCL component. This component hides the specific features of each platform, so that the run-time components can be implemented in a portable fashion and adapted to the target's processor architecture and board.

RTO, BWP, RLP and RLP/T can be found in a new shelf called "Quality of Service"'. Final users

[1] work supported by the French research office, grant number 01 K 0742
[2] Lesser General Public License

can then build their own customized applications through the flexible and easy-to-use interface provided by the CLEOPATRE framework.

8 Conclusions

This paper pointed out the need of more flexible scheduling solutions for real-time applications dealing with multimedia and active monitoring systems. Our main contribution was actually to propose and validate new scheduling algorithms, namely RLP and RLP/T. Their purpose is to enhance the QoS of periodic tasks that allow skips (i.e the ratio of task instances that do execute within their deadline) while providing a QoS guarantee (i.e the ratio of task instances that must complete within their deadline). We considered a real-world problem (*i.e.* a multimedia application) to bring to light how these algorithms can be implemented in practice in order to provide a better QoS. Simulation results show that the improvements with both RLP and RLP/T are quite significant compared with basic algorithms. These new QoS functionalities are available under Linux/RTAI. Our future work includes extending these QoS scheduling algorithms to multiprocessor systems.

Bibliography

[1] G. Bernat, A. Burns, Combining (n/m)-hard feadlines and dual priority schaduling, *18th IEEE Real-Time Systems Symposium*, pp 46-57, 1997.

[2] G. Bernat, A. Burns, A. Llamosi, Weakly-hard real-time systems, *In IEEE Transactions on Computers*, Vol. 50, No. 4, pp 308-321, 2001.

[3] G.-C. Buttazzo, M. Caccamo, Minimizing Aperiodic Response Times in a Firm Real-Time Environment, *IEEE Trans. Software Eng.*, Vol. 25, No. 1, pp 22-32, 1999.

[4] M. Caccamo, G.-C. Buttazzo, Exploiting skips in periodic tasks for enhancing aperiodic responsivess, *18th IEEE Real-Time Systems Symposium*, 1997.

[5] H. Chetto, M. Chetto, Some Results of the Earliest Deadline Scheduling Algorithm. *In Proceedings of the IEEE Transactions on Software Engineering*, Vol. 15, No. 10, pp 1261-1269, 1989.

[6] T. Garcia, A. Marchand, M. Silly-Chetto, Cleopatre: a R&D project for providing new real-time functionalities to Linux/RTAI. *5th Real-Time Linux Workshop*, 2003.

[7] M. Hamdaoui, P. Ramanathan, A Dynamic Priority Assignment Technique for Streams with (m,k)-firm deadlines. *IEEE Transactions on Computers*, Vol. 44, No. 4, pp 1443-1451, 1995.

[8] G. Koren, D. Shasha, Skip-Over Algorithms and Complexity for Overloaded Systems that Allow Skips. *16th IEEE Real-Time Systems Symposium (RTSS'95)*, Pisa, Italy, 1995.

[9] P. Mantegazza, E. Bianchi, L. Dozio, M. Angelo, D. Beal, DIAPM. RTAI Programming Guide 1.0, *Lineo Inc.*, 2000.

[10] A. Marchand, M. Silly-Chetto, QoS Scheduling Components based on Firm Real-Time Requirements, *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'05)*, Le Caire (Egypt), 2005.

[11] A. Marchand and M. Silly-Chetto, RLP: Enhanced QoS Support for Real-Time Applications, *11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05)*, Hong-Kong, 2005.

[12] A. Marchand, M. Silly-Chetto, Dynamic Real-Time Scheduling of Firm Periodic Tasks with Hard and Soft Aperiodic Tasks. *Journal of Real-Time Systems*, Vol. 32, No. 1-2, pp 21-47, 2006.

[13] M. Silly-Chetto, The EDL Server for Scheduling Periodic and Soft Aperiodic Tasks with Resource Constraints, *Journal of Real-Time Systems*, Vol. 17, pp 1-25, 1999.

[14] M. Silly-Chetto, H. Chetto, N. Elyounsi, An Optimal Algorithm for Guranteeing Sporadic Tasks in Hard Real-Time Systems. *IEEE Symposium on Parallel and Distributed Processing*, pp 578-585, 1990.

[15] T. Tia, J. Liu, J. Sun, R. Ha, A Linear-Time Optimal Acceptance Test for Scheduling of Hard Real-Time Tasks, *Technical report, Department of Computer Science*, University of Illinois at Urbana-Champaign, IL, 1994.

[16] R. West, C. Poellabauer, Analysis of a Window-constrained scheduler for real-time and best-effort packet streams, *21st IEEE Real-Time Systems Symposium*, Orlando, USA, 2000.

[17] R. West, Y. Zhang, K. Schwan, C. Poellabauer, Dynamic window-constrained scheduling of real-time streams in media servers, *IEEE Trans. on Computers*, Vol. 53, pp. 744-759, 2004.

[18] V. Yodaiken, The RTLinux Approach to Real-Time, *FSMLabs Inc.*, 2004.

[19] Y. Zhang, R. West, X. Qi, Avirtual deadline scheduler for window-constrained service guarantees, *Tech. Rep. 2004-013*, Boston University, 2004.

Audrey Marchand
University of Nantes
Laboratoire d'Informatique de Nantes Atlantique
2, rue de la Houssinière - BP 92208
44322 Nantes Cedex 03,
FRANCE
E-mail: audrey.marchand@univ-nantes.fr

Maryline Chetto
University of Nantes
Institut de Recherche en Communications et Cybernétique de Nantes
1, rue de la Noe
44321 Nantes Cedex 03
FRANCE
E-mail: maryline.chetto@univ-nantes.fr

Audrey Marchand graduated in Computer Engineering at the Ecole polytechnique of the University of Nantes (France), in 2002. After getting a Master Degree in Applied Computer Science at Ecole Centrale de Nantes in 2003, she received the PhD degree in October 2006 from the University of Nantes. From October 2006 to August 2007, she hold a Post-Doc researcher position at the Polytechnic University of Valencia, Spain. She is currently an Associate Professor at the University of Nantes, France. Her research interests include real-time scheduling theory, OS service mechanisms, quality of service guarantees in real-time systems, and Linux-based real-time OSes and applications.

Maryline Chetto received the degree of Docteur de 3ème cycle in control engineering and the degree of Habilitée à Diriger des Recherches in Computer Science from the University of Nantes, France, in 1984 and 1993, respectively. From 1984 to 1985, she held the position of Assistant professor of Computer Science at the University of Rennes, while her research was with the Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes. In 1986, she returned to Nantes and is currently a professor with the Institute of Technology of the University of Nantes. She is conducting her research at IRCCyN. Her main research interests include scheduling and fault-tolerance technologies for real-time applications. She has published more than 60 journal articles and conference papers in the area of real-time operating systems. She is the leader of a French national R&D project, namely Cleopatre, supported by the French government, which aims to provide free open source real-time solutions.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 366-373

Ant Colony Solving Multiple Constraints Problem: Vehicle Route Allocation

Sorin C. Negulescu, Claudiu V. Kifor, Constantin Oprean

Abstract: Ant colonies are successfully used nowadays as multi-agent systems (MAS) to solve difficult optimization problems such as travelling salesman (TSP), quadratic assignment (QAP), vehicle routing (VRP), graph coloring and satisfiability problem. The objective of the research presented in this paper is to adapt an improved version of Ant Colony Optimisation (ACO) algorithm, mainly: the Elitist Ant System (EAS) algorithm in order to solve the Vehicle Route Allocation Problem (VRAP). After a brief introduction in the first section about MAS and their characteristics, the paper presents the rationale within the second section where ACO algorithm and its common extensions are described. In the approach (the third section) are explained the steps that must be followed in order to adapt EAS for solving the VRAP. The resulted algorithm is illustrated in the fourth section. Section five closes the paper presenting the conclusions and intentions.
Keywords: Ant Colony Optimisation, Vehicle Route Allocation Problem, Multi-Agent Systems.

1 Introduction

Ant colonies are used nowadays as multi-agent systems to solve different optimization problems from the NP-hard class. Examples of such problems are traveling salesman (TSP), quadratic assignment (QAP), vehicle routing (VRP), graph colouring and satisfiability problem.

The multi-agent systems are copying some or all the characteristics of their biological counterparts (depending on how much these characteristics are helpful to the MAS solving a particular type of problem) such as [6]:

- distributed society of autonomous individuals/agents;
- fully distributed control among the agents;
- localized communications among the individual;
- stochastic agent decisions;
- system-level behaviours transcending the behavioural repertoire of the single (minimalist) agent;
- simple interaction rules.

Consequently, the overall very important features of the system are: robustness, adaptability and scalability.

The paper presents the technique of solving yet another difficult problem (VRAP) which belongs to the NP-hard class of problems. Related work, regarding the ACO and EAS algorithms, is described recently (2006-2007) in [1], [2], [3], [9]; to weaken redundancy, here details are skipped over. As a result, the rest of the paper is structured as follows: Section 2 expounds the rationale where Ant Colony Optimisation (ACO) and its common extensions are explained. Section 3 describes the steps that must be followed in order to adapt EAS for solving the VRAP. The next section concentrates on presenting the resulted algorithm. Conclusions and directions of future work (section 5) are closing the paper.

2 Rationale and Approach

The ant colony optimization algorithm (ACO), introduced by Marco Dorigo in 1992 in his PhD thesis, is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs and is inspired by the behaviour of ants in finding paths from the colony to food [8].

In all early Ant System (AS) algorithms, ants are searching for (candidate) solutions based on two main components: pheromone trails and problem-dependent heuristic information. These algorithms have suffered frequent modifications in order to improve their efficiency. Thus, the AS [5] developed into the Elitist Ant System (EAS) [4], because each ant that finds a better solution has the chance to deposit more pheromone.

Other systems that emerged from the AS are the Max-Min Ant System (MMAS) [10] and the Ant-Q [7] where the deposited pheromone amount is directly proportional to the quality of the found solution. In the Rank-Based Ant System (ASrank) all solutions are ranked accordingly to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the more optimal solutions deposit more pheromone than the less optimal solutions.

The basic VRAP consists in allocating vehicles from a fleet to different routes in such a way that the average distance travelled in a month is (optimally) equal for all vehicles.

Before designing an ACO algorithm some questions regarding important aspects of the problem must be answered first [6]:

- The representation of the problem (pheromone model): *What are the most effective state features to consider for learning "the good decisions"?*
- Heuristic variables: *What are they and what is their relative weight regarding the pheromone in the decisions?*
- Ant-routing table: *How pheromone and heuristic variables are combined together to define the goodness of a decision?*
- Stochastic decision policy: *How to make good exploration without sacrificing exploitation of promising/good actions?*
- Regarding pheromones: *Is it useful to put limits in pheromone ranges? How should the pheromone initialization and evaporation be dealt with? What is the best policy for pheromone updating: online or offline?*
- Scheduling of the ants: *How and/or how many per iteration?*
- Restarting the algorithm after stagnation: *Would it be better to restart the algorithm when stagnation is detected?*
- Daemon components: *What is the effect of local search?*

The standard form of the EAS algorithm for solving the TSP is presented in figure 1.

// Initialize pheromone trails
for (every edge i, j) {
$\tau = \tau_0$
}
// Choose the starting town for every ant
for ($k = 1; k \leq m; k$++) {
Place ant k on a randomly chosen city
}
// Initialize the best tour and length
T^+ = the shortest tour found from the beginning
L^+ = the length of the best tour
// Main loop
for ($t = 1; t \leq Tmax; t$++) {
// Compute a tour for every ant
for ($k = 1; k \leq m; k$++) {
Build tour $T_k(t)$ by applying $n-1$ times the following step:
Choose the next node (city) j with the probability
$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}(t)]^\beta}$, if $j \in J$
$p_{ij}^k(t) = 0$, if $j \notin J$
where i is the current city.
}
// Compute the tour lengths for all ants
for ($k = 1; k \leq m; k$++) {
Compute the length $L_k(t)$ of the tour $T_k(t)$ produced by ant k
}
// Update the best tour if an improved tour is found
if (an improved tour is found) {
Update T^+ and L^+
print T^+ and L^+
}
// Global update for the pheromone trails
for (every edge i, j) {
Update the pheromone trails by applying the rule:
$\tau_{ij}(t) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) + e \cdot \tau_{ij}^e(t)$, where
$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$
$\Delta\tau_{ij}^k(t) = \left\{ \frac{Q/L^k(t), if (i,j) \in T^k(t)}{0, otherwise} \right\}$
and
$\tau_{ij}^e(t) = \left\{ \frac{Q/L^+, if (i,j) \in T^+}{0, otherwise} \right\}$
}
// Calculate the intensity of the pheromone for next iteration
for (every edge i, j) {
$\tau_{ij}(t+1) = \tau_{ij}(t)$
}
}

Figure 1: Elitist Ant System pseudocode for solving the Travelling Salesman Problem.

In this paper is presented a problem that has more constraints than TSP in its basic form:

- The number of routes (destinations);
- The number of vehicles in the fleet;
- The route allocation history (the routes must be allocated in an equally distributed manner for each month - TURNUS);
- The average distance travelled by vehicles in a month (this distance must be (optimally) equal for all the vehicles - AVERAGE DISTANCE);
- The driver's rest (of two days) between trips (REST).

Because of these constraints, new variables are necessary to be included in the algorithm for solving VRAP. Moreover a different approach is necessary regarding the development of the graph used to solve the problem. In figure 2 are depicted the graphs for solving TSP and VRAP accordingly.

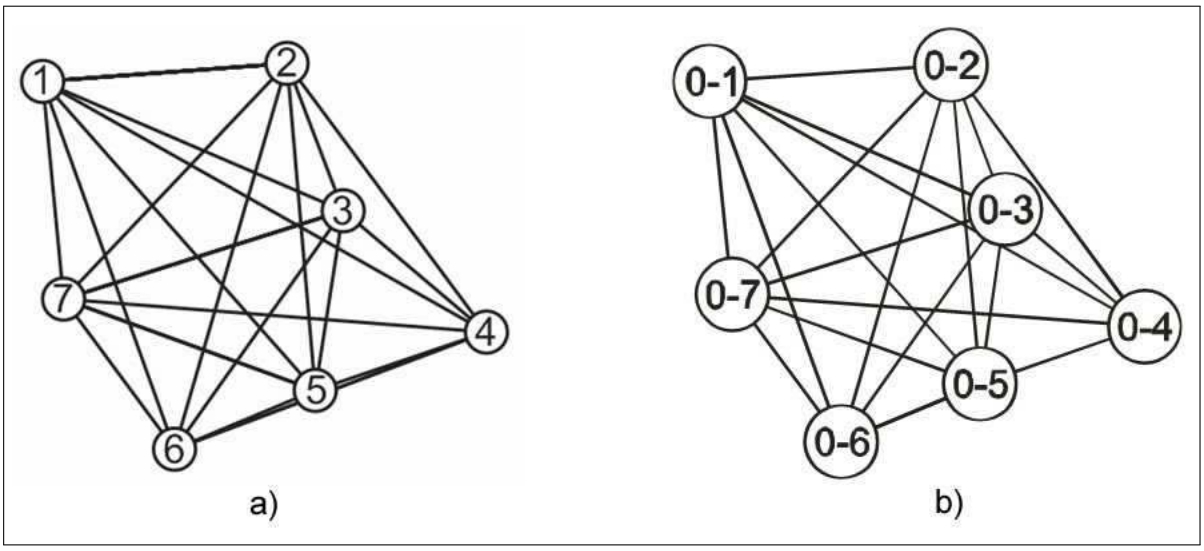


Figure 2: Comparison between the graphs for solving TSP (a) and VRAP (b).

If in the case of TSP the nodes in the graph are representing the cities that the salesman must reach, and the arches are standing for the roads (their cost being directly proportional to the distance between cities), in the case of VRAP the nodes (seen as pairs node0 and noden) are equivalent to a certain route (e.g. Bucharest - Paris) and the arches are representing the cost of allocating a vehicle to that route.
Another approach is that the node0 is a unique virtual node regardless of the name of the city that corresponds to the departure station (e.g. node0 can be Bucharest, Paris, etc.).
The cost function must include the following factors:

- The vehicle history (the cost is greater if the vehicle departed from node0 and arrived at noden more recently and smaller otherwise);
- The average distance ran by a vehicle in a month (the cost is directly proportional with the difference between the average distance and the actual distance).
- The rest (of two days) of the drivers (the cost is directly proportional with the difference between the required resting days and the actual resting days).

3 The resulted algorithm

The algorithm for solving VRAP by using the modified Elitist Ant System is presented in figure 3.

// Variables used
// T_{max} - maximum numbers of iterations
// R - number of routes
// V - number of vehicles
// H - routes history matrix
// D_{ij} - distance between cities i and j
// P_{ij} - pheromone intensity between nodes i and j
// a - number of ants used to solve the problem (a is also equal with the number of routes)
// $T_{kr}(t)$ - the matrix tour of ant k at the iteration t
// C - the vector containing ant tours costs
// T^+ - vector containing the best tour
// C^+ - the cost of the best tour
// $\alpha \simeq 0.7$ - how much the ants takes in consideration the history
// $\beta \simeq 0.3$ - how much the ants takes in consideration the pheromone intensity
// $\gamma \simeq 0.8$ - how much the ants takes in consideration the routes distances
// $\rho \simeq 0.5$ - percent of the pheromone that evaporates
// $\varepsilon \simeq 0.5$ - the multiplication value for the pheromone intensity of the elitist ant
// p_0 - minimum pheromone intensity on arches $= 10^{-6}$

// Initialize pheromone trails
for (every edge i, j) {
    $P_{ij} = p_0$
}
// Initialize the best tour and best tour cost
$T^+ =$ the best tour found from the beginning
$C^+ =$ the length of the best tour

// Main loop
for ($t = 1; t \leq Tmax; t$++) {
    // For every vehicle (every ant colony)
    for ($v = 1; v \leq V; v$++) {
        // Choose the starting town for every ant
        for ($k = 1; k \leq a; k$++) {
            Place ant k on node zero (start node)
        }
        // Compute a tour for every ant
        for ($k = 1; k \leq a; k$++) {
            Build tour $T_k^r(t)$ by applying $R-1$ times the following step:
            Choose the next node i in witch the ant k will go
            with the probability
            $p_{vi}^k(t) = \frac{[H_{vi}(t)]^\alpha \cdot [P_{vi}(t)]^\beta \cdot [V_{vi}]^\gamma}{\sum_{v \in T_k^r} [H_{vi}(t)]^\alpha \cdot [P_{vi}(t)]^\beta \cdot [V_{vi}]^\gamma}$, if $i \notin T_k^r(t)$
            $p_{vi}^k(t) = 0$, if $i \in T_k^r(t)$,
            where v is the current vechicle
        }

```
                // Compute the tour costs for all ants
                for (k = 1; k ≤ a; k++) {
                        Compute the cost C_k(t) of the tour T_k^r(t) produced by ant k
                }
                // Update the best tour if an improved tour is found
                if (an improved tour is found) {
                        Update T^+ and C^+
                        Print T^+ and C^+
                }
                // Global update for the pheromone trails
                for (every edge i, j) {
                        Update the pheromone trails by applying the rule:
                        P_ij(t) = (1 − ρ) · P_ij(t) + ΔP_ij(t) + ε · P_ij^e(t)
                }
                // Calculate the intensity of the pheromone for next iteration
                for (every edge i, j) {
                        P_ij(t + 1) = P_ij(t)
                }
        }
}
```

Figure 3: Elitist Ant System pseudocode for solving the Route Allocation Problem.

The α, β and γ parameters are controlling the relative weight of the *history*, *pheromone intensity* and the *distance* in the process of choosing the next node in the itinerary. The formula for computing the pheromone intensity between nodes i and j at the t moment is:

$P_{ij}(t) = (1-\rho)\cdot P_{ij}(t) + \Delta P_{ij}(t) + \varepsilon \cdot P^e_{ij}(t)$, where

- $(1-\rho)\cdot P_{ij}(t)$ - the pheromone quantity that evaporates on the arch between nodes i and j at the moment t;
- $\Delta P_{ij}(t) = \sum_{k=1}^{a} \Delta P^k_{ij}(t)$ - the total pheromone quantity that is deposited on the arch between nodes i and j at the moment t as sum of pheromone quantity deposited by all the ants that have used this arch;
- $\Delta P^k_{ij}(t) = \left\{ \frac{1-H_i^{T^i_k}, if (i,j) \in T^i_k(t)}{0, otherwise} \right\}$ - the pheromone quantity deposited by ant k on the arch between the nodes i and j if that arch belongs to its itinerary $(T_{ki}(t))$ at the moment t. This quantity is inversely proportional with the history of the vehicle on the route associated to the node that points to that arch;
- $\Delta P^e_{ij}(t) = \left\{ \frac{1-H_i^{T^e_k}, if (i,j) \in T^+(t)}{0, otherwise} \right\}$ - the pheromone quantity deposited by the *elitist* ant k on the arch between nodes i and j if that arch belongs to its itinerary $(T_{ki}(t) = T^+)$ at the moment t. This quantity is inversely proportional with the history of the vehicle on the route associated to the node that points to that arch.

This process repeats until a certain condition is met (number of iterations, time, the finding of an acceptable solution, etc.).

4 Conclusions and intentions

The self-organisation (as in the case of ant colonies) indirectly supports the quality of the allocation service through the lack of the distortions or the deficiencies that may appear in a centralized coordination. The resulting implemented algorithm is robust (both reliable and error tolerant). The optimisation algorithm (Elitist Ant System) proved to be perfectly adaptable to the *Vehicle Route Allocation Problem*. The ant colony algorithms are showing an obvious potential as problem-solving tool, at least in the context of simple MAS, and relevant results can be achieved on affordable hardware configurations.
The *short-range* intentions are:

- Improving the cost formula;
- Finishing the research regarding the control of pheromones and search paths;
- Speeding up the search by fine-tuning the algorithm.

The *middle-range* target is to apply the optimisation in a real-time manner by modifying the algorithm.

5 Acknowledgements

This research is supported by the Ceex 116 / 2006 project: eTransMobility - Agent Oriented System for Modelling and Optimizing the Transport of Persons; coordinated by "Lucian Blaga" University of Sibiu.

Bibliography

[1] Bărbat B.E, Moiceanu A., Pleşca S., Negulescu S.C. (2007).Affordability and Paradigms in Agent-Based Systems. *Computer Sc. J. of Moldova, 15, 2(44)*, pp.178-201.

[2] Bărbat B.E., Negulescu S.C. (2006). From Algorithms to (Sub-)Symbolic Inferences in Multi-Agent Systems. *International Journal of Computers, Communications and Control, 1, 3*, pp.5-12.

[3] Bărbat B.E., Negulescu S.C., Zamfirescu C.B. (2005). Human-Driven Stigmergic Control. Moving the Threshold. In N. Simonov (Ed.), *Proc. of the 17th IMACS World Congress (Scientific Computation, Applied Mathematics and Simulation)*, pp.86-92. Paris: ISBN 2- 915913-02-01.

[4] Bonabeau E., Dorigo M., Theraulaz G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press.

[5] Dorigo M., Maniezzo V., Colorni A. (1996). The Ant System: Optimisation by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26 (1), pp.29-42.

[6] Gambardella L.M., Di Caro G. (2005). The Ant Colony Optimization (ACO) Metaheuristic: a Swarm Intelligence Framework for Complex Optimization Tasks. Retrived 2008, from University of Bologna: First Summer School on Aspects of Complexity. Web site: http://www.cs.unibo.it/˜fioretti/AC/AC2005/docs/slides_dicaro.pdf

[7] Gambardella L.M., Dorigo M. (1995). Ant-Q: A Reinforcement Learning Approach to the Travelling Salesman Problem. In A. Prieditis and S. Russell (Ed.), *Proceedings of the Eleventh International Conference on Machine Learning* (pp.252-260). San Francisco, CA: Morgan Kaufmann.

[8] Negulescu S.C., Bărbat B.E. (2004). Enhancing the effectiveness of simple multi-agent systems through stigmergic coordination. In ICSC-NAISO (Ed.), *Fourth International ICSC Symposium on ENGINEERING OF INTELLIGENT SYSTEMS (EIS 2004)*, pp.149-156. Canada: ICSC-NAISO Academic Press.

[9] Negulescu S.C., Zamfirescu C.B., Bărbat B.E. (2006). User-Driven Heuristics for nondeterministic problems. *Studies in Informatics and Control (Special issue dedicated to the 2nd Romanian-Hungarian Joint Symp. on Applied Computational Intelligence), 15, 3*, pp.289-296.

[10] Stützle T., Hoos H.H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems, 16 (8)*, pp.889-914.

Sorin C. Negulescu
"Lucian Blaga" University of Sibiu
Faculty of Engineering
4, Emil CIORAN, IM 502
550025 Sibiu, Romania
E-mail: sorin_negulescu@yahoo.com

Claudiu V. Kifor
"Lucian Blaga" University of Sibiu
Faculty of Engineering
4, Emil CIORAN, IM 502
550025 Sibiu, Romania
E-mail: claudiu.kifor@ulbsibiu.ro

Constantin Oprean
"Lucian Blaga" University of Sibiu
Faculty of Engineering
4, Emil CIORAN, IM 502
550025 Sibiu, Romania
E-mail: constantin.oprean@ulbsibiu.ro

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 374-383

Computation Results of Finding All Efficient Points in Multiobjective Combinatorial Optimization

Milan Stanojević, Mirko Vujošević, Bogdana Stanojević

Abstract: The number of efficient points in criteria space of multiple objective combinatorial optimization problems is considered in this paper. It is concluded that under certain assumptions, that number grows polynomially although the number of Pareto optimal solutions grows exponentially with the problem size. In order to perform experiments, an original algorithm for obtaining all efficient points was formulated and implemented for three classical multiobjective combinatorial optimization problems. Experimental results with the shortest path problem, the Steiner tree problem on graphs and the traveling salesman problem show that the number of efficient points is much lower than a polynomial upper bound.
Keywords: multiple objective optimization, combinatorial optimization, complexity of computation.

1 Introduction

For all combinatorial problems cardinality of the feasible solution set grows exponentially with the problem size. For one group of combinatorial problems (e.g. the shortest path problem, the shortest spanning tree problem, assignment problem etc.) algorithms that can find a solution of a single-criterion problem in polynomial time are known. This problem class is denoted by $\mathcal{P}$. For other combinatorial problems (e.g. traveling salesman problem, Steiner tree problem, knapsack problem etc.) so called *non deterministic polynomial* algorithms exist. These problems belong to the class denoted by $\mathcal{NP}$. For this problem's class it is not proved whether polynomial algorithms exist or not. There is a third class of problems (e.g. finding all spanning trees for a given graph) for which it is known that they can be solved only by exponential algorithms. In such problems result usually consists of exponential amount of data, so the exponential time is needed just to represent them. Let's denote this class by $\mathcal{E}$.

In all known literature which concerns multiobjective combinatorial optimization (MOCO), mostly the set of Pareto optimal solutions is being observed, and it has been stated that its size can grow exponentially with the problem size. Moreover, with sufficient number of uncorrelated criteria it is possible to achieve that every feasible solution is Pareto optimal [1]. This implies that there is no efficient method of determining all Pareto optimal solutions for problems of bigger dimensions, because such a procedure would not be even $\mathcal{NP}$ hard, but strictly exponential, i.e. it would belong to class $\mathcal{E}$. Such results are confirmed for many known problems, such as: the shortest spanning tree, the shortest path problem, traveling salesman problem, assignment problem and knapsack problem [2, 3, 4, 5, 7].

In Section 2 some multiobjective optimization models are introduced. Section 3 presents the implemented algorithm which is applied in order to find all efficient points of multiobjective combinatorial optimization problems. Experimental results described in Section 4 show that the number of efficient points is even much lower than a polynomial upper bound. In Section 5 some concluding remarks are formulated.

2 Multiobjective optimization models

The general model of multiobjective optimization problem can be briefly formulated as follows.

$$\min_{x \in X} f(x) \tag{1}$$

where x is the decision variable, X is the feasible solution set, and $f = (f_1, \ldots, f_p)$ is the p-dimensional vector of objective functions.

The solution $x^* \in X$ of Problem (1) is Pareto optimal if there is no other solution $x \in X$ such that $f_k(x) \leq f_k(x^*)\ \forall k = 1, \ldots, p$, where at least one of these inequalities is a strict inequality. If x^* is Pareto optimal solution of Problem (1), the point $y^* = f(x^*)$ is called the efficient point in the criteria space.

The set of all Pareto optimal solutions is called the Pareto set and it is denoted by X_{Par}. The set of all efficient points in the criteria space is called the efficient points set and it is denoted by Y_{eff}.

A solution obtained by minimizing the objective function $f_k(x)$ over X, is called the marginal solution for the k-th criterion of Problem (1).

One marginal solution which is Pareto optimal is called Pareto optimal marginal solution. If $\bar{x}$ is a Pareto optimal marginal solution then $f(\bar{x})$ is called efficient marginal point.

Remark 1. For each criterion $k = 1, \ldots, p$ at least one Pareto optimal marginal solution exists. It can be obtained by lexicographic optimization putting k-th criterion to have the first priority.

All notions regarding Problem (1) can be applied to the next models which are defined as its particular cases.

The model of multiobjective combinatorial optimization problem can be formulated as follows.

$$\min_{x \in \chi} f(x) \tag{2}$$

where x is the decision variable, χ is the feasible solution set of the model which is a subset of the power set of a finite set E (i.e. $\chi \subset \wp(E)$, $E = \{e_1, ..., e_m\}$), and $f = (f_1, ..., f_p)$ is the $p-$dimensional vector of objective functions.

Moreover, $x \in \chi$ can be represented by a binary m-dimensional variable $(x_1, x_2, ..., x_m) \in \{0,1\}^m$ where $x_k = 1$ if and only if the corresponding element $e_k \in E$ belongs to x and $x_k = 0$ if and only if the corresponding element $e_k \in E$ does not belong to x.

The experiments related to the number of efficient points of multiobjective optimization problems were done on three types of multiple objective network problems: multiobjective shortest path problem (SPP), multiobjective Steiner tree problem on graphs (STP) and multiobjective traveling salesman problem (TSP).

For all the three problems, an undirected graph $G = (V, E)$ with $|V| = v$ vertices, $|E| = m$ edges and $w_k : E \to \mathbb{Z}^+$, $k = 1, 2, ..., p$ weight functions on the edges is given. In addition, for SPP starting vertex $s \in V$ and target vertex $t \in V$ and for STP set of terminal vertices $T \subset V$ are given.

For each of the problems, each feasible solution represents a specific graph structure. For SPP it is a path from s to t and the feasible set χ is a set of all such paths. For STP, χ represents set of all Steiner trees of graph G and terminal nodes T. And for TSP χ is a set of all Hamiltonian cycles.

For any of the problems, a feasible solution $x \in \chi$ is a set of edges that belong to a feasible graph structure. Alternatively, a feasible solution is a vector $(x_1, x_2, \ldots, x_m) \in \{0,1\}^m$ that satisfies a set of constraints specific to each problem.

The form of goal functions in each of above mentioned problems can be twofold, depending on the type of k-th criterion:

- When a criterion represents length or weight, the corresponding goal function is linear as follows

$$f_k(x) = \sum_{j=1}^{m} c_j^k x_j \tag{3}$$

  and it is minimized.

- When a criterion type represents capacity, the corresponding goal function is

$$f_k(x) = \min_{1 \leq j \leq m} c_j^k x_j \tag{4}$$

and it is maximized.

Problems of minimizing a function of type (3) are called *minisum*, while problems of maximizing a function of type (4) are called *maximin* or *bottleneck* problems. Coefficients c_j^k, $j = 1,\ldots,m$, $k = 1,\ldots,p$ in general case are real numbers that represent length, height, weight or capacity of elements of set E. In our experiments it is presumed that coefficients c_j^k, $j = 1,\ldots,m$, $k = 1,\ldots,p$ have integer values. Practically that does not mean a loss of generality because in real life problems coefficients are rational numbers which can be transformed to integers.

In one multiobjective combinatorial optimization problem both kinds of goal function can exist. If functions of type (4) exist, they can be transformed to

$$f_k(x) = \max_{1\le j\le m}\left(-c_j^k x_j\right), \tag{5}$$

that have to be minimized, in order to match the general formulation (2).

3 Applied algorithm

In order to find all efficient points for mentioned problems an original algorithm was formulated and implemented. The algorithm is based on ε-constraints method which is usualy used in apriori approach (when the decision maker's preferences about criteria are known before optimization). Our method is developed for aposteriori approach (when optimization is done with purpose to present information about nondominated solutions to decision maker).

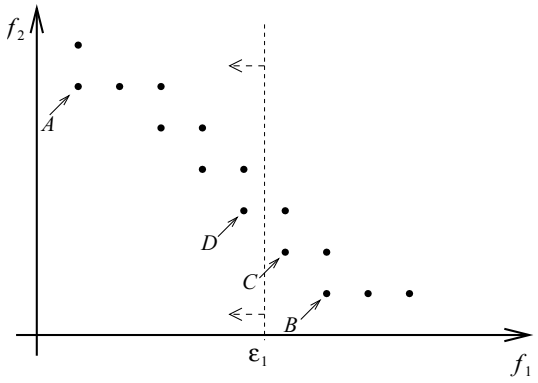


Figure 1: Sketch of applying algorithm

For finding all efficient points of a combinatorial optimization problem with two objective functions

$$\min\left\{\left(f_1(x), f_2(x)\right) | x \in X\right\}$$

the following algorithm was applied.

Algorithm 2.

Input: *Parameters of the problem.*

1. *For both criteria find efficient marginal points* $f^k = \left(f_1^k, f_2^k\right), k = 1,2.$
2. *Identify index s such as* $\left|f_s^1 - f_s^2\right| = \min\left\{\left|f_1^1 - f_1^2\right|, \left|f_2^1 - f_2^2\right|\right\}$. *Use* f_s *as a search criterion and another one, denoted by* f_o, *as an optimization criterion.*

3. *Initialize* $Y_{eff} = \left\{ \left(f_1^1, f_2^1\right), \left(f_1^2, f_2^1\right) \right\}$ *and* $\varepsilon_s = f_s^o - 1$.

4. *Solve problem* $\min \{ f_o(x) \,|\, x \in X, f_s(x) \leq \varepsilon_s \}$ *and denote by* x^* *its solution.*

5. $Y_{eff} = Y_{eff} \cup \{ (f_1(x^*), f_2(x^*)) \}$.

6. *Put* $\varepsilon_s = f_s(x^*) - 1$. *If* $\varepsilon_s > f_s^s$ *then go to Step 4, otherwise STOP.*

Output: *Set* Y_{eff}.

An example of a part of criteria space and efficient points obtained by Algorithm 2 is graphically represented in Figure 1.

Points A and B represent eficient marginal points $\left(f_1^1, f_2^1\right), \left(f_1^2, f_2^2\right)$ of the problem. They were obtained by independent optimization of criteria f_1 and f_2 respectively and they are efficient points of the multiobjective problem. In this example f_1 is the search criterion and f_2 is the optimization criterion. Points C and D are also efficient points of the problem and they were obtained in two succesive iterations. Efficient point C is obtained in the first iteration. In the next iteration point D will be obtained by optimizing second objective function over the initial feasible set reduced by the additional constraint $f_1(x) \leq \varepsilon_1$.

Remark 3. Algorithm 2 can be extended for problems with more than two criteria by putting one criterion to be optimization criterion and all others to be search criteria. Efficiency of such procedure can be questionable because all search criteria have to be recursively checked. This implies time and resource consumption problems. Also, efficient marginal points can be difficult to be determined in this case.

4 Experiments

Problem SPP, in its single objective version, belongs to the class $\mathcal{P}$ and its multiobjective version represents one of the most studied problems of MOCO. Problem STP in its single objective version with a *minisum* criterion belongs to the class $\mathcal{NP}$, but if objective function is of type *maximin*, it belongs to the class $\mathcal{P}$ [9]. The single objective TSP, with any type of objective function (*minisum* or *maximin*) is an $\mathcal{NP}$ hard problem.

4.1 Description of the developed computer programs

For each of the problems, a specific computer program was developed by authors. Each program determines all efficient points (and one solution for each of them) using Algorithm 2 adapted to the corresponding problem.

The programs for SPP and STP support instances with both kinds of objective functions *minisum* and *maximin*. Program for TSP supports only instances with *minisum* criteria type.

For all the experiments random instances with certain characteristics were generated. All instances had two non correlated criteria. Each result of experiments is obtained as an average of 10 randomly generated instances with the same characteristics.

For SPP problem instances had specific structure in order to provide paths to have at least $\sqrt{m}$ edges. Graph density varied between 36% and 60% depending on the number of vertices. Instances with smaller number of vertices had higher density.

Instances for examining STP problem were generated so each vertex has a certain probability to be connected to any of other vertices. Both, too dense or too spare graphs would not be proper for this kind of problem. Average number of a vertex degree was 7 for all dimensions, e.g. average density was 35% for graphs with 20 vertices and 14% for graphs with 50 vertices. The number of terminals was 5 for all instances.

TSP instances were generated in a similar way as STP, but density was higher, 50% for all graph dimensions.

4.2 Example

In order to present the procedure of performing experiments a short example of finding all efficient points of a small instance of STP problem is given below.

The instance has the following characteristic: unoriented graph with 10 vertices and 20 edges (consequently, graph density is 44% and average number of vertex degree is 4), 5 terminal nodes, two criteria of type *minisum*. Edges weights were generated randomly from interval $[10, 99]$. Criteria were not correlated.

The upper triangular matrix of edges weights for both criteria is given below.

$$\begin{bmatrix}
0 & 98/95 & - & - & 45/17 & - & 92/12 & - & - & 41/45 \\
 & 0 & - & 67/58 & - & 12/89 & - & - & 32/47 & - \\
 & & 0 & 65/98 & 22/61 & - & 71/67 & 19/32 & - & - \\
 & & & 0 & - & 33/49 & - & 83/32 & - & - \\
 & & & & 0 & 18/70 & 11/93 & 51/47 & 10/53 & - \\
 & & & & & 0 & - & 44/51 & - & - \\
 & & & & & & 0 & - & - & 12/29 \\
 & & & & & & & 0 & - & - \\
 & & & & & & & & 0 & 15/25 \\
 & & & & & & & & & 0
\end{bmatrix}$$

Results obtained by developed program using Algoritm 2 are presented in Table 1.

Table 1: Output results - coordinates of efficient points

No.	First criterion values	Second criterion values	Remarks
1	130	286	supported point
2	174	276	-
3	176	236	supported point
4	199	234	-
5	232	231	-
6	236	200	supported point
7	275	186	supported point

Seven efficient points were obtained and their coordinates in criteria space are given in Columns 2 and 3 of Table 1. Also, for each efficient point it is determined wether they are supported or non supported points (information given in Column 4 of Table 1).

4.3 Types of the performed experiments

Tree groups of experiments were performed.

The first group was inspired by a supposition that the upper bound for the number of efficient points depends on the length of the intervals from which edges lengths can get integer values. Three such intervals were defined: I1=[10, 99], I2=[100, 999] and I3=[1000, 9999] (as sets of two, three and four digits numbers) which contain sets of integer values of cardinality 90, 900 and 9000, respectively. Instances were generated so the lengths of edges would get random values from a certain interval, independently for each criterion. All combinations of the intervals for the first (C1) and second (C2) criterion were

checked. All the experiments from this group were performed on graphs with 20 vertices. Both criteria were of *minisum* type.

For all problems, SPP, STP and TSP, two values were observed: upper bound (UB) and actual numbers of efficient points (EFF).

The upper bound was calculated by the formula

$$UB = \min\left\{f_1^2 - f_1^1, f_2^1 - f_2^2\right\} + 1 \tag{6}$$

where $\left(f_1^k, f_2^k\right), k = 1,2$ are efficient marginal points of the k-th criterion. Both UB and EFF are calculated as the average of results obtained from 10 randomly generated instances.

In paper [6] a different upper bound was also considered. Although that upper bound has a polynomial growth with the problems size, it is more rough than UB. On the other side for its calculation were used only parameters of instances. Upper bound UB presented in this paper is far more precise. But, in order to obtain it, p^2 optimizations per instance are necessary to be performed in order to obtain efficient marginal points.

The analysis and comparisons of upper bound and the actual number of efficient points were performed in order to get an idea about order of magnitude and relations between the values of the UB and EFF. Results of this group of experiments are given in Table 2.

Table 2: Dependence of the upper bound and efficient points number on the range of edge lengths for SPP, STP and TSP

prob:		SPP		STP		TSP	
C1	C2	UB	EFF	UB	EFF	UB	EFF
I1	I1	136	9.6	133	7.5	483	40.5
I1	I2	147	6.6	172	8.9	546	44.1
I1	I3	166	7.6	159	7.9	498	39.3
I2	I2	1027	6.5	905	8.7	4951	44.0
I2	I3	1870	9.8	1804	9.3	5489	45.1
I3	I3	13967	8.5	13461	8.0	47606	39.1
average			8.1		8.4		42.0

The second group of experiments was performed in order to check the dependency of the number of efficient points on the graph size. Because of the exponential complexity of the algorithms for finding all efficient points for all tree problem types when objective functions are of *minisum* type, the experiments were performed for instances with up to 50 vertices. Performing experiments it was concluded that the number of efficient points for STP with both criteria types *minisum* and *maximin*, significantly more depends on the number of terminal vertices than on total number of vertices in graph. Because of that, instances of problems SPP and TSP with *minisum* criteria types were compared separately.

Experiments results for this group are represented in Table 3.

In order to demonstrate "independence" of efficient points number on graphs size for STP, some experimental results are given for *maximin* criteria types. For this problem, algorithm is polynomial, so instances with much bigger number of vertices were able to be solved. These results are given in Table 4.

The third group of experiments considered the types of criteria. Three combinations were observed: when both criteria were of type *minisum* (S/S), when the first criterion was of type *maximin* and second was of type *minisum* (M/S) and when both criteria were of type *maximin* (M/M).

This time the TSP problem was excluded from the experiments because the available software did not support solving TSP with *maximin* criterion. Instances with both, 20 and 50 vertices were observed.

Table 3: Dependence of the efficient points number on the graph size for SPP and TSP

problem	SPP		TSP	
v	UB	EFF	UB	EFF
10	622	3.5	771	4.7
20	1096	8.3	4809	43.9
30	2158	15.0	8365	105.5
40	2333	19.1	14116	223.5
50	2371	16.8	18521	373.9

Table 4: Depedence of the efficient points number on the graph size for *maximin* STP

intervals:	$I1$		$I2$		$I3$		average for size
v	UB	EFF	UB	EFF	UB	EFF	EFF
50	35	5,1	224	3,9	3928	7,1	5,4
100	33	5,7	235	3,6	4269	7,2	5,5
200	24	4,0	267	5,5	3233	5,0	4,8
500	26	4,3	181	3,1	2357	3,7	3,7
1000	13	2,3	153	4,1	1858	3,9	3,4
2000	14	3,6	161	4,9	1830	4,1	4,2
average for interval		4,2		4,2		5,2	4,5

The results of the third group of experiments are represented in Table 5.

Table 5: Dependence of the efficient points number on the type of criteria for SPP and STP

criteria type:		C=S/S		C=M/S		C=M/M	
problem	v	UB	EFF	UB	EFF	UB	EFF
SPP	20	2886	8.1	938	5.4	309	2.6
	50	5689	20.0	1141	12.8	739	5.4
STP	20	2772	8.4	908	7.8	667	5.5
	50	2815	8.9	818	9.6	674	5.8

5 Conclusion

We can make the following conclusions which are based on results presented in Tables 2, 3, 5.

Although the problems SPP, STP and TSP have very different nature, their number of efficient points show very similar characteristics.

The most interesting are the values in columns EFF. First of all, they are surprisingly small, and second, the influence of the observed parameters to it is very low or even insignificant.

Observing Table 2, it is obvious and expected that upper bound UB grows with the size of the intervals from which edges take their values. Very unexpected is that the actual number of efficient points does

not show any dependence on the size of the intervals for all three problems.

It is also obvious that TSP has about five times bigger number of efficient points than SPP and STP which have similar number. Explanation is that for instances we used, TSP solutions contain more edges than solutions of SPP and STP. Consequences are that the distance between the two efficient marginal points is bigger, because of that UB is also bigger and it is expected that bigger number of efficient points can be between them.

Observing the results from Table 3 a little deviation can be noticed for SPP between graphs with 40 and 50 vertices. Namely, the number of efficient points on this stage starts to decrease. However, we concluded that the deviation is accidental, especially because in the next group of experiments, on different instances with the same characteristics (SPP, 50 vertices, S/S) is obtained value 20.0 (shown in Table 5) which matches the value expected in Table 3. Here the number of efficient points is bigger for TSP and moreover, it grows faster than for STP. This is in accordance to the previous explanation because the number of edges in solution for TSP grows linearly with number of vertices and for SPP is approximately $\sqrt{m}$.

Finally, the results from Table 5 show that the number of efficient points is smaller for *maximin* than for *minisum* type of criteria, i.e. if more criteria are of *maximin* type, smaller will be the number of efficient points. A slightly deviation of that rule is in the last row where for M/S combination of criteria types is a bigger number of efficient points than for S/S. Still we consider this as an accidental deviation.

It was mentioned before and presented in Table 4 that the number of efficient points for STP does not depend much on the number of vertices and it is also obvious from the last two rows of Table 5.

In all the considerations in this paper it was assumed that criteria are not correlated. On the other hand, the number of efficient points decreases with the increase of correlation between criteria. Since it is known that between many criteria used in practice correlation exists (the length, time and price of path, price and reliability etc.), we can expect even less number of efficient points when it comes to practical problems.

Bibliography

[1] M. Ehrgott, *Multicriteria Optimization*, Springer-Verlag, 2000.

[2] M. Ehrgott, X. Gandibleux, A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization, *OR Spektrum*, Vol. 22, pp. 425-46 2000.

[3] V.A. Emelichev, V.A. Perepelitsa, On Cardinality of the Set of Alternatives in Discrete Many-criterion Problems, *Discrete Math. Appl.* Vol. 2, pp. 461-471, 1992.

[4] H.W. Hamacher, G. Ruhe, On Spanning Tree Problems with Multiple Objectives, *Ann. Oper. Res.*, Vol. 52, pp. 209-230, 1994.

[5] I.V. Sergienko, V.A. Perepelitsa, Finding the Set of Alternatives in Discrete Multicriterion Problems, *Cybernetics* Vol. 23, pp. 673-683, 1987.

[6] M. Stanojević, M. Vujošević, B. Stanojević, Number of Efficient Points in Some Multiobjective Combinatorial Optimization Problems, *International Journal of Computers, Communications & Control*, Vol.III (supl. issue), pp. 497-502, 2008.

[7] M. Visée, J. Teghem, M. Pirlot, E.L. Ulungu, Two-phases Method and Branch and Bound Procedures to Solve the Bi-objective Knapsack Problem, *J. Glob. Optim.* , Vol. 12, pp. 139-155, 1998.

[8] M. Vujošević, M. Stanojević, Multiobjective Traveling Salesman Problem and a Fuzzy Set Approach to Solving It. In: D. Ivanchev, M.D. Todorov (eds), *Applications of Mathematics in Engeneering and Economics*, Heron Press, Sofia, pp. 111-118, 2002.

[9] M. Vujošević, M. Stanojević, A Bicriterion Steiner Tree Problem on Graph", *Yugosl. J. Oper. Res.*, Vol. 13, pp. 25-33, 2003.

Milan Stanojević
University of Belgrade
Faculty of Organizational Sciences
154 Jove Ilića, 11000 Belgrade, Serbia
E-mail: milans@fon.bg.ac.yu

Mirko Vujošević
University of Belgrade
Faculty of Organizational Sciences
154 Jove Ilića, 11000 Belgrade, Serbia
E-mail: mirkov@fon.bg.ac.yu

Bogdana Stanojević
Transilvania University of Braşov
Department of Computer Science
50 Iuliu Maniu, Braşov, Romania
E-mail: bpop@unitbv.ro

Milan Stanojević was born in Belgrade, Serbia in 1965. He graduated at University of Belgrade, Faculty of Organizational Sciences in 1990. He obtained doctoral degree at the same faculty in 2005. Since 1993 he works at Faculty of Organizational Sciences, in the begining as a teaching assistant and now as an assistant professor of operational research.
He has published more than 40 papers in national and international journals, and conference procceedings in the field of operational research. His research interest includes multiobjective optimization, combinatorial optimization and software for operational research.

Mirko Vujošević was born in Podgorica, Yugoslavia in 1951. He graduated in electrical engineering at University of Belgrade where he also finished his postgraduate studies and earned his doctorate. From 1976 to 1995 he worked at Mihailo Pupin Institute, Belgrade, and now he hold the chair of Operational Research and Statistics at the Faculty of Organizational Sciences, University of Belgrade.
He has published more than 150 professional papers on different topics of operational research, reliability, maintenance, inventory control and applied mathematics. He is author and co-author of two monographs, (one of them published by Elsevier), five textbooks, and several chapters in monographies. He is member of editorial boards of several scientific and professional journals and associated editor of YUJOR - Yugoslav Journal of Operational Research. He is member of Programme Committees of several national and interanational conferences, as well as several professional societies: DOPIS - Yugoslav Operational Research Society (he was its president for eight years), PRIM - Serbian Society for Applied and Industrial Mathematics, IEEE - Institute of Electric and Electronic Engineers, ORS - Operational Research Society (U.K.), ISIR - International Society for Inventory Research and INFORMS - Institute for Operations Research and Management Science. He is member of Academy of Engineering Sciences of Serbia.

Bogdana Stanojević was born in Oradea, Romania in 1972. She graduated Mathematics and Computer Science specialization at "Transilvania" University of Braşov, in 1995 and she obtained her doctoral degree in Mathematics in 2003 from the Romanian Academy. Currently she is an associate professor at Computer Science Department of Transilvania University of Braşov. Her research interests include different aspects of Fuzzy Optimization, Multiple Objective Optimization and Mathematical Fundamentals of Computers.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 384-394

Redistributing Fragments into a Distributed Database

Leon Ţâmbulea, Manuela Horvat-Petrescu

Abstract: A distributed system database performance is strongly related to the fragment allocation in the nodes of the network. An heuristic algorithm for redistributing the fragments is proposed. The algorithm uses the statistical information relative to the requests send to a distributed database. This algorithm minimizes the size of the data transferred for solving a request. Assuming that a distribution of the fragments in the nodes of a network is known, the algorithm generates a plan to transfer data fragments, plan that will be used to evaluate a request.
Keywords: distributed database, fragment allocation, allocation algorithm, transfer cost, heuristic algorithm, redistribution algorithm

1 Introduction

Let's consider a distributed database C, formed by n nodes (sites) S_i, $0 <= i < m$. Each node contains a local database and has the capability to evaluate requests. The distributed database C contains a set of n fragments F_j, $0 <= j < n$. Each F_j fragment has a specific dimension $dim(F_j)$ - the dimension can be measured in bytes, pages, so on. Different fragments noted with L_j can be stored in a S_i node.

Let's assume that a user is sending a request q (a select, an update, a query). For solving this request, are necessary a set of fragments $r(q)$ and for each fragment an access right (read/write - depending of the request) is required.

The great majority of requests received by a distributed database are requests used for data retrieving. The fragments that should be transferred between the nodes of the network are required when evaluating a request. An optimal allocation of the fragments in the nodes of the network is necessary for the request evaluation time to be minimum. In [1, 3, 4, 7, 9, 10, 11] are mentioned other solutions for solving the allocation problem. Because of the complexity of this problem (NP-complete problem) more heuristic algorithms were proposed, algorithms with a lower complexity that provide only an approximate solution. In [8] is described a model that propose a dynamic redistribution of the fragments using the statistical information gathered in a specific period of time. Queries and information about the performed fragment transfer can be obtained for a specific period of time in a distributed database. Using this data obtained over a longer period of time, a redistribution of the fragments can be performed for minimizing the size of the data transfer.

This article is organized as follows: In section 2 is described the model for evaluating a query, and also the useful information that can be obtained when evaluating this query. The third section describes the problem of redistributing the fragments in the nodes of a distributed database. For solving this problem is proposed an heuristic algorithm that minimizes the size of the data transferred between the network nodes. In section 4 is proposed an algorithm for generating a transfer plan of the fragments when evaluating a query. This transfer plan is obtained using the query evaluation plan and the fragments distribution in the nodes of the network. In the final section of the paper is presented a conclusion section.

2 Query Evaluation

An execution plan can be determined for each request q. In centralized databases, lots of studies regarding how to find an execution plan with a minimum cost were performed, and some results were implemented in commercial database management systems.

Using the execution plan, a request q can be split in a number of sub requests $\{ q_i, i \in I_q \}$, and each sub request q_i corresponds to an operator in the relational algebra. An operator requires one fragment (if it's an unary operator) or two fragments (if is a binary operator). These arguments (fragments) can be stored in a node in the distributed database or can be the result of evaluating other operators.

In a distributed system, a request q can be evaluated as follows [5, 6]:

- Centralized, in a specific node S from the network. All the fragments $r(q)$ necessary for solving the request q should be transferred in this node S. The node S in which the evaluation is performed can be determined as the total fragments $r(q)$ transfer cost to be minimum.
- Distributed: each sub request q_i is evaluated in a separate node S_j in the network and in the S_j node should be found only the $r(q_i)$ fragments. The $r(q_i)$ fragments can be fragments stored in the S_j node, can be the result of a previous evaluation in this node, or can be transferred from other node of the network.

In order to evaluate the cost of a request q, there has to be evaluated the cost of the operators and the cost of the fragments transfer to the nodes where these operators are evaluated ([5, 6]). In this section as in the next one, we will analyze the fragment redistribution problem in the nodes of a distributed database in order to obtain a minimum cost for data fragments transfer when evaluating requests.

Let's note with $c_{i,j}$ the cost of the transferring a data unit from the node S_i to the node S_j. For a fragment F (that is stored in the node or is a fragment resulted from evaluating previous requests) transferred from node S_i to the node S_j the total cost is $c_{ij} * dim(F)$. Because the c_{ij} cost is relative but the difference between them (in absolute terms) is relatively small, we'll suppose in this paper that they are constants; $c_{i,j} = 1,\ 0 <= i, j < m$. This assumption simplifies the solution for the proposed problem.

A request q can be split into elementary queries (operations) that execute over different fragments stored in the node N. If such an elementary request (a join operator for example) uses fragments that are not stored in the same node, than all the fragments should be transferred to the same node.

In the following example, the distributed database is composed by two relations: A and B, and has horizontal fragmentation, such as:

$$A = A_1 \cup A_2 \ ; \ B = B_1 \cup B_2 \ ;$$

The request/query q for this database is:

$$q = A * \sigma_{c}(B)$$

where c is a condition and "*" represents a join operator.

If all the fragments are stored in the same node than the request q can be evaluated in the node without requiring other data transfers. If all the fragments (A_1, A_2, B_1 and B_2) are stored in different nodes (noted with S_0, S_1, S_2 and S_3) then evaluation of the request q implies a data transfer cost. We'll consider that the requests q is required in a node S different from the previous mentioned nodes (S_0, S_1, S_2 and S_3) where the data fragments are stored. Two evaluation strategies can be considered:

1. The evaluation is performed in the node S, so each required fragment is transferred from the node where is stored into the S node. The size of the whole transfer is:

$$dim(A_1) + dim(A_2) + dim(B_1) + dim(B_2).$$

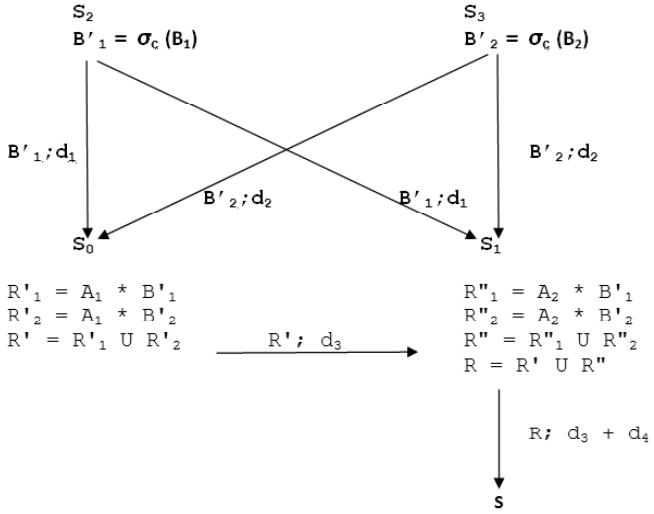
2. The evaluation is performed in a distributed manner: for this case the request q can be transformed as follows:

$$q = A * \sigma_c(B) = (A_1 \cup A_2) * \sigma_c(B_1 \cup B_2) =$$
$$= [A_1 * \sigma_c(B_1 \cup B_2)] \cup [A_2 * \sigma_c(B_1 \cup B_2] =$$
$$= [A_1 * (\sigma_c(B_1) \cup \sigma_c(B_2))] \cup [A_2 * (\sigma_c(B_1) \cup \sigma_c(B_2))]$$

The request q can be evaluated using the following graph, where we consider that the result of evaluating $\sigma_c(B_i)$ will have a size/dimension d_i smaller that $dim(B_i)$, $i=1,2$.

Let's note with $B'_i = \sigma_c(B_i)$, i=1,2 the results of the selection from the S_2 and S_3 nodes and $d_3 = dim(B'_1)$, $d_4 = dim(B'_2)$.

In the nodes of the following graph (the nodes of the graph represent the nodes of the network) are presented the operators (unary and binary) that are evaluated and on the links are shown the data transfers required in the evaluation process.



The total cost of the transfer will be $2\,(d_1 + d_2 + d_3) + d_4$.

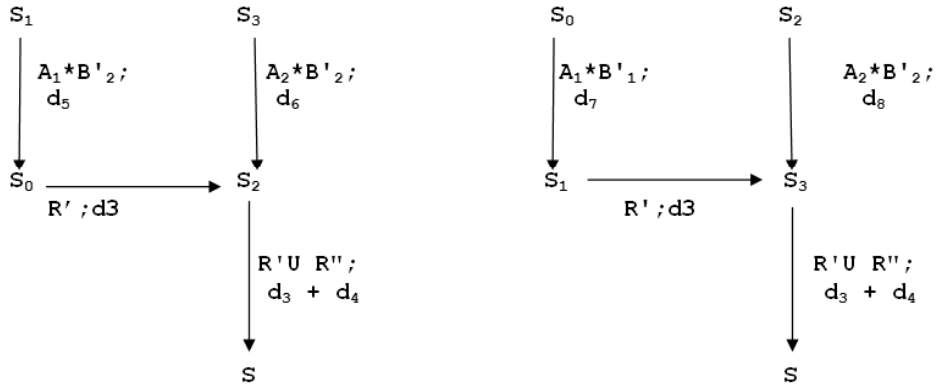
In a distributed database the fragments can be replicated - a fragment can be stored in more than one site/node. For the distributed database presented above, we'll exemplify for two fragment distribution plans the corresponding execution plan.

3.

Node	S_0	S_1	S_2	S_3
Fragments	A_1 , A_2	B_1 , B_2	A_1 , B_1	A_2 , B_2

4.

Node	S_0	S_1	S_2	S_3
Fragments	A_1 , B_1	A_1 , B_2	A_2 , B_1	A_2 , B_2



3 Redistribution of the fragments

Let's assume the following situation: in a distributed database for a specific period of time T is required to obtain the answers to a set of requests Q. The problem raised is the redistribution of the fragments as the size of the data transfers between the nodes for the same set of requests Q (if they are repeated) to be decreased. The size of the transfer will be zero if the database in completed distributed (so each fragment is stored in each node).

We'll assume that the size of each node in the network is limited by a maximum value noted with *DMax*, and the number of the replica r_j for a fragment F_j is between two limits:
$1 <= r_j <= RMax_j$, $0 <= j <n$.

The problem of optimal redistribution of the fragments is quite difficult as the optimization performed by the query optimizer must be taken into account and also must be taken into account the size of the data transfer between the nodes. Due to the complexity of this problem, more heuristic algorithms were proposed (for example [1, 3, 4, 7, 9, 10, 11]). We describe a new and much simpler algorithm which offers an approximate solution for the proposed problem.

The result of generating an execution plan for a query q is a list of operators that should be evaluated: $op_1(X)$ - for an unary operator or $op_2(X, Y)$ in case of binary operators, where X and Y are fragments stored in the database or are the result of evaluating previous operators. If is an unary operator, then the evaluation can be performed in the node where the operand is stored and no data transfer is required. In case of a binary operator $op_2(X, Y)$, the evaluation can be performed in the node where the X fragment is stored or is determined, or where the Y fragment is stored or is determined. The node is chosen in order to minimize the data transfer size.

From the information obtained when evaluating the set of requests Q, in a fragment redistribution is used only the information related to the data transfer: "the fragment X (or the fragment obtained from an evaluation in a node where the fragment X is stored) is transferred in a node where is stored the fragment Y (or the fragment obtained from an evaluation in a node where the fragment Y is stored)".

The data transfers occur only when evaluating binary operators. For example, for the evaluation strategy *b* from the previous section the following transfers occur:

Binary operator	Data Fragment	Observations
$R'_1 = A_1 * B'_1$	B_1, A_1, d_1	The meaning of the first line is: d_1 is transferred from a node where B_1 is stored to a node where A_1 is stored
$R'_2 = A_1 * B'_2$	B_2, A_1, d_1	
$R''_1 = A_2 * B'_1$	B_1, A_2, d_1	
$R''_2 = A_2 * B'_2$	B_2, A_2, d_2	
$R' = R'_1 \cup R''_2$	A_1, A_1, d_3	Does not require transfer
$R'' = R''_1 \cup R''_2$	A_2, A_2, d_4	Does not require transfer
$R = R' \cup R''$	A_1, A_2, d_3	

The final result where the fragment A_2 with size $d_3 + d_4$ is stored must be transferred to the node S.

For a set of requests Q that were evaluated in a period of time, a journal with two types of information can be created:

1. A set of tuples: $T_1 = \{ (F_i, F_j, d_i), i \in I \}$, where a tuple $t = (F_i, F_j, d_i)$ has the meaning that from a node that stores the F_i fragment, some data with size d_i is transferred in a node where the fragment F_j is stored.
2. A set of tuples: $T_2 = \{ (F_q, S_q, d_q), q \in Q\}$, where a tuple $t = (F_q, S_q, d_q)$ corresponds to a query $q \in Q$ and has the following meaning: from the node where the fragment F_q is stored (and where the query evaluation was finalized) some data with size d_q are transferred in the node S_q where the result for the query q is required.

The algorithm has two phases:

First phase: the distribution of the fragments in the virtual nodes is performed such as the total size of the data transfer to be minimum.

Second phase: the m virtual nodes created in the first phase are associated with the real nodes in such a manner as the size of the data transfer involved in answering the queries to be minimum.

A pair of fragments F_i and F_j can be found in more than one tuple from T_1 set, and can correspond to a set of binary operators required by the queries from the set Q. If these two fragments are stored in the same node, than there is no data transfer for neither one of the binary operators. In the proposed algorithm, a matrix $V = (v_{i,j})$, $0 <= i, j <n$ will be determined. The $v_{i,j}$ represents the size/dimension of the data fragments that don't have to be transferred when evaluating the queries from Q in the case when the fragments F_i and F_j are stored in the same node.

The V matrix can be easily determined using the following algorithm:

1. $v_{i,j} = 0$, $0 <= i, j <n$;
2. For each $t = (F_i, F_j, d_i) \in T_1$: $v_{i,j} = v_{i,j} + d_i$;
3. For each $0 <= i <n$, $0 <= j <i$: $w = v_{i,j} + v_{j,i}$; $v_{i,j} = w$, $v_{j,i} = w$

The V matrix is symmetrical and the values have the mentioned significance. The values: $v_{i,j}$, $0 <= j <= i <n$ represent the inferior block of the matrix V.

These values will be sorted in a descending order and will be used (in this order) to generate a new data fragments distribution in the distributed database.

The proposed algorithm is given in the next paragraph and contains four phases:

1. For each node S_i , $0 <= i < m$ the following initializations will be performed:

   (a) $s_i = 0$; *(the size/dimension used by the node* S_i*)*

   (b) $L_i = \Phi$; *(the set of the fragments stored in the node* S_i*)*

   (c) For each fragment F_j, $0 <= j < n$: $r_j = 0$; *(the number of replicas of the fragment* F_j*)*

2. The values from the V matrix are retrieved in a descending order. A tuple $t = (F_i, F_j, v_{i,j})$ corresponding to a value $v_{i,j}$ from the V matrix represents a gain obtained for the total transfer size/dimension if the fragments F_i and F_j are stored in the same node. For this tuple t will be determined the node that offers the best advantage in the case that the mentioned fragments will be added there. For finding this node, the algorithm will compute the gain c_k obtained if these fragments are attached to the node S_k, and will keep track of an indicator sit_k. The value from sit_k is referring to the state of the fragments $\{F_i, F_j\}$ relative to the set L_k , for $0 <= k < m$. (L_k is the set of the nodes existing in S_k).

   (a) *If* $F_i \in L_k$ *and* $F_j \in L_k$, *(the fragments* F_i, F_j *are already added to the node* S_k, *then:*
   $c_k = 0$; $sit_k = 1$;

   (b) *If* $F_i \in L_k$ *and* $F_j \notin L_k$, *then:*
   If $r_j < RMax_j$ *and* $S_k + dim(F_j) <= DMax$, *then*
   $sit_k = 2$, $c_k = 0$;
   For $F_p \in L_k$: $c_k = c_k + v_{jp}$;
   Note: the fragment F_j can be added to the node S_k and the gain obtained will be c_k;
   Else $sit_k = 1$, $c_k = 0$;

   (c) *If* $F_i \notin L_k$ *and* $F_j \in L_k$, *then:*
   If $r_i < RMax_i$ *and* $S_k + dim(F_i) <= DMax$, *then*
   $sit_k = 3$, $c_k = 0$;
   For $F_p \in L_k$: $c_k = c_k + v_{ip}$;
   Else $sit_k = 1$, $c_k = 0$;

   (d) *If* $F_i \notin L_k$ *and* $F_j \notin L_k$, *then:*
   If $r_i < RMax_i$ *and* $r_j < RMax_j$, *and* $S_k + dim(F_i) + dim(F_j) <= DMax$, *then*
   $sit_k = 4$, $c_k = v_{ij}$;
   For $F_p \in L_k$: $c_k = c_k + v_{ip} + v_{jp}$;
   Else $sit_k = 1$, $c_k = 0$;

   A node S_k with c_k maxim will be determined. Depending on the value sit_k the following data are modified: L_k, S_k, r_i, r_j.

3. If a fragment F_i was not used in the binary operators obtained from Q in the period of time T used to analyze and to evaluate the requests from set Q, then the result in the precedent step will be $r_i = 0$, and that means that the fragment F_i was not stored in the nodes of the network. In this case, a node from the network with enough space to store the fragment F_i without removing other fragments will be selected.

   (a) $sit = 0$;

   (b) *For each* $0 <= k < n$:
   If $S_k + dim(F_i) <= DMax$, *then*
   F_i *is added to the node* S_k, *(*S_k, L_k *will change;* $r_i = 1$*);*

sit = 1;

Endif

If *ind=0* after running the previous algorithm it means that the fragment F_i could not be added to a node then other fragment $G_k \in L_k$ will be searched for. The fragment G_k is chosen as the node S_k will have the smaller lost when removing it:

(c) *For 0 <= k <m:*

$c_k = \infty$;

For each $Fp \in L_k$*:*

$cc = \sum v_{i,j}$; $F_j \in L_k$, $j \neq p$

If $cc < c_k$ *then* $c_k = cc$, $G_k = F_p$

(d) Is determined the S_k node having a minimum value for c_k. In the S_k node the G_k fragment is replaced by the F_i fragment. The value from L_k, r_i, and r position corresponding to G_k will be updated.

4. The R_i set containing the nodes where the fragment F_i is stored is computed for each fragment F_i. (R_i is the set of the F_i replicas)

At the end of this algorithm were determined the value of L_k for each node S_k, $0 <= k <n$. The virtual nodes can be allocated to the real nodes using the information obtained from the set of requests Q. The virtual nodes can be allocated to the real nodes according to the node where evaluation is required. This allocation will be done using a second set of tuples: $T_2 = \{(F_q, S_q, d_q), q \in Q\}$. For a request q, if the fragment F_q is stored in the node S_q (S_q is the node where the answer to the request q is required) then the data with the size d_q are not transferred.

We'll assume that the virtual nodes where the fragments are allocated are S_k, $0 <= k <m$, and the real nodes from the distributed database are SR_k, $0 <= k <m$.
With $c_{i,j}$ is noted the total gain obtained when transferring the answers of the requests from Q if the virtual node S_i is stored in the real node SR_j. ($c_{i,j}$ is the size of the data from all the answers for all the requests that are transferred from S_i to SR_j.) The $c_{i,j}$ values can be computed as follows:

$c_{i,j} = 0$; $0 <= i, j <m$;

For each $t = (F_i, SR_j, d) \in T_2$,

For each $S_k \in R_i$ *(the nodes containing* F_i *replicas):*

$c_{k,j} = c_{k,j} + d$

Is build a graph with the nodes S_k, SR_k, $0 <= k <m$, and the value $c_{i,j}$ will be associated to the line from S_i to SR_j. For this graph is determined the maximum cupling with the maximum value [2]. Using this coupling, a virtual node S_i will correspond to a real node SR_j where the L_i replicas are stored. Using this redistribution the R_i set could be recomputed.

4 Fragments transfer plan when evaluating requests

A request q is split by the execution plan in as set of operators $\{op_0, \ldots op_{n-1}\}$. The result of the request q is the result of evaluating the last operator, as the other operators generate intermediate results. Evaluation plans can be created for each operator, these plans detailing all the data transfers required in the evaluation process and the size of the transfers. If more than one evaluation plan (that evaluates in the same manner) can be created for an operator, than the plan that requires less data (in terms of size)

for transfer will be used.

Coming next is an evaluation plan for a request showing the data transfers. The plan can be noted with:

$$P = (\ evaluationList;\ dim),$$

where *"evaluationList"* is a succession of actual evaluation of the operators:

$$evaluationList : (nod_i,\ op_1,\ nod_f)\ [,\ (nod_i',\ op_2,\ nod_f')]...$$

A tuple *(nodi, op_1, nod_f)* has the following meaning: the operator *"op_1"* is evaluated in the *"nod_f"* after a data transfer from *"nod_i"*. If $nod_i = nod_f$ then there is no data transfer when the operator is evaluated (the unary operators always encounter this situation). The fragments used from the nodes nod_i and nod_f are effectively stored in the nodes or are the result of previous evaluations performed in these nodes.

"dim" is the total size/dimension of the data transferred in order to evaluate the plan from the evaluationList. This size can represent:
a) the number of the transfers involved in evaluation
b) an average size/dimension of the transferred data. The average size can be found by using values $v'_{i,j}$ computed in the same time with the values $v_{i,j}$ from section 3. ($v'_{i,j}$ would be the average size of the data transferred between the node S_i and S_j when the request set Q is evaluated.)

In the following algorithm will be used for simplicity the first meaning of the *"dim"* (explained in the paragraph a) from above).

We will note with *P(op)* the set of evaluation plans for an operator *op*. If $p = (t_0,\ t_1,\ ....\ t_{n-1};\ d) \in P(op)$ where $ti = (S^1{}_i,\ op_i,\ S^2{}_i)$, then the following notations can be used:
$dim(p) = d;$
$plan(p) = t_0,\ t_1,\ ....\ t_{n-1}$ = the transfers list required by the plan p
$result(p) = S^2{}_{n-1}$ = the node where can be found the result of the plan p evaluation

An algorithm for generating the plan *P(op)* is presented in the following paragraph (the data transfers are taken into consideration).

1. If *"op"* is an unary operator, so it has the form *op(x)* then:
   (a) If x is a fragment stored in the distributed database then
   $P(op) = \{((S, op, s); o) | S \in R(X)\}$,
   where *R(X)* represents the set of the nodes where the fragment x is stored.
   (b) If x is a fragment obtain from a previous evaluation of an operator *op'* then:
   $P(op) = \{(plan(p), (result(p), op, result(p)); dim(p)) | p \in P(op)\}$
2. If *"op"* is a binary operator, so it has the form *op(x,y)* then:
   (a) If x and y are fragments stored in the distributed database then:
   $P(op) = \{\ (\ (S_1, op, S_2)\ ;\ 1)$ where $S_1 \in R(X) - R(Y)\ ,\ S_2 \in R(Y)\}$
   $\cup\ \{\ (\ (S_1, op, S_2)\ ;\ 1)$ where $S_1 \in R(Y) - R(X)\ ,\ S_2 \in R(X)\}$
   $\cup\ \{\ (\ (S_1, op, S_1)\ ;\ 0)$ where $S_1 \in R(X) \cap R(Y)\}$.

   The evaluation of the operator *"op"* can be made in a node where the fragment x is stored and fragment y is transferred, or in a node where the fragment y is stored and fragment x is

transferred, or in a node where both fragments are stored.

(b) If x is a fragment stored in the distributed database and y is a fragment obtained from a previous evaluation of a *op'* operator, then:

$P(op) = P_1(op) \cup P_2(op) \cup P_3(op)$ where:
$P_1(op) = \{(plan\ (p), (result(p), op, S); dim(p) + 1)$
where $p \in P(op')$, $S \in R(X)$ - { result(p) }}

$P_2(op) = \{(plan\ (p), (S, op, result(p)); dim(p) + 1)\}$
where $p \in P(op')$, $S \in R(X)$ - { result(p) }}

$P_3(op) = \{(plan(p), (\ result(p), op, result(p)); dim(p))$
where $p \in P(op')$, if result(p) *in* $R(X)\}$

The evaluation of the operator *"op'"* can be made in:
- a node where the fragment x is stored by transferring the result from a node where *op'* can be evaluated
- in a node where *op'* is evaluated by transferring the x fragment
- a node there *op'* can be evaluated and where the fragment x is stored

(c) If x is a fragment obtained from a previous evaluation of a *op'* operator and y is a fragment stored in the distributed database, the plan is build as in b2.

(d) If x and y are the results of evaluation the operators op_1 and op_2 then:

$P(op) = P_1(op) \cup P_2(op) \cup P_3(op)$ where:
$P_1(op) = \{(plan\ (p_1), plan\ (p_2), (result(p_1), op, result(p_2)); dim(p_1) + dim(p_2) + 1)$
where $p_1 \in P(op_1)$, $p_2 \in P(op_2)$, $result(p_1) \neq result(p_2)\}$

$P_2(op) = \cup\ \{(plan(p_1), plan(p_2), (result(p_2), op, result(p_1)); dim(p_1) + dim(p_2) + 1)$
where $p_1 \in P(op_1)$, $p_2 \in P(op_2)$, $result(p_1) \neq result(p_2)\}$

$P_3(op) = \cup\ \{(plan(p_1), plan(p_2), (result(p_1), op, result(p_1)); dim(p_1) + dim(p_2))$
where $p_1 \in P(op_1)$, $p_2 \in P(op_2)$, $result(p_1) = result(p_2)\}$.

After the plans were generated in the cases a2, b2, b3, b4 there is the chance to obtain for an operator two different plans $p1 \neq p2$ but with the same final evaluation node $final(p1) = final(p2)$. In this case is chosen the plan with smaller dim. Using this transformation an operator will have maximum one plan for a final node.

Example: Considering the following distribution of the fragments:

Node	S_1	S_2	S_3
Fragments	A	B	A, C

Assuming the request: $q = (A * \sigma_c (B)) \cup C$.
The operator's plans are:

$P(\sigma_c) = \{((S_2, \sigma_c, S_2); 0)\}$

$P(*) = \{((S_2, \sigma_c, S_2), (S_2, *, S_1); 1), ((S_2, \sigma_c, ,S_2), (S_2, *, S_3); 1),$

$((S_2, \sigma_c, S_2), (S_1, *, S_3); 1), ((S_2, \sigma_c, S_2), (S_3, *, S_2); 1)\}$

$P(\cup) = \{((S_2, \sigma_c, S_2), (S_2, *, S_1), (S_1, \cup, S_3); 2),$

$((S_2, \sigma_c, S_2), (S_2, *, S_3); (S_3, \cup, S_3); 1),$

$..... \}$

In the previous example can be noticed that exists a request evaluation plan with only one transferred fragment.

5 Conclusion

In section 3 is proposed an algorithm for redistribution of the fragments of a distributed database. The algorithm uses the request execution plan and some information that can be obtained when evaluating the relational determinant operators from the execution plan. This algorithm minimizes the size of the data transferred between the nodes of the network when the requests are evaluated.

For evaluating a request several fragments are needed, the fragments are stored in the nodes of a distributed database. An algorithm that determines an execution plan which minimizes the number of the fragments transferred between the nodes is proposed in section s4.

The results can be extended as follows:
-In the data transfer take into consideration the cost of the transfer instead of the size of the transferred data.
-In redistribution algorithm use a maximum size/dimension for each node instead a unique maximum size for all nodes.
-In the algorithm that finds the transfer plan between the nodes (algorithm used to evaluate a request) use a transfer cost instead of the number of the tranferred fragments.

Bibliography

[1] C.H. Cheng, W.K. Lee, K.F. Wong, "A Genetic Algorithm-Based Clustering Approach for Database Partitioning", *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews,* 32: 215-230, 2002.

[2] R. Diestel, "Graph Theory", *Springer-Verlag,* Heidelberg 2000, Electronic Edition.

[3] J. Graham, "Efficient Allocation in Distributed Object Oriented Databases", *Proceedings of the ISCA 16th International Conference on parallel and Distributed Computing Systems* , Reno Nevada, August 2003.

[4] Y. Huang, J. Chen, "Fragment Allocation in Distributed Database Design", *Fragment Allocation in Distributed Database Design, Journal Of Information Science And Engineering,* 17, 491-506 (2001).

[5] I. Lungu, A. G. Fodor, "Optimizing Queries in Distributed Systems," *Revista Informatica Economica nr. 1* (37), 67-72, 2006.

[6] M. T. Õzsu and P. Valduriez, "Principles of Distributed Database Systems", *2nd ed.,* Prentice-Hall International Editions, 1999.

[7] A. Sleit, W. AlMobaideen, S. Al-Areqi, A. Yahya, "A Dynamic Object Fragmentation and Replication Algorithm In Distributed Database Systems", *American Journal of Applied Sciences 4 (8),* 613-618, 2007.

[8] L. Tâmbulea, M. Horvat, "Dynamic Distribution Model in Distributed Database", *Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III (2008),* Suppl. issue: Proceedings of ICCCC 2008, pp. 512-515.

[9] T. Ulus, M. Uysal, "Heuristic Approach to Dynamic Data Allocation in Distributed Database Systems", *Pakistan Journal of Information and Technology 2 (3)* , 231-239, 2003.

[10] S. Upadhyaya, S. Lata, "Task allocation in Distributed computing VS distributed database systems: A Comparative study", *IJCSNS International Journal of Computer Science and Network Security* , VOL.8 No.3, March 2008.

[11] O. Wolfson, S. Jajodia, "An Algorithm for Dynamic Data Distribution", *Proceedings of the 2nd Workshop on the Management of Replicated Data (WMRD-II)* , Monterey, CA, Nov. 1992.

Leon Ţâmbulea
Babeş Bolyai University, Cluj-Napoca
Faculty of Mathematics and Computer Science
Department of Computer Science
M.Kogãlniceanu No.1
E-mail: leon@cs.ubbcluj.ro

Manuela Horvat
Babeş Bolyai University, Cluj-Napoca
Faculty of Mathematics and Computer Science
Department of Computer Science
M.Kogãlniceanu No.1
E-mail: manuela.petrescu@gmail.ro

Manuela Horvat-Petrescu is a PhD.C. in Computer Science Department at Faculty of Mathematics and Computer Science of Babes Bolyai university of Cluj-Napoca.She received the Batchelor Degree in Computer Science, Master Degree in Component Based Programming and is working now as a Software Engineer in Montran Corporation.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. III (2008), No. 4, pp. 395-411

McCLS: Certificateless Signature Scheme for Emergency Mobile Wireless Cyber-Physical Systems

Zhong Xu, Xue Liu, Guoqing Zhang, Wenbo He

Abstract: Mobile Ad Hoc Network is a self-configurable and self-organizing wireless network of mobile devices without fixed infrastructure support, which makes it a good candidate as underlying communication network for the Cyber-Physical Systems in emergency conditions such as earthquake, flood, and battlefields. In these scenarios, efficient communication schemes with security support are especially desired. Two cryptography approaches, the public key cryptography and the identity-based cryptography, face the costly and complex key management problem and the "key escrow" problem in the real-life deployment. Recently, the certificateless public key cryptography (CL-PKC) was introduced to address these problems in previous approaches. However, the efficiency of the schemes based on CL-PKC is not high and can be improved further.

In this paper, we present an improved certificateless signature scheme (McCLS) based on bilinear pairings. First, we theoretically compare the efficiency of McCLS with that of existing certificateless signature schemes (CLS). Second, an empirical study is conducted to compare the traditional AODV with the McCLS scheme based on AODV (McDV) in their efficiency and effectiveness against two most common attacks (i.e. *redirection attack* and *rushing attack*). Results from theoretical analysis show that the new McCLS scheme is more efficient than existing CLS solutions, and results from empirical studies show that the McDV is able to resist the two common attacks without causing substantial degradation of the network performance.

Keywords: Certificateless Signature, MANETs, Cyber-Physical Systems, Security;

1 Introduction

A salient feature of cyber-physical systems (CPS) is that it integrates computing, monitoring, and communication capabilities, and constantly interacts with the physical environment. As a result, cyber-physical system must be dependable, safe, secure and efficient [16].

Many emergency applications such as earthquake, flood and battlefields [10] proposed for CPS will be implemented on networked environments where computing devices are connected through wireless links. For many applications such as the military applications, fixed infrastructure may not be available in the environment or even be destroyed [9]. It is important to solve the connectivity problems with self-configurable and self-organizing characteristics. A possible solution for the lack of communication means is deployment of the Mobile Ad Hoc Networks (MANETs).

While MANETs provide a great flexibility for establishing communications, they are particularly prone to the security threats of eavesdropping, interception and routing attacks. Some of these problems may be solved or mitigated with the use of cryptographic schemes [7]. In the recent literature many papers make specific proposals on determining how to use Public Key Infrastructure (PKI) [27, 23, 4, 15] and Identity-Based Public Key Cryptography (ID-PKC) [20, 13, 8, 25] cryptographic techniques to secure MANETs.

The traditional PKI signature scheme uses a centralized certificate authority to issue a digital certificate that binds a user with the corresponding public key. The requirement of certificate authority inevitably leads to complex certificate management problems in practice.

The ID-PKC which was introduced by Shamir [20] is developed from traditional PKI to simplify the certificate management process. In the ID-PKC based scheme, user's public key is derived directly from certain aspects of his identity such as email address which is assumed to be publicly known. A private key is generated by a trusted third party – Private Key Generator (PKG). However, a new inherent problem is brought by this approach, namely the "key escrow" problem since the private key of user is known to the PKG. As a result, the PKG is able to impersonate any user of its choice, or decrypt messages.

In order to solve the costly and complex key management problems in PKI and the "key escrow" problem in ID-PKC respectively, Al-Riyami and Paterson [1] proposed the first Certificateless Public key Cryptography (CL-PKC) scheme. In the certificateless signature (CLS) scheme, Key Generation Center (KGC) only provides user with a partial private key, which is related to the user's identity and the master private key only known by PKC. Then the user generates the remaining part of the private key and the corresponding public key. As a result, the KGC does not know the user's private key because the user's private key is generated by user itself, thereby solving the "key escrow" problem in ID-PKC based schemes.

However, CLS schemes are usually computationally intensive, and hence they are not readily applicable in practical applications. In this paper, we present McCLS scheme, a new CLS scheme for mobile wireless cyber-physical systems.

Compared with existing CLS schemes, McCLS scheme only requires one pairing operation in the verification phase, and none in the signing phase. Since the pairing operation is the most time-consuming computation in pairing-based cryptosystems, our McCLS scheme has less computation overhead and therefore is more efficient than those schemes proposed previously in [1, 12, 14, 26]. We also provide a detailed security proof for McCLS scheme based on the Computational Diffie-Hellman Problem (*CDHP*) [6]. Then an empirical study is conducted to compare the McCLS based on AODV (McDV) with the traditional AODV in their efficiency and effectiveness against the two most common attacks, *redirection attack* and *rushing attack*, based on QualNet simulation software [19]. Results show that our scheme is efficient in terms of computation overhead and it can resist *redirection attack* [18] and *rushing attack* [11].

The remainder of the paper is organized as follows. Section 2 provides a brief description on the related work. Section 3 introduces the preliminaries and the background on the security model and the attack model. Section 4 presents our efficient McCLS scheme. Section 5 analyzes the security of McCLS scheme in detail. Section 6 evaluates the performance of McCLS scheme under the redirection attack and the rushing attack. Finally, Section 7 concludes this paper with summaries and the directions of future work.

2 Related Work

Cyber-Physical Systems (CPS) are physical and engineered systems whose operations are integrated, monitored, and controlled by a computational core [17]. CPS integrate the communication and computation with the physical process [2]. Since cyber-physical systems constantly interact with the physical environment, they must be dependable, safe, secure and efficient [16].

CPS is a new active research area. The position papers published in the NSF workshop on Cyber-Physical System [16] presents a good overview of the different aspects of CPS research. Though security is an important research issue of CPS , little work has been done [3] so far for the security of CPS.

Since many emergency applications proposed for CPS will be implemented on mobile ad hoc networks (MANETs), it is natural to ask the question if security schemes proposed for MANETs are practical for CPS. To overcome the security problems in MANETs due to their infrastructure-less nature, we need some new methods to solve these problems. One of these methods is the lightweight and efficient key management scheme. Recently, in order to solve the key management problem in public key

cryptography and “key escrow" problem in identity-based cryptography schemes, Al-Riyami and Paterson [1] proposed the first certificateless signature (CLS) scheme but fail to provide the security proof. Later, Huang et al. [12] found that this CLS scheme was insecure against a Type I forger attack. A modified CLS scheme was proposed with security proved under the random oracle model [5]. However, the scheme requires more pairing operations than the original scheme proposed in [1]. In [14], Li et al. therefore, proposed another CLS scheme, with a formal security analysis omitted. Another shortcoming of this scheme is that the verification algorithm requires four quite expensive pairing operations. Zhang et al. [26] presented a CLS scheme with a formal security analysis but it still needs four pairing operations in the verification phase. Following that, Yap et al. [22] proposed a new CLS scheme, which requires no pairing operation in the signing phase but requires two pairing operations in the verification phase.

However, since pairing operations are costly in computation and are usually time consuming, using more pairing operations in the scheme will make it difficult to be applied for emergency cyber-physical systems, because CPS need constantly interact with the physical environment and with stringent timing requirements. In this paper, we present McCLS scheme, which is more efficient and hence is a good alternative to be used in cyber-physical systems.

A good security protocol must be resilient against security attacks. In the following, we briefly introduce two most commonly studied attacks. Later in this paper, we prove that the proposed McCLS scheme is resilient against these two attacks.

Redirection attack [18] is one of the many possible attacks in MANETs. In this attack, a malicious node sends a forged Route Reply (RREP) packet to a source node by altering control message fields with falsified values. When a source node receives multiple RREPs, by comparing the destination sequence numbers contained in RREP packets, it regards the largest one as the most recent routing information and selects the route through which that RREP packet has been sent. If the attacker sends the RREP with destination sequence number higher than that of the real destination node to the source node, the data traffic will be directed toward the attacker. It then drops all data packets it receives instead of forwarding them to the next node on the routing path. Consequently, the source and destination nodes will lose communication with each other.

Rushing attack [11] usually aims at a reactive routing protocol. Every node in the network only forwards the first route discovery packet that it receives and drops the rest. Malicious nodes can “rush" the route request packets towards the destination. As a result, the nodes that receive these “rushed" request packets forward them and discard other route requests that arrive later. The resulted routes would then include the malicious nodes. In this way, the attacker is placed in an advantageous position.

3 Preliminaries

In this section, we present some mathematical background which helps in realizing CLS based on the bilinear pairing. It is commonly used in CLS schemes to realize signature and verification [1, 12, 14].

We define two cyclic groups G_1, G_2, where G_1 is an additive group and G_2 is a multiplicative group, where both groups have a prime order p. Let e be a computable bilinear map $e : G_1 \times G_1 \rightarrow G_2$. We have the following conditions:

1. Bilinearity: For any $P, Q, R \in G_1$, we have $e(P+Q,R) = e(P,R)e(Q,R)$. For $a, b \in \mathbb{Z}_p^*$ and $P, Q \in G_1$, we have $e(aP, bQ) = e(P,Q)^{ab} = e(P, abP) = e(abP, P)$.

2. Non-degeneracy: There exists $P, Q \in G_1$, such that $e(P,Q) \neq 1$.

3. Computability: There is an efficient algorithm to compute $e(P,Q)$ for all $P, Q \in G_1$

The map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field.

An efficiently computable bilinear map e provides an algorithm for solving the Decision Diffie-Hellman Problem *(DDHP)* [6]. That is, given $(P, aP, bP, cP) \in G_1$ and $a, b \in \mathbb{Z}_p^*$, decide whether $c \equiv ab \in \mathbb{Z}_p^*$.

In bilinear pairing, Decision Diffie-Hellman (*DDH*) problem is easy and Computational Diffie-Hellman (*CDH*) problem [22] is still hard. That is, for $a, b \in \mathbb{Z}_p^*$, given (P, aP, bP), computing abP is infeasible.

3.1 Certificateless scheme

Usually, a certificateless signature (CLS) scheme consists of five polynomial time algorithms [1]:

- **Setup.** KGC runs a probabilistic algorithm to initialize the system. It receives a security parameter k and returns a randomly chosen master key and a list of public parameters **param**.
- **Extract Partial Private Key.** KGC takes the master key and an identity $ID \in \{0,1\}^*$ as inputs, and outputs a private partial key D_{ID}.
- **Generate Key Pair.** The user takes a list of public parameters **param** as inputs, outputs a private key S_{ID} and a public key P_{ID}.
- **CL-Sign.** The user takes a list of public parameters **param**, full private keys (D_{ID}, S_{ID}), and a message M to produce a signature σ on M.
- **CL-Verify.** Anyone in this algorithm may take $\{\textbf{param}, ID, P_{ID}\}$ and a message M as inputs, and outputs *true* if and only if σ is the valid signature, or a symbol $\perp$ to indicate a failure.

We may note that, once the user received the public parameters, such as public key of KGC, user chooses secret value to generate his key pair including user's private key and user's public key. Thus the user's full private key is composed of the partial private key generated by KGC and the user's private key generated by user himself. Neither the KGC nor the user can generate the full private keys by himself, therefore solving the "key escrow" problem.

3.2 Adversarial Model

As defined in [1, 22], there are two types of adversaries, Type I and Type II, with different capabilities. In CLS, Type I Adversary A_I acts as a third part who tries to impersonate a user. It is not allowed to know the KGC's master private key. However, A_I can replace the public key P_{ID} with values of its choice due to nature of the public key generated by the user. This means the adversary is able to fool the user accepting the signature, which is signed by the adversary's public key. Type II adversary A_{II} represents a malicious KGC who knows the master private key. That is, A_{II} can compute the partial private key by itself. But A_{II} does not know the user's private key S_{ID} and it cannot replace the user's public keys P_{ID} .

Definition 1. A CLS scheme is secure against existential forgery on adaptive chosen message and ID attacks against adversary A, of Type I or Type II if no polynomial time algorithm has a non-negligible advantage against a challenger C in the following game [1]:

1. The challenger C takes a security parameter k and runs the *Setup* algorithm. Challenger C gives A the system parameters **param**. If A is of Type I, the challenge C keeps the master private key to itself. Otherwise, C gives the master private key to A.
2. A can request C to answer the following types of queries:

- **Partial Key Extraction** (For Type I adversary only). C returns to A's partial private key D_{ID} as the result of running **Extract Partial Private Key** algorithm.
- **Secret Value and Public Key Extraction**. C returns to A's private key S_{ID} associated with A's public key P_{ID} as the result of running **Extract Partial Private Key** and **Generate Key Pair** algorithms. In the case of Type I adversary, C returns if the user's public key P_{ID} has been replaced.
- **Public Key Replacement** (For Type I adversary only). A can replace the associated public key P_{ID} to a new public key P'_{ID} which is chosen by itself.
- **Sign**. C returns a valid signature σ using **CL-sign** algorithm regardless whether the public key P_{ID} has been replaced or not.

3. Eventually, A outputs a signature (ID^*, m^*, σ^*). A wins game if $Verify(param, P_{ID^*}, m^*, \sigma^*) = true$ and the generated output fulfills the following conditions:

   - CL-Sign(ID^*, m^*) has never been queried.
   - If adversary A is Type I, ID^* has not been submitted to **Partial Key Extraction**.
   - If adversary A is Type II, ID^* has not been submitted to **Secret Value and Public Key Extraction**.

4 McCLS Scheme

McCLS scheme is motivated by the identity-based signature from [24]. Our verification phase algorithm requires one pairing operation only, hence McCLS scheme outperforms the other existing CLS schemes in terms of efficiency. Besides, message signing in McCLS scheme is fast as it involves no pairing computation. McCLS scheme is comprised of the following five stages.

- **Setup.** Given a cyclic group G_1 of prime order p, with an admissible pairing e and its generator P, KGC picks $s \in \mathbb{Z}_p^*$ and sets $P_{pub} = sP$. Then Chooses two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_p^*$. The public system parameter list is (P, P_{pub}, H_1, H_2), and the master private key is $msk = s$.
- **Extract Partial Private Key.** Given an identity ID, KGC computes $Q_{ID} = H_1(ID)$ and $D_{ID} = sH_1(ID)$. Output D_{ID} as the partial private key corresponding to $Q_{ID} = H_1(ID)$.
- **Generate Key Pair.** The user generates a secret value $x \in \mathbb{Z}_p^*$, the public key is $P_{ID} = xP_{pub}$. The user's private key is $S_{ID} = x$.
- **Sign.** Given the user's full private keys (D_{ID}, S_{ID}) and a message M, user picks a number $r \in \mathbb{Z}_p^*$ and outputs a signature $\sigma = (V, S, R)$ where $S = \frac{1}{S_{ID}} D_{ID}$, $R = (r - S_{ID})P$ and $V = H_2(M, R, P_{ID})rP$.
- **Verification.** Given the signature (V, S, R) of a message M for the identity ID, anyone in this algorithm can act a verifier to compute $h = H_2(M, R, P_{ID})$. Then checks whether $(P_{pub}, V - hR, S/h, Q_{ID})$ is a valid Diffie-Hellman tuple, that is, computes whether $e(P_{pub}, Q_{ID}) = e(V - hR, S/h)$. If yes, accept the signature. Otherwise, reject it.

5 Analysis of McCLS Scheme

In this section, we analyze the correctness, performance and security proof of McCLS scheme.

5.1 Correctness

The correctness of McCLS scheme can be verified as follows:

$$\begin{aligned} & e(V-hR, S/h) \\ = \; & e(hrP-hrP+xhP, S/h) \\ = \; & e(xhP, D_{ID}/xh) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Note that $e(P_{pub}, Q_{ID})$ is independent of the message, and only needs to be computed once and for all. So McCLS scheme is more efficient than other previous schemes.

5.2 Performance

McCLS scheme only requires two scalar multiplication in signature phase and two scalar multiplication computations and one pairing operation in verification phase. The pairing operations are expensive comparing with scalar multiplication and exponentiation.

The comparison between the exiting schemes and McCLS scheme according to efficiency of sign and verification algorithms and the length of public keys is shown in Table 1. It shows that McCLS scheme has the lowest pairing operations requirement and has the same length of public key as other CLS schemes.

Table 1: Comparison of the CLS Schemes

	AP [1]	LCS [14]	ZWXF [26]	YHG [22]	McCLS
Sign	1p+3s	2s	3s	2s	2s
Verify	4p+1e	4p+2s	4p	2p+3s	1p+3s
Pklen	2 points	2 points	1 points	1 point	1 point

Pklen: the public key length;
s: the scalar multiplication computation;
p: the pairing operation;
e: the exponential computation.

5.3 Security Proof

In this section we discuss the security of McCLS scheme under the security model discussed in section 3. The main theorems concerning the security of our scheme are:

Theorem 2. *Our certificateless signature scheme is existentially unforgeable against a Type I adversary A_I in the random oracle model under the assumption that the CDH problem in G_1 is infeasible.*

Proof. Suppose there exists an adversary A_I which has an advantage in attacking McCLS scheme. We build a challenger C that uses A_I to solve the *CDH* problem. C receives an instance (P, aP, bP) of the *CDHP*. Its goal is to compute abP. On the setup phase, C sets P as the generator of the group, and sets $P_{pub} = aP$ where a is the master key, which is unknown to A_I. In order to avoid collision, C maintains a list $L = (ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$ throughout the game. The list is initially empty. C then starts to answer oracle queries with the following procedures [26]:

- H_1 **Queries**. Suppose A_I makes q_{H_1} queries to H_1 oracle, where q_{H_1} denotes the maximum number of queries. Randomly choose $j \in [1, q_{H_1}]$. When an identity ID_i is submitted to oracle H_1 where $i \in [1, q_{H_1}]$, if $i = j$, assume that $ID_i = ID^*$ at this point, C saves a list $L_1 = (ID_i, Q_i, y_i)$ where $Q_i = bP$, $y_i = \perp$ (indicate to failure). Otherwise, C generates a random number y_i and lets $Q_i = y_i P$, then saves $L_1 = (ID_i, Q_i, y_i)$.

- **Partial Key Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $D_{ID_i} = \perp$. If $D_{ID_i} \neq \perp$, C returns D_{ID_i} to A_I. If $D_{ID_i} = \perp$, and $ID_i \neq ID^*$, C answers with $D_{ID_i} = y_i P_{pub} = y_i(aP)$ as partial private key. C then returns D_{ID_i} to A_I and adds it to L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $(D_{ID_i} = y_i P_{pub} = y_i(aP))$. Then challenger C sets $(s_{ID}, P_{ID}) = \perp$ and adds $ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i}$ to the list L.

- **Public Key Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} \neq \perp$, C returns P_{ID_i} to A_I. Otherwise, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = x_i P_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_I and adds (s_{ID_i}, P_{ID_i}) to L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = x_i P_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_I and adds (s_{ID_i}, P_{ID_i}) to L.

- **Secret Value Extraction** (ID_i) **Queries**. When A_I makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C checks whether $D_{ID_i} = \perp$. If $D_{ID_i} = \perp$, C executes **Partial Key Extraction Queries** to obtain D_{ID_i}. If $P_{ID_i} = \perp$, C makes **Public Key Extraction Queries** to obtain $s_{ID_i} = x_i, P_{ID_i} = x_i P_{pub}$. Then C saves the value and adds full private keys (D_{ID_i}, s_{ID_i}) to the list L.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C executes **Partial Key Extraction Queries** to obtain $D_{ID_i}$and makes **Public Key Extraction Queries** to obtain (s_{ID_i}, P_{ID_i}). Then C saves the value and adds full private keys (D_{ID_i}, s_{ID_i}) to the list L.

- **Public Key Replacement**(ID_i, P'_{ID}) **Queries**. When A_I makes the query on (ID_i, P'_{ID_i}), C finds L and performs as follows:
  - If the list L contains $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $P_{ID_i} = P_{ID'_i}$ and $s_{ID_i} = \perp$.
  - If the list L does not contain $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$, C sets $D_{ID_i} = \perp$, $P_{ID_i} = P_{ID'_i}$ and $s_{ID_i} = \perp$. And then C adds to the list L.

- H_2 **Queries**. When A_I makes the query on (m, R, P_{ID_i}), C first scans if a list $L_2 = (m, R, P_{ID_i}, h_j)$ has been defined. If defined, return the list to A_I. Otherwise, C picks a random $h_j \in \mathbb{Z}_p^*$ as the hash value and returns h_j, and adds it to L_2.

- **Sign Queries**(ID_i, M_j). When A_I asks for a signature by user ID_i on message M_j. C finds $(ID_i, D_{ID_i}, s_{ID_i}, P_{ID_i})$. If D_{ID_i} not found, C runs **Partial Key Extraction Queries**. If (P_{ID_i}, s_{ID_i}) not found, C runs **Public Key Extraction Queries**. Note that if $ID_i \neq ID^*$, A_I is able to generate signature on any messages using corresponding full private keys (D_{ID_i}, s_{ID_i}). As far as $ID_i = ID^*$, assume that P_{ID_i} is current public key and corresponding private key $s_{ID_i} = x$, where $x \in \mathbb{Z}_p^*$, additionally submits through the A_I. This is because the public key has been replaced earlier by A_I, then C cannot know the corresponding private key and thus the signing oracle's answer may not be correct.

  On receiving sign queries, C does the following:

1. Choose random $r_j \in \mathbb{Z}_p^*$ and look up the list L_2 for h_j, if not found, C runs H_2 **Queries** to get h_j.
2. Compute $V_j = h_j(x + \frac{a}{r_j})P$ and $S_j = r_j Q_i = r_j bP, R_j = xP$;
3. Return the signature $\sigma = (V_j, S_j, R_j)$.

Now, σ is returned to A_I, which appears to be valid signature since

$$\begin{aligned} & e(V_j - h_j R, S_j/h_j) \\ = \; & e(h_j(x+\tfrac{a}{r_j})P) - h_j xP, r_j bP/h_j) \\ = \; & e(h_j aP/r_j, r_j bP/h_j) \\ = \; & e(aP, bP) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Finally, A_I will output a valid forgery $r = (ID_j, M_j, R_j, S_j, V_j)$. If $ID_j \neq ID^*$, C outputs the FAIL and aborts the simulation. Otherwise, we can compute r_j through $r_j = \frac{ah_j}{V_j - h_j x}$ [21], since $(P_{pub}, V_j P - h_j R_j, S_j/h_j, Q_i)$ is a valid Diffie-Hellman tuple. Apply r_j to S_j, we have

$$S_j = \frac{ah_j}{V_j - h_j x} Q_i$$

$$S_j = \frac{ah_j}{V_j - h_j x} bP$$

$$abP = S_j(V_j - h_j x)/h_j \quad .$$

So $abP = S_j(V_j - h_j x)/h_j$ is the answer to our *CDHP* instance. If the A_I can break our scheme, then the attacker solves the *CDH* problem.

Theorem 3. *Our certificateless signature scheme is existentially unforgeable against the A_{II} adversary in the random oracle model under the assumption that the CDH problem in G_1 is infeasible.*

Proof. Suppose there exists an adversary A_{II} which has advantage in attacking McCLS scheme. We build a challenger C that uses A_{II} to solve the *CDH* problem. C receives an instance (P, aP, bP) of the *CDHP*. Its goal is to compute abP. On the setup phase, C sets P as the generator of the group, and sets $P_{pub} = sP$ where s is the master key, which is known to A_{II}. In order to avoid collision, C maintains a list $L = (ID_i, s_{ID_i}, P_{ID_i})$ throughout the game. The list is initially empty. C then starts to answer oracle queries with the following procedures:

- H_1 **Queries**. Suppose A_{II} makes q_{H_1} queries to H_1 oracle, where q_{H_1} denotes the maximum number of queries. Randomly choose $j \in [1, q_{H_1}]$. When an identity ID_i is submitted to oracle H_1 where $i \in [1, q_{H_1}]$, if $i = j$, assume that $ID_i = ID^*$ at this point, C saves a list $L_1 = (ID_i, Q_i, y_i)$ where $Q_i = aP$, $y_i = \perp$ (indicate to failure). Otherwise, C generates a random number y_i and lets $Q_i = y_i P$, and saves $L_1 = (ID_i, Q_i, y_i)$.
- **Public Key Extraction** (ID_i) **Queries**. When A_{II} makes the query on ID_i, C finds L and performs as follows:
  - If the list L contains $(ID_i, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} \neq \perp$, C returns P_{ID_i} to A_{II}. Otherwise, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = bP_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_{II} and adds (s_{ID_i}, P_{ID_i}) to L.
  - If the list L does not contain $(ID_i, s_{ID_i}, P_{ID_i})$, C picks a random $x_i \in \mathbb{Z}_p^*$, and sets $P_{ID_i} = bP_{pub}$, $s_{ID_i} = x_i$. C then returns P_{ID_i} to A_{II} and adds (s_{ID_i}, P_{ID_i}) to L.

- **Secret Value Extraction** (ID_i) **Queries**. When A_{II} makes the query on ID_i, if $ID_i = ID^*$, then C aborts and halts the simulation. Otherwise C finds L and performs as follows:

  - If the list L contains $(ID_i, s_{ID_i}, P_{ID_i})$, C checks whether $P_{ID_i} = \perp$. If $P_{ID_i} = \perp$, C makes **Public Key Extraction Queries** to obtain $(s_{ID_i} = x_i, P_{ID_i} = x_i P_{pub})$. Then C saves the value and adds user's private keys s_{ID_i} to the list L.
  - If the list L does not contain $(ID_i, s_{ID_i}, P_{ID_i})$, C executes **Public Key Extraction Queries** to obtain (s_{ID_i}, P_{ID_i}). Then C saves the value and adds user's private keys s_{ID_i} to the list L.

- H_2 **Queries**. When A_{II} makes the query on (m, R, P_{ID_i}), C first scans whether a list $L_2 = (m, R, P_{ID_i}, h_j)$ has been defined. If defined, return the list to A_{II}. otherwise, C picks a random $h_j \in \mathbb{Z}_p^*$ as the hash value of Hand returns h_j, and adds it to L_2.

- **Sign Queries**(ID_i, M_j). When A_{II} asks for a signature by user ID_i on message M_j. C finds $(ID_i, s_{ID_i}, P_{ID_i})$. If (P_{ID_i}, s_{ID_i}) not found, C runs **Public Key Extraction Queries**.

  On receiving sign queries, C does the following:

  1. Choose random $r_j \in \mathbb{Z}_p^*$ and look up the list L_2 for h_j, if not found, C runs H_2 **Queries** to get h_j.
  2. Compute $V_j = (\frac{sh_j + bh_j}{r_j x_i})P$ and $S_j = r_j x_i Q_i = r_j x_i aP, R_j = \frac{bP}{r_j x_i}$;
  3. Return the signature $\sigma = (V_j, S_j, R_j)$.

  Now, σ is returned to A_{II}, which appears to be valid signature since

$$\begin{aligned} & e(V_j - h_j R, S_j / h_j) \\ = \; & e((\tfrac{sh_j + bh_j}{r_j x_i} P) - h_j \tfrac{bP}{r_j x_i}, r_j x_i aP / h_j) \\ = \; & e(\tfrac{sPh_j}{r_j x_i}, r_j x_i aP / h_j) \\ = \; & e(sP, aP) \\ = \; & e(P_{pub}, Q_{ID}). \end{aligned}$$

Finally, A_{II} will output a valid forgery $r = (ID_j, M_j, R_j, S_j, V_j)$. If $ID_j \neq ID^*$, C outputs the FAIL and aborts the simulation. Otherwise, we can compute r through $r_j = \frac{sh_j + bh_j}{xV_j}$, since $(P_{pub}, V_j P - h_j R_j, S_j / h_j, Q_i)$ is a valid Diffie-Hellman tuple. Apply r_j to S_j, we have

$$S_j = \frac{sh_j + bh_j}{x_i V_j} x_i Q_i$$

$$S_j = \frac{Q_i sh_j + bh_j aP}{V}$$

$$abP = \frac{V_j r_j - Q_i S_j h_j}{h_j} \quad .$$

So $abP = \frac{V_j r_j - Q_i S_j h_j}{h_j}$ is the answer to our *CDHP* instance. If the A_{II} can break our scheme, then the attacker solves the *CDH* problem.

Table 2: General parameters

Parameter	Value
Transmitter	250m
Bandwidth	2Mb/s
Simulation time	600s
Environment	900m×900m
Traffic type	CBR (Constant Bit Rate)
Packet rate	4 packets/s
Packet size	512 bytes
Node maximum speed	0, 5,10,15,20 m/s
Pause time	0s
Attack nodes	1,2 and 4 Redirection, 1,2 and 4 Rushing
Queuing policy at routers	First-in-first-out

6 Evaluation and Analysis

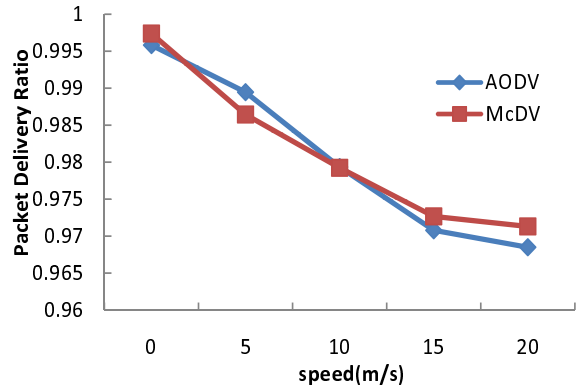
In this section, an efficient McCLS scheme named McDV based on the Ad hoc On-Demand Distance Vector Routing (AODV) is proposed. We start the simulations using QualNet [19] in order to compare the original AODV protocol without any security requirements with McDV based on the CLS with routing authentication extension. We also evaluate the performance of two schemes under 1, 2 and 4 nodes *redirection attacks* and 1, 2 and 4 nodes *rushing attacks*, as this is more realistic in the real emergency applications. Our implementation retains most of the AODV mechanisms, such as route discovery, reverse path setup, forwarding path setup, route maintenance, and so on. In our experiments, 20 nodes move around in a rectangular area of 900×900m according to a mobility model, i.e., the random waypoint model. The nodes spread randomly over the network. Each node starts its journey from a random location to a random destination. We vary the nodes speed from 0m/s to 20m/s, and set the nodes pause time as 0s. Table 2 lists the values of the common parameters used in all experiment. Other parameters will be given in the description of each specific experiment.

The performance of McDV is compared using the following performance metrics.

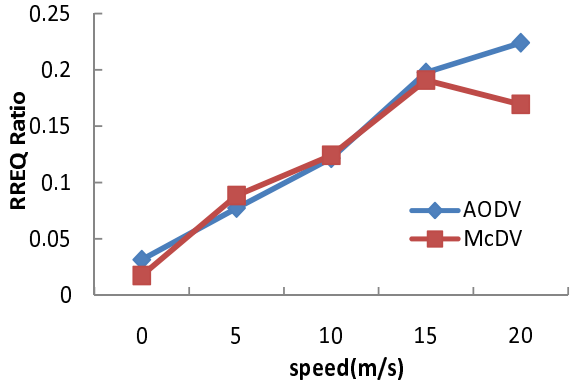
- Packet Delivery Ratio: Ratio of the number of packets received by the destination over the number of packets sent by the source.
- RREQ Ratio: Ratio of sum number of RREQ initiated, forwarded and retried over the sum of number of data packets sent as source and data packets forwarded. Present the number of RREQ packets transmitted through the network.
- End-to-End Delay: The average time experienced by each packet when traveling from the source to the destination.
- Throughput: Ratio of the total bytes sent by all sources nodes over the total time.
- Packet Drop Ratio: Ratio of the number of packets discarded by attacking nodes over the total number of packets sent by all sources.

Effects of various metrics on different protocols: Experiments in this section are used to study the performance between McDV and AODV. The results are shown in Fig. 1.

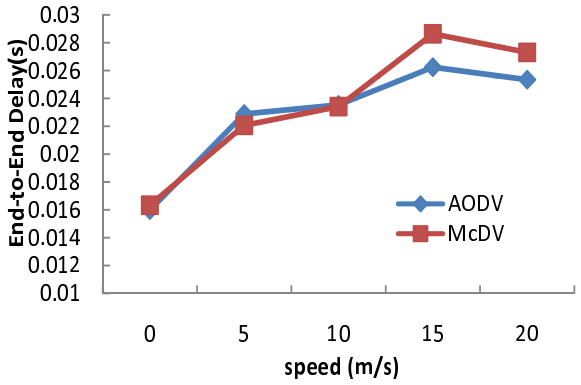
The packet delivery ratio and the RREQ ratio are shown in Fig. 1(a) and Fig. 1(b), respectively. We can see that McDV could work well in the experiment because the packet delivery ratio and RREQ ratio in AODV are very similar to that of McDV, without causing any substantial degradation of the network



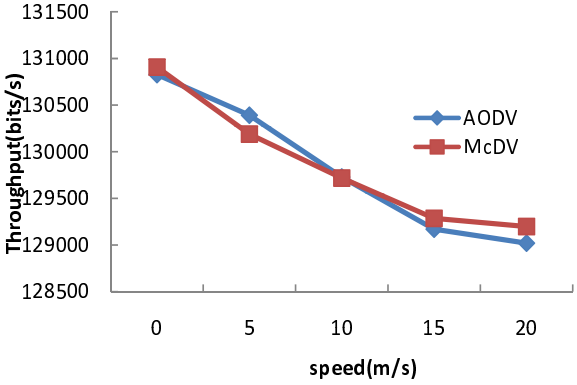
(a) Comparison with AODV and McDV, Packet Delivery Ratio



(b) Comparison with AODV and McDV, RREQ Ratio



(c) Comparison with AODV and McDV, End-to-End Delay(s)



(d) Comparison with AODV and McDV, Throughput(bit/s)

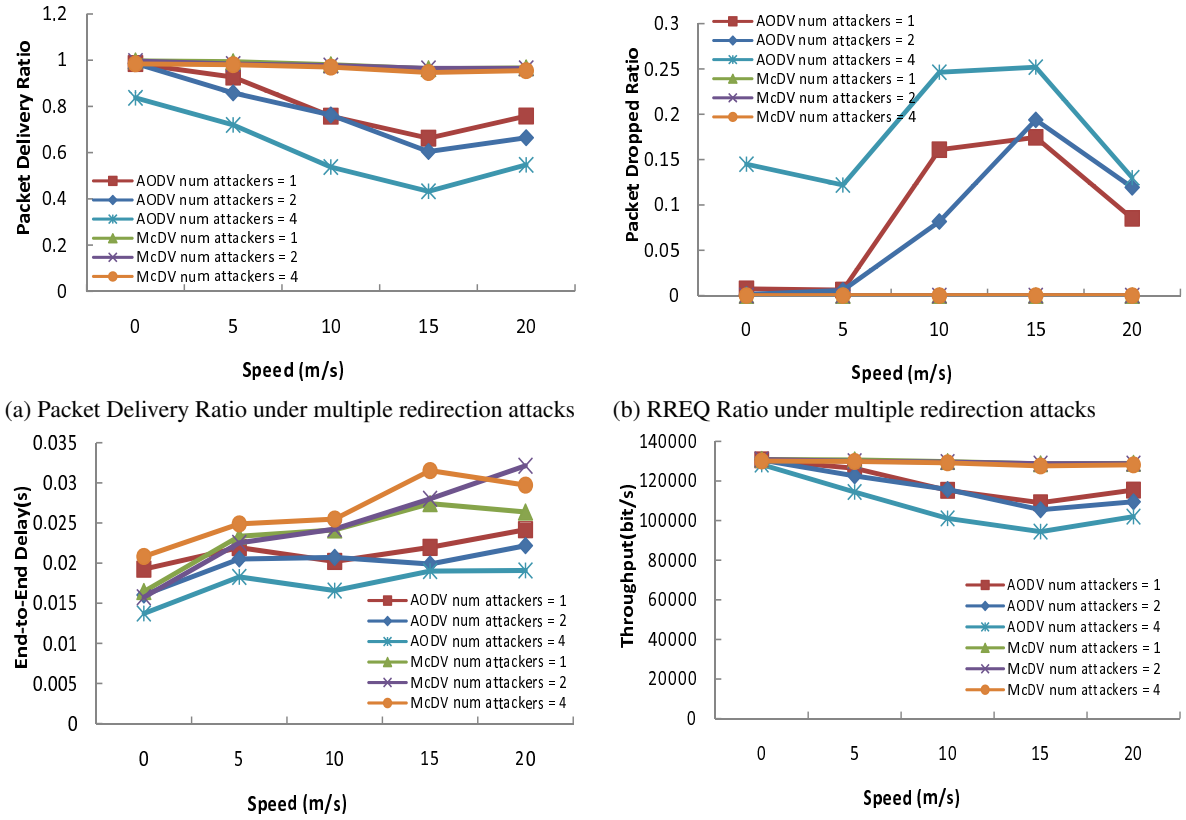
Figure 1: Effects of various metrics on different protocols

performance. As nodes speed increases, the number of data packets reaching the destination decreases and the number of RREQ packets transmitting through the networks increases.

End-to-end delay of McDV scheme is shown in Fig. 1(c). Our scheme has a little bit higher delay than that of AODV due to the exchange of packets during authentication phase of the security process. Result shows that McDV has a similar end-to-end delay with AODV at a relatively low speed, however, when the maximum speed of nodes is higher than 15m/s, AODV outperforms McDV scheme. More specifically, our scheme needs authentication operation, and those additional operations are computed in our scheme but not in AODV. We only measure delays for data packets that survived to reach their destination.

Throughput of McDV works well as result shown in Fig. 1(d) because the effect of throughput of the network is very small (around 0.16%). However, if this scheme in other real scenarios such as disaster scenarios, battlefield scenarios, or even very high-speed scenarios, the effect of throughput of the network may reduce more than this.

Effects of multiple attackers with redirection attacks: We simulated AODV and McDV under redirection attacks by varying the nodes speed from 0m/s to 20m/s while setting the number of attack nodes to 1, 2 and 4 nodes, respectively. We first study the packet delivery ratio and packet dropped ratio. From the results of Fig. 2(a), we can see that packet delivery ratio drops as the speed increases when we use AODV routing protocol under redirection attacks. Meanwhile, we observe that given the same speed of nodes, the higher the number of attackers, the lower the packet delivery ratio in AODV. The packet delivery ratio in the case of 4 attackers declines dramatically to 43% as the speed of nodes



(a) Packet Delivery Ratio under multiple redirection attacks

(b) RREQ Ratio under multiple redirection attacks

(c) End-to-End Delay under multiple redirection attacks

(d) Throughput under multiple redirection attacks

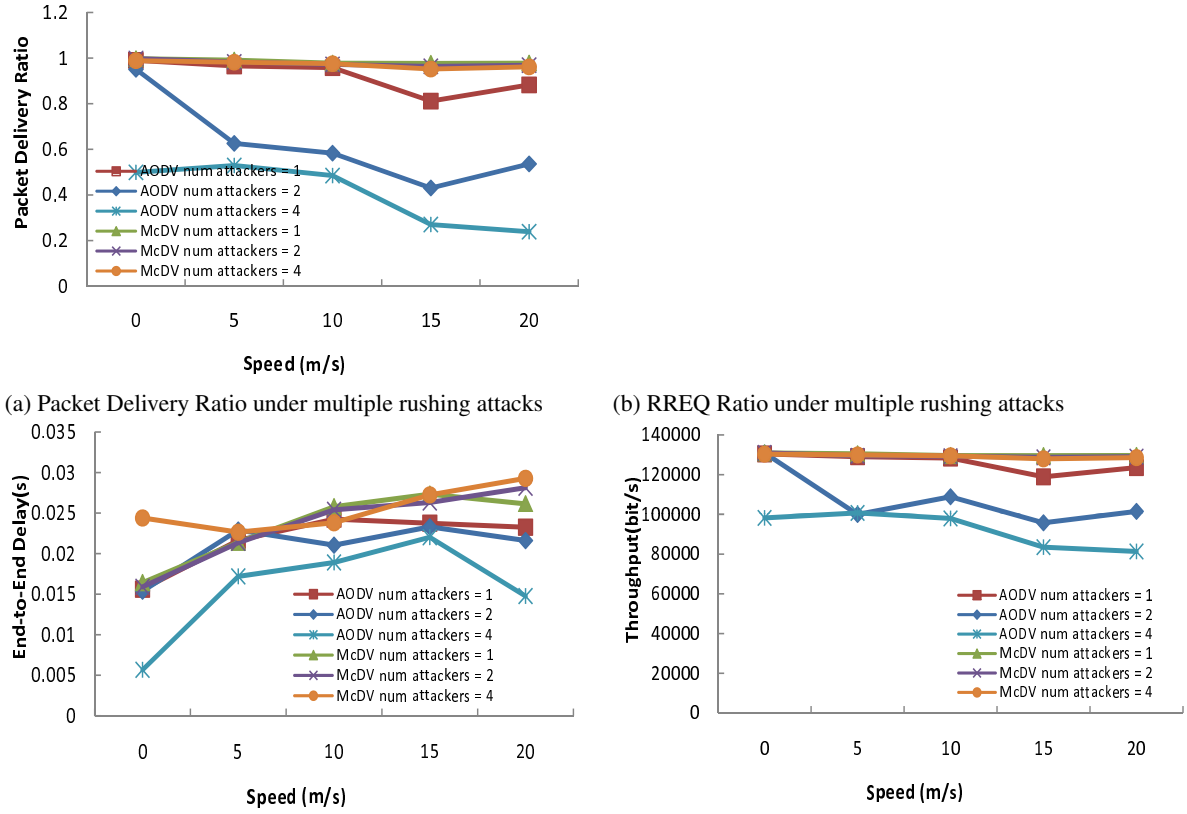
Figure 2: Effects of multiple attackers with redirection attacks

increases to 15m/s. In contrast, the packet delivery ratio of McDV maintains from 94% to 98% even the number of attackers increases to 4 which is slightly lower than normally packet delivery ratio as we can see in Fig. 1(a). All of these are brought by the fact that our routing scheme retains most of the AODV's mechanisms and the extra operations of sign phase and verification phase are very low.

As we would expect from Fig. 2(b), McDV is able to detect all redirection attacks and the packet dropped ratio is zero. On the contrary, as the attack nodes increase, the packet dropped ratio also rises at the same speed when using AODV. Especially, the highest packet dropped ratio of AODV is almost 25% at speed of 15m/s. McDV can detect all the attacks because the node will verify the signature when it receives the packet. Only if this packet passes the verification, the receiving node updates its routing table entry according to the information carried in the packet. Otherwise, the node will drop this packet.

Readers may note that in Fig. 2(c), given the same speed of nodes, the end-to-end Delay in the McDV under redirection attacks are slightly higher than the end-to-end delay in the AODV under redirection attacks. This is simply due to our definition of the end-to-end delay, which is defined as the time a packet takes to travel from the source to the destination. Given the same network size, the same number of senders and the same number of receivers, as attacker or more attackers are added to the network, the number of available nodes forwarding packet decreases, making the average end-to-end delay decrease.

The result in Fig. 2(d) shows the throughput in the network. We can see that the higher the attackers, the lower the throughput at the same speed in AODV. As the speed goes up in AODV, the throughput of network decreases. When the speed is 15m/s, the throughput of AODV drops to 76% comparing with that of original protocol. In contrast, our scheme has the similar trend as the original AODV protocol. As the speed is 15m/s and the network is under 4 redirection attackers, the most effect of throughput is



(a) Packet Delivery Ratio under multiple rushing attacks

(b) RREQ Ratio under multiple rushing attacks

(c) End-to-End Delay under multiple rushing attacks

(d) Throughput under multiple rushing attacks

Figure 3: Effects of multiple attackers with rushing attacks

around 0.9%.

Effects of multiple attackers with rushing attacks: In this section, we compare the varying metrics of AODV and that of McDV under 1, 2 and 4 rushing attacks, respectively. The graph in Fig. 3(a) shows that, the higher the nodes speed, the lower the packet delivery ratio is when using AODV. However, the packer delivery ratio declines dramatically to 24% as the number of attackers increases to 4 nodes and at the speed of 20m/s. On the other hand, the lowest packet delivery ratio in McDV still maintains 95% when the nodes are at the speed of 15m/s. The Fig. 3(b) shows that, given the same speed, the higher the attacker node(s), the higher the packet dropped ratio is under AODV. In contrast, McDV can detect all the rushing attacks, thus the packet dropped ratio is zero.

These results indicate that the AODV protocol performs worse under the rushing attacks than under the redirection attacks. This is because we set the transmit distance as 740m to simulate the rushing attacks. In this situation, the malicious nodes may readily access to the forwarding group and discard all data packets. With the number of attackers increasing, the packet delivery ratio decreases and the packet dropped ratio rises. In contrast, McDV maintains high packet delivery ratio and the packet dropped ratio is zero. This is due to its less computation overhead and efficient implementation of signature and verification.

The Fig. 3(c) shows that the end-to-end delay in rushing attacks. McDV end-to-end delay is slightly higher than AODV end-to-end delay. The explanation for this is similar to the situation discussed in the case of redirection attacks. The difference is that when a node is converted to attacker, the probability of this attacker being selected into the forwarding group increases, and the average end-to-end delay decreases.

Fig. 3(d) shows the throughput of two protocols under rushing attacks. Although mechanisms of redirection attacks and rushing attacks are different, they have a similar way to affect the throughput. The throughput drops more severely under rushing attacks than under redirection attacks. In particular, the lowest throughput almost drops to 63% under 4 rushing attacks when nodes at the speed of 20m/s. In contrast to the AODV protocol under rushing attacks, our scheme has very similar throughput to the original protocol.

7 Conclusion

An efficient certificateless signature scheme named McCLS is proposed in this paper. This scheme is based on the bilinear Diffie-Hellman assumption in the random oracle model for emergency mobile wireless cyber-physical systems. Since McCLS only requires one pairing operation in the verification phase, and none in the signing phase, theoretically it is more efficient than existing certificateless signature schemes. We also present simulation of McDV which is based on McCLS scheme and compare its performance under two most common attacks (i.e. *redirection attack* and *rushing attack*) with typical protocol-AODV providing no protection mechanism. These results show that McDV can completely resist the two kinds of attacks without causing substantial degradation of network performance. In the future, we will further investigate security schemes in the wide physical environment. Thereby we can find schemes which either prevent more comprehensive external attacks or resist internal attacks from the compromised nodes.

8 Acknowledgment

This work was supported in part by an NSERC discovery grant 341823-07 and a National Study-Abroad Scholarship of P.R.China under Grant No. [2007] 3020. Part of this work has been published in preliminary form in the proceedings of The First International Workshop on Cyber-Physical Systems, in conjunction with ICDCS 2008, Beijing, China.

Bibliography

[1] S. S. Al-Riyami and K. G.Paterson. Certificateless Public Key Cryptography. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*. LNCS, Springer-Verlag, 2003.

[2] E. A.Lee. Cyber-Physical Systems - Are Computing Foundations Adequate. Technical report, UC Berkeley, 2006.

[3] M. Anand, E. Cronin, and M. Sherr. Security Challenges in Next Generation Cyber Physical Systems. Technical report, University of Pennsylvania, 2007. http://www.truststc.org/scada/papers/paper33.pdf.

[4] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L.Wolf. A Cluster-based Security Architecture for Ad Hoc Networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2393–2403 vol.4, 7-11 March 2004.

[5] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[6] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology - CRYPTO 2001: 21st Annual International Cryptology Conference*, volume 2139, pages 213–229. LNCS, 2001.

[7] V. Daza, J. Herranz, P. Morillo, and Carla. Cryptographic techniques for mobile ad-hoc networks. *Comput. Networks*, 51(18):4938–4950, 2007.

[8] H. Deng, A. Mukherjee, and D. P. Agrawal. Threshold and Identity-Based Key Management and Authentication for Wireless Ad Hoc Networks. In *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004*, pages 107– 111, April 2004.

[9] B. D.Noble and J. Flinm. Wireless,Self-organizing Cyber-physical systems. Technical report, University of Michigan, 2006. http://varma.ece.cmu.edu/cps/Position-Papers/Noble-Flinn.pdf.

[10] W. He, Y. Huang, K. Nahrstedt, and W. C. Lee. Smock: A self-contained public key management scheme for mission-critical wireless ad hoc networks. In *PERCOM '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pages 201–210, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. In *Proc of the ACM Workshop on Wireless Security (WiSe 2003)*, pages 30–40, 2003.

[12] X. Huang, W. Susilo, Y. Mu, and F. Zhang. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In *International Conference on Cryptology and Network Security (CANS), LNCS*, volume 4, 2005.

[13] A. Khalili, J. Katz, and W. Arbaugh. Toward Secure Key Distribution in Truly Ad Hoc Networks. In *Proc. IEEE Workshop Security and Assurance in Ad Hoc Networks*, pages 342– 346, Jan 2003.

[14] X. Li, K. Chen, and L. Sun. Certificateless Signature and Proxy Signature Schemes from Bilinear Pairings. *Lithuanian Mathematical Journal*, 45(1), 2005.

[15] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks. *IEEE/ACM Transactions on Networking*, 12(6):1049–1063, 2004.

[16] National Science Foundation. Cyber-physical systems. Technical report, NSF Workshop on Cyber-Physical Systems, 2006. http://varma.ece.cmu.edu/cps/.

[17] National Science Foundation. Computer systems research. Technical report, NSF, 2007. http://www.nsf.gov/pubs/2007/nsf07504/nsf07504.htm.

[18] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. A secure routing protocol for ad hoc networks. *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 78–87, 12-15 Nov. 2002.

[19] Scalable Network Technologies. QualNet Simulator. http://www.scalable-networks.com/.

[20] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO: Proceedings of Crypto*, 1984.

[21] S. Xu, Y. Mu, and W. Susilo. Online/Offline Signatures and Multisignatures for AODV and DSR Routing Security. In *11th Australasian Conference on Information Security and Privacy,ACISP 2006*. LNCS, 2006.

[22] W.-S. Yap, S.-H. Heng, and B.-M. Goi. An efficient certificateless signature scheme. In *EUC Workshops*, volume 4097 of *Lecture Notes in Computer Science*, pages 322–331, 2006.

[23] S. Yi and R. Kravets. Moca: Mobile Certificate Authority for Wireless Ad Hoc Networks. In *Proc. Second Ann. PKI Research Workshop (PKI '03)*, Apr 2003.

[24] H. Yoon, J. H. Cheon, and Y. Kim. Batch Verifications with ID-Based Signatures. In *ICISC: International Conference on Information Security and Cryptology*. LNCS, 2004.

[25] Y. Zhang, W. Liu, W. Lou, Y. Fang, and Y. Kwon. AC-PKI: Anonymous and Certificateless Public-key Infrastructure for Mobile Ad Hoc Networks. In *2005 IEEE International Conference on Communications, 2005. ICC 2005*, pages 3515–3519, May 2005.

[26] Z. Zhang, D. S. Wong, J. Xu, and D. Feng. Certificateless Public-Key Signature: Security Model and Efficient Construction. In *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 293–308, 2006.

[27] L. Zhou and Z. Haas. Securing Ad Hoc Networks. *Network, IEEE*, 13(6):24–30, Nov/Dec 1999.

Zhong Xu[1,2], Xue Liu
[1]McGill University
School of Computer Science
3480 University Street, Montreal, Quebec, Canada, H3A 2A7
E-mail: {zhongxu,xueliu}@cs.mcgill.ca

Guoqing Zhang
[2]Northwestern Polytechnical University
College of Automation
Xi'an, Shaanxi, China
E-mail: gqzhang@cs.mcgill.ca

Wenbo He
University of Illinois at Urbana-Champaign
Dept. of Computer Science
Urbana, IL, USA.
E-mail: wenbohe@uiuc.edu

Zhong Xu received the B.E Degree in Automation from Xi'an Technological University in 2001 and the M.E Degree in Computer Science from XiDian University in 2005. Currently, Zhong is a joint Ph.D student in McGill University, Montreal, Canada and Northwestern Polytechnical University, Xi'an, China. From August 2001 to August 2002, he was an assistant lecturer in Xi'an Technological University, China. His research interests include Security of Ad Hoc Networks, Information Security, Embedded Systems and Cyber-Physical Systems.

Dr. Xue (Steve) Liu is an Assistant Professor in the School of Computer Science at McGill University. He is also affiliated to the Centre for Intelligent Machines (CIM). Xue obtained his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2006. He obtained his B.S. degree in Mathematics and M.S. degree in Automatic Control both from Tsinghua University, China. He worked briefly in the Hewlett-Packard LabsIBM T. J. Watson Research Center. He received the Ray Ozzie Fellowship, the Saburo Muroga Fellowship, the Mavis Memorial Fund Award, and the C. W. Gear Outstanding Graduate Award, from the University of Illinois at Urbana-Champaign. He has filed 5 patents, and published more than 50 research papers in international journals and major peer-reviewed conference proceedings.

Guoqing Zhang received the B.E Degree and M.E. degree in Automation both from Northwestern Polytechnical University. Currently, he is a Ph.D student in Northwestern Polytechnical University, Xi'an, China. His research interests include Vehicular Ad-hoc Networks, Information Security and Embedded Systems.

Wenbo He is currently a Ph.D student at Department of Computer Science, in University of Illinois at Urbana-Champaign, where she is advised by Professor Klara Nahrstedt. She received the Mavis Memorial Fund Scholarship Award from College of Engineering of UIUC in 2006, and C. W. Gear Outstanding Graduate Award from Department of Computer Science in 2007. She is also a recipient of Vodafone Fellowship in 2005-2008. Wenbo received the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2000. She received the M.Eng. degree in automatic control theory from Tsinghua University, Beijing, China, in 1998, and the B.E. degree in automatic control from the Harbin Engineering University, Heilongjiang, China, in 1995. From August 2001 to January 2005, she was a Software Engineer with Cisco Systems Inc., Champaign, IL. Her research interests include pervasive and mobile computing, and network security and privacy.

Author index

Description

International Journal of Computers, Communications & Control (IJCCC) is a quarterly peer-reviewed publication started in 2006 by Agora University Editing House - CCC Publications, Oradea, ROMANIA.

Beginning with 2007, EBSCO Publishing is a licensed partner of IJCCC Publisher.

Every issue is published in online format (ISSN 1841-9844) and print format (ISSN 1841-9836).

Now we offer free online access to the full text of all published papers.

The printed version of the journal should be ordered, by subscription, and will be delivered by regular mail.

IJCCC is directed to the international communities of scientific researchers from the universities, research units and industry.

IJCCC publishes original and recent scientific contributions in the following fields:

- Computing & Computational Mathematics
- Information Technology & Communications
- Computer-based Control

To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).

The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by minimum two reviewers and will be published only after end of the editorial workflow.

The peer-review process is single blinded: the reviewers know who the authors of the manuscript are, but the authors do not have access to the information of who the peer-reviewers are.

IJCCC also publishes:

- papers dedicated to the works and life of some remarkable personalities;
- reviews of some recent important published books

Also, IJCCC will publish as supplementary issues the proceedings of some international conferences or symposiums on Computers, Communications and Control, scientific events that have reviewers and program committee.

The authors are kindly asked to observe the rules for typesetting and submitting described in Instructions for Authors.

Editorial Workflow

The editorial workflow is performed using the online Submission System.

The peer-review process is single blinded: the reviewers know who the authors of the manuscript are, but the authors do not have access to the information of who the peer-reviewers are.

The following is the editorial workflow that every manuscript submitted to the IJCCC during the course of the peer-review process.

Every IJCCC submitted manuscript is inspected by the Editor-in-Chief/Associate Editor-in-Chief. If the Editor-in-Chief/Associate Editor-in-Chief determines that the manuscript is not of sufficient quality to go through the normal review process or if the subject of the manuscript is not appropriate to the journal scope, Editor-in-Chief/Associate Editor-in-Chief *rejects the manuscript with no further processing*.

If the Editor-in-Chief/Associate Editor-in-Chief determines that the submitted manuscript is of sufficient quality and falls within the scope of the journal, he sends the manuscript to the IJCCC Executive Editor/Associate Executive Editor, who manages the peer-review process for the manuscript.

The Executive Editor/Associate Executive Editor can decide, after inspecting the submitted manuscript, that it should be rejected without further processing. Otherwise, the Executive Editor/Associate Executive Editor assigns the manuscript to the one of Associate Editors.

The Associate Editor can decide, after inspecting the submitted manuscript, that it should be rejected without further processing. Otherwise, the Associate Editor assigns the manuscript to minimum two external reviewers for peer-review. These external reviewers may or may not be from the list of potential reviewers of IJCCC database.

The reviewers submit their reports on the manuscripts along with their recommendation of one of the following actions to the Associate Editor: Publish Unaltered; *Publish after Minor Changes; Review Again after Major Changes; Reject* (Manuscript is flawed or not sufficiently novel).

When all reviewers have submitted their reports, the Associate Editor can make one of the following editorial recommendations to the Executive Editor: Publish Unaltered; Publish after Minor Changes; Review Again after Major Changes; Reject.

If the Associate Editor recommends *"Publish Unaltered"*, the Executive Editor/Associate Executive Editor is notified so he/she can inspect the manuscript and the review reports. The Executive Editor/Associate Executive Editor can either override the Associate Editor's recommendation in which case the manuscript is rejected or approve the Associate Editor's recommendation in which case the manuscript is accepted for publication.

If the Associate Editor recommends "Review Again after Minor Changes", the Executive Editor/Associate Executive Editor is notified of the recommendation so he/she can inspect the manuscript and the review reports.

If the Executive Editor/Associate Executive Editor overrides the Associate Editor's recommendation, the manuscript is rejected. If the Executive Editor approves the Associate Editor's recommendation, the authors are notified to prepare and submit a final copy of their manuscript with the required minor changes suggested by the reviewers. Only the Associate Editor, and not the external reviewers, reviews the revised manuscript after the minor changes have been made by the authors. Once the Associate Editor is satisfied with the final manuscript, the manuscript can be accepted.

If the Associate Editor recommends *"Review Again after Major Changes"*, the recommendation is communicated to the authors. The authors are expected to revise their manuscripts in accordance with the changes recommended by the reviewers and to submit their revised manuscript in a timely manner. Once the revised manuscript is submitted, the original reviewers are contacted with a request to review the revised version of the manuscript. Along with their review reports on the revised manuscript, the reviewers make a recommendation which can be "Publish Unaltered" or "Publish after Minor Changes" or "Reject". The Associate Editor can then make an editorial recommendation which can be "Publish Unaltered" or "Review Again after Minor Changes" or "Reject".

If the Associate Editor recommends rejecting the manuscript, either after the first or the second round of reviews, the rejection is immediate.

Only the Associate Editor-in-Chief can approve a manuscript for publication, where Executive Editor/Associate Executive Editor recommends manuscripts for acceptance to the Editor-in-Chief/Associate Editor-in-Chief.

Finally, recommendation of acceptance, proposed by the Associate Editor Chief, has to be approved by the Editor-in-Chief before publication.

Instructions for authors

The papers must be prepared using a LaTeX typesetting system. A template for preparing the papers is available on the journal website http://journal.univagora.ro. In the template.tex file you will find instructions that will help you prepare the source file. Please, read carefully those instructions. (We are using MiKTeX 2.4).

Any graphics or pictures must be saved in Encapsulated PostScript (.eps) format.

Papers must be submitted electronically to the following address: ccc@univagora.ro. You should send us the LaTex source file (just one file - do not use bib files) and the graphics in a separate folder. You must send us also the pdf version of your paper.

The maximum number of pages of one article is 20. The publishing of a 12 page article is free of charge (including a bio-sketch). For each supplementary page there is a fee of 50 Euro/page that must be paid after receiving the acceptance for publication. The authors do not receive a print copy of the journal/paper, but the authors receive by email a copy of published paper in pdf format.

The papers must be written in English. The first page of the paper must contain title of the paper, name of author(s), an abstract of about 300 words and 3-5 keywords. The name, affiliation (institution and department), regular mailing address and email of the author(s) should be filled in at the end of the paper. Manuscripts must be accompanied by a signed copyright transfer form. The copyright transfer form is available on the journal website.

Please note: To avoid unnecessary delays in publishing you are kindly asked to consider all recommendations expressed in the template. We do not accept submissions in other formats (pdf only, Microsoft Word, etc).

Checklist:

1. Completed copyright transfer form.
2. Source (input) files.
    - One LaTeX file for the text.
    - EPS files for figures in a separate folder.
3. Final PDF file (for reference).

Order

If you are interested in having a subscription to "Journal of Computers, Communications and Control", please fill in and send us the order form below:

ORDER FORM		
I wish to receive a subscription to "Journal of Computers, Communications and Control"		
NAME AND SURNAME:		
Company:		
Number of subscription:	Price Euro	for issues yearly (4 number/year)
ADDRESS: City: Zip code: Country: Fax: Telephone: E-mail:		
Notes for Editors (optional)		

1. Standard Subscription Rates for Romania (4 issues/2007, more than 400 pages, including domestic postal cost): 90 EURO.

   2. Standard Subscription Rates for other countries (4 issues/2007, more than 400 pages, including international postal cost): 160 EURO.

For payment subscription rates please use following data:

HOLDER: Fundatia Agora, CUI: 12613360
BANK: BANK LEUMI ORADEA
BANK ADDRESS: Piata Unirii nr. 2-4, Oradea, ROMANIA
IBAN ACCOUNT for EURO: RO02DAFB1041041A4767EU01
IBAN ACCOUNT for LEI/ RON: RO45DAFB1041041A4767RO01
SWIFT CODE (eq. BIC): DAFBRO22

Mention, please, on the payment form that the fee is "for IJCCC".

EDITORIAL ADDRESS:

CCC Publications
Piata Tineretului nr. 8
ORADEA, jud. BIHOR
ROMANIA
Zip Code 410526
Tel.: +40 259 427 398
Fax: +40 259 434 925
E-mail: ccc@univagora.ro, Website: www.journal.univagora.ro

Copyright Transfer Form

To The Publisher of the International Journal of Computers, Communications & Control

This form refers to the manuscript of the paper having the title and the authors as below:

The Title of Paper (hereinafter, "Paper"):

.....

The Author(s):

.....

.....

.....

.....

The undersigned Author(s) of the above mentioned Paper here by transfer any and all copyright-rights in and to The Paper to The Publisher. The Author(s) warrants that The Paper is based on their original work and that the undersigned has the power and authority to make and execute this assignment. It is the author's responsibility to obtain written permission to quote material that has been previously published in any form. The Publisher recognizes the retained rights noted below and grants to the above authors and employers for whom the work performed royalty-free permission to reuse their materials below. Authors may reuse all or portions of the above Paper in other works, excepting the publication of the paper in the same form. Authors may reproduce or authorize others to reproduce the above Paper for the Author's personal use or for internal company use, provided that the source and The Publisher copyright notice are mentioned, that the copies are not used in any way that implies The Publisher endorsement of a product or service of an employer, and that the copies are not offered for sale as such. Authors are permitted to grant third party requests for reprinting, republishing or other types of reuse. The Authors may make limited distribution of all or portions of the above Paper prior to publication if they inform The Publisher of the nature and extent of such limited distribution prior there to. Authors retain all proprietary rights in any process, procedure, or article of manufacture described in The Paper. This agreement becomes null and void if and only if the above paper is not accepted and published by The Publisher, or is withdrawn by the author(s) before acceptance by the Publisher.

Authorized Signature (or representative, for ALL AUTHORS): .....

Signature of the Employer for whom work was done, if any: .....

Date: .....

Third Party(ies) Signature(s) (if necessary): .....