CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Information Technology

Proceedings of the

7th Prague Embedded Systems Workshop

June 27-29, 2019

Roztoky u Prahy, Czech Republic

Editors:

doc. Ing. Hana Kubátová, CSc.
doc. Ing. Petr Fišer, Ph.D.
Ing. Jaroslav Borecký, Ph.D.

ISBN 978-80-01-06607-2

Message from the Program Chairs

The Prague Embedded Systems Workshop is a research meeting intended for the presentation of Ph.D. students' results and partial progress in their research in the field of all aspects of embedded systems design, including their testing, reliability, secure, safe, and low-power applications and communications. The workshop is organized annually by members of the Department of Digital Design of the Faculty of Information Technology of the Czech Technical University in Prague, for the seventh time this year. The main aim of PESW is to boost mutual discussions and establishing possible future cooperation between young researches not only inside EU. Therefore, the PESW workshop will be based on oral presentations and discussions.

There are three types of students´ submissions and presentations at PESW 2019:

- Full papers describing the student's original research. These papers will undergo a standard reviewing process, and if accepted, they will be included in the Proceedings with ISBN.
- Abstracts of authors' earlier published and successfully presented papers (at conferences, journals, etc.). These contributions will not be reviewed; emphasis will be put on the presentation and discussion. These abstracts will be included in the Proceedings with ISBN.
- Student posters - abstracts of defended Bc. and MSc. theses with subsequent poster presentation. These abstracts will not be included in the Proceedings.

16 papers were accepted for PESW 2019 presentation, from which there were 1 full paper and 15 abstracts. Contributions from Czech, Polish and Italian university research teams were accepted this year.

The technical program is also highlighted by three keynote speakers in the areas of security, testing, and network monitoring:

- Randomness in emerging technologies: Functional robustness vs. security.
  Speaker: Elena-Ioana Vatajelu, TIMA - CNRS / Université Grenoble Alpes, France
- Automotive testing challenges.
  Speaker: Paolo Bernardi, Politecnico di Torino, Italy
- Hardware Acceleration Techniques for Network Monitoring and Security.
  Speaker: Jan Kořenek, BUT, Brno, Czech Rep.; CESNET.

Seven technical sessions were formed, with the following topics:

- Fault Tolerance & Reliability
- Testing & Test Generation
- Security
- Cryptosystems & Cryptanalysis
- Network Monitoring
- Traffic Processing and Analysis
- Other

Last but not least we would like to thank to our sponsors (CTU in Prague, ASICentrum, STMicroelectronics, CESNET, CZ.NIC, ISECO.cz). Special thanks go to IEEE: IEEE Student Branch at Czech Technical University in Prague and IEEE Young Professionals, organizing student contest, and Czechoslovakia Section of IEEE.

We wish you to spend fruitful and communicative time in Roztoky.

Hana Kubátová and Petr Fišer

Committees

Workshop Chairs

Hana Kubátová, CTU in Prague (CZ)

Petr Fišer, CTU in Prague (CZ)

Programme Committee (preliminary)

P. Bernardi, Politecnico di Torino (IT)

A. Bosio, École Centrale de Lyon (FR)

J. Buček, CTU in Prague (CZ)

T. Čejka, CTU in Prague, Prague (CZ)

G. Natale, TIMA, Grenoble (FR)

P. Fišer, CTU in Prague, Prague (CZ)

J.L. Gaudiot, University of California, Irvine (USA)

I. Levin, Tel-Aviv University (IL)

K. Jelemenská, STU Bratislava (SK)

L. Kekely, BUT, Brno (CZ)

P. Kitsos, TEI West. Greece (GR)

Z. Kotásek, BUT, Brno (CZ)

H. Kubátová, CTU in Prague, Prague (CZ)

F. Leporati, Univ. di Pavia (GR)

R. Lórencz, CTU in Prague (CZ)

A. McEwan, University of Leicester (UK)

N. Mentens, KU Leuven (BE)

P. Mróz, University of Zielona Gora (PL)

M. Novotný, CTU in Prague, Prague (CZ)

A. Orailoglu, UC San Diego (USA)

M. Ottavi, University of Rome (IT)

E. Sanchez, Politecnico di Torino (IT)

J. Schmidt, CTU in Prague, Prague (CZ)

M. Skrbek, CTU in Prague, Prague (CZ)

R. Stojanovic, Univ. of Podgorica Montenegro (ME)

J. Strnadel, BUT, Brno (CZ)

R. Ubar, Tallinn Univ. of Technology (EE)

P. Velan, ICS MUNI (CZ)

H.T. Vierhaus, Brandenburg University of Technology (DE)

W. Zając, University of Zielona Gora (PL)

Special Session on Network Security Chair

Tomáš Čejka, CTU in Prague (CZ)

Student Poster Session Co-Chairs

Tomáš Kolárik, CTU in Prague (CZ)

Jan Bělohoubek, CTU in Prague (CZ)

Organizing Committee

H. Kubátová, CTU in Prague (CZ)

P. Fišer, CTU in Prague (CZ)

R. Kinc, AMCA (CZ)

E. Uhrová, AMCA (CZ)

Contents

Keynotes

Randomness in emerging technologies: Functional robustness vs. security

Speaker: **Elena-Ioana Vatajelu**, *TIMA - CNRS / Université Grenoble Alpes*

The rapid development of low power, high density, high performance SoCs has pushed the CMOS devices to their limits and opened the field to the development of emerging technologies. The STT-MRAM and RRAM have emerged as promising choices for embedded memories due to their reduced read/write latency and high CMOS integration capability. Their inner properties make them ideal for implementation of memory blocks (mach and main memory) and, in addition, they are suitable for the implementation of basic security primitives such Physically Unclonable Functions (PUFs) and True Random Number Generators (TRNGs). PUFs are emerging primitives used to implement low-cost device authentication and secure secret key generation. On the other hand, TRNGs generate random numbers from a physical process. This talk will present a survey of today's and tomorrow's technologies and explain how it is possible to exploit (i) the high variability affecting the electrical device characteristics to build a robust, unclonable and unpredictable PUF, and (ii) the stochastic characteristics to generate randomly distributed numbers. In addition, it will underline the conflict between functional robustness and security quality of ICs designed with such devices.

Elena-Ioana Vatajelu

Dr. Elena-Ioana Vatajelu is a researcher with CNRS in TIMA Laboratory, Grenoble, France. She has 10 years of research experience in design, test and reliability of Integrated Circuits. She received a PhD in Electronic Engineering with distinction from Universitat Politècnica de Catalunya (Spain) in 2011. She has been involved in several European Projects (FP5 and FP7) and Spanish and Italian National projects. Dr. Vatajelu has served on the Technical Program Committees and Organizing Committees of conferences and symposia in design automation and test domains, such as DATE, IEEE VTS, IEEE ETS, IEEE DCIS, IEEE DDECS. Her main research interests are on reliability and robustness assessment, design-for-reliability, test strategies and security primitives for CMOS and beyond CMOS RAMs in traditional and non-Von Neumann computing (neuromorphic and CIM) paradigms. She has published 50 journal and conference papers in the area of dependable memories.

Automotive testing challenges

Speaker: **Paolo Bernardi**, *Politecnico di Torino, Italy*

Manufacturing Automotive System-on-Chip is becoming always more challenging. That's because of the current complexity of the functionality to design, and also due to the very stringent quality requirements this kind of devices must meet. It is estimated that the quality aspects are weighting up to the 50% of the entire productive flow costs, since they "pollute" the conception of the chip with bulky test oriented circuitry and demand for several expensive test equipment to be used to ensure a perfect product being sold. The talk will depict a general scenario about all efforts to put in the manufacturing flow of a today's automotive chip, including technology qualification, design for testability, wafer sort/final test/burn-in/system level test, in-field self-test, certification tools and field return failure analysis.

Paolo Bernardi

Paolo Bernardi (MS'02 and PhD'06 in Computer Science) is an Associate Professor of the Politecnico di Torino University, where he works in the Electronic CAD and Reliability research group. His current interests includes System-on-Chip test and reliability, especially in the direction of high quality automotive devices. Prof. Bernardi is the General Chair of the Test Technology Educational Program (TTEP) and the Program Chair of the Automotive Reliability and Test (ART) Workshop held in conjunction with the International Test Conference. He was recently acting as Topic Chair for the European Test Symposium (ETS), the Design and Diagnosis of Electronic Circuits Symposium (DDECS) and the International On-Line Test Symposium (IOLTS). In 2018, he has been the General Chair of the Design and Technology of Integrated Circuits (DTIS) conference.

Hardware Acceleration Techniques for Network Monitoring and Security

Speaker: **Jan Kořenek**, *BUT, Brno, Czech Rep.; CESNET*

High-speed packet processing is important especially in network monitoring and in security systems, where any packet drop can decrease the precision of monitoring or avoid detection or mitigation of malicious traffic. Current CPUs are not able to provide enough performance for security analysis of network traffic, especially in high-speed networks. To achieve wire-speed 100 Gbps throughput every packet has to be processed in less than 5 ns. Therefore the talk will summarise time-critical operations in network security systems, which require hardware acceleration. Then It will be introduced how deep pipelines, perfect hashing, and pipelined automata can help to achieve 100 Gbps packet processing of network security systems. The talk will address also the flexibility of hardware acceleration and integration of hardware architectures into future SmartNIC devices.

Jan Kořenek

Jan Kořenek is an associate professor at Brno University of Technology. Jan has been working since 2002 on many European and national research projects, where FPGA technology was used for an acceleration of IPv6 protocol routing, network traffic monitoring, NetFlow statistic measurement and fast regular expression matching in a packet payload. These projects provide substantial experiences in the hardware acceleration of algorithms for network applications and devices. Since 2003, He worked for CESNET as a leader of Hardware group at Liberouter project. In May 2007, He co-founded INVEA-TECH company which is a university spin-off focused on high speed network monitoring and security systems. Jan is an author and co-author of many novel hardware architectures, which has been used in commercially successful devices. For example, he is an co-author of COMBO-CG 100 Gb card, which received Czech Head award in the category Industria. His research interests are in the areas of hardware acceleration, reconfigurable architectures, embedded systems and network security and monitoring. Since 2012, He has been the head of Security and Administration Tools (SAT) department at CESNET. The SAT department is focused on research and development of new tools for network infrastructure. CESNET is an association of universities of the Czech Republic and the Czech Academy of Sciences.

Smart Electronic Locks and Their Reliability

Ondřej Čekan, Jakub Podivínský, Jakub Lojda, Richard Pánek,
Martin Krčma, Zdeněk Kotásek
Brno University of Technology, Faculty of Information Technology,
Centre of Excellence IT4Innovations
Božetěchova 2, 612 66 Brno, Czech Republic

{icekan,ipodivinsky,ilojda,ipanek,ikrcma,kotasek}
@fit.vutbr.cz

Keywords. Electronic Lock, Stepper Motor, FPGA, Fault Tolerance, Stimuli Generation.

Abstract

Our research focuses on an analysis of electronic smart locks and explores the influences of faults on its controller unit. Electronic smart locks often utilize stepper motor as an actuator. Stepper motors, however, need a controller, which is usually implemented in a processor. The aim of our research is to examine the consequences of a failing controller processor. In our previous research, we developed a platform for fault tolerance testing with the ability to monitor the impacts on the mechanical part. We also developed a framework for accelerated testing of fault tolerance properties. The processor can be implemented in an FPGA (Field Programmable Gate Array) in order to be able to emulate HW faults inside the processor.

The concept of testing a smart lock is presented in Fig. 1, where all components are running on PC which allows us rapid prototyping and evaluation. Our experimental results utilizing the direct generation of invalid stimuli for the stepper motor. In our research, we found out that random errors probably could not be used for an unauthorized unlock, especially if the lock utilizes a mechanical gearbox. Deeper logic and knowledge of the correct sequence of steps used by the selected motor are needed to perform an attack to unlock the lock. On the other hand, random sequences could cause that lock not to be locked by falsifying the lock request sequence. The second interesting fact is that *x%* of faults in the valid sequence give the same rotation angle as *100-x%* of faults.

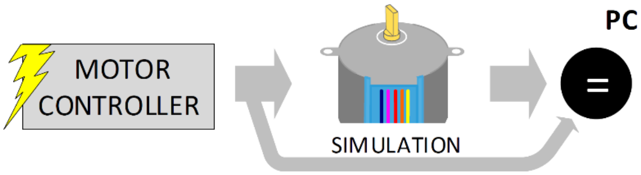


Fig. 1: The concept of testing a smart lock – the first step.

Paper origin

The original paper has been accepted at 22nd Euromicro Conference on Digital System Design in Kallithea, Chalkidiki, Greece [1].

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II), the project IT4Innovations excellence in science - LQ1602, the BUT project FIT-S-17-3994 and the JU ECSEL Project SECREDAS (Product Security for Cross Domain Reliable Dependable Automated Systems), Grant agreement No. 783119.

References

[1] Čekan, O.; Podivínský, J.; Lojda, J.; Pánek, R.; Krčma, M.; Kotásek, Z.: Testing Reliability of Smart Electronic Locks: Analysis and the First Steps Towards. In: 2019 Euromicro Conference on Digital System Design. Kallithea: IEEE Computer Society, 2019, accepted for publishing.

Fault Recovery for Coarse-Grained TMR Soft-Core Processor Using Partial Reconfiguration and State Synchronization

Karel Szurman, Zdeněk Kotásek
Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence
Bozetechova 1/2, 61266 Brno, Czech Republic

{iszurman, kotasek}@fit.vutbr.cz

Keywords. TMR, fault recovery, state synchronization, processor, FPGA reconfiguration

Abstract

SRAM FPGAs are being more commonly integrated into safety-critical systems nowadays. These digital circuits can provide suitable platform for a fault tolerant system implementation meeting the trade-offs between performance, reliability, cost and hardware resources. However, SRAM technology is vulnerable to radiation-induced faults and mainly to Single Event Upset (SEU) effect. The SEU can cause "bit-flip" faults in SRAM memory cells which may affect internal FPGA routing (clock and reset signals), user memory (flip-flops, block RAM) and the functionality of implemented circuits. SEU mitigation must be implemented into the safety-critical design to achieve required system reliability in the harsh environment. SEU mitigation strategy may combine hardware redundancy and Partial Dynamic Reconfiguration (PDR) in order to implement error detection, self-repair ability and fault recovery mechanism into the system. With respect to the compromise between the system reliability and the resource overhead, various hardware redundancy schemes can be used. The most used form is Triple Modular Redundancy (TMR) which can be applied on different granularity levels in the system design. Coarse-grained TMR and PDR are often combined in one reconfigurable architecture. The time between SEU occurrence and the completion of fault recovery become a crucial parameter because the reliability of the TMR with one failed replica is worse than the reliability of an unprotected system. The fault recovery process can be generally divided into three phases: 1) fault detection, 2) fault removal by reconfiguration of a region containing replica identified as faulty, and 3) state synchronization bringing the reconfigured replica into the operating state consistent with other correctly operating replicas.

Combination of TMR and PDR is the approach also often addressed by fault mitigation methods designed for soft-core processors. The processor state is stored in internal memories and various architectural registers. After a faulty processor replica is reconfigured, its internal registers holding the processor state need to be synchronized with their up-to-dated copies from other processors replicas which were correctly operating. Various approaches had been proposed by researchers in the past. Four different synchronization methods which balance differently the trade-off between the synchronization speed and hardware overhead are evaluated in [2]. Synchronization of processors in known-blocking state by dumping and reading all processors data through shared Wishbone interconnection memory is presented in [3]. The largest amount of data which needs to be synchronized is the content of internal memories. With respect to a huge resource overhead, the use of a shared memory accessible from all three processor replicas is only practice solution. The critical part of the processor state synchronization is the maintenance of all internal registers. This requires implementation of a synchronization mechanism directly in the hardware to enable access to all registers and to minimize the synchronization time.

We propose a fault recovery mechanism for soft-core processor NEO430. In our PDR design, the NEO430 CPU core is protected by reconfigurable TMR architecture. In the TMR, the same input signals are shared between all CPU replicas and their output signals are brought into the majority voters. Each TMR voter is enhanced by additional error detection logic for identification of a failed CPU. The FPGA design floorplan is divided into two static and dynamic areas. Replicated CPU instances are placed into dedicated Partial Reconfiguration Modules (PRMs) in the dynamic area. Other design components are static; including reconfiguration controller GPDRC, synchronization controller and TMR voters. In the experiments, the reconfiguration of specific PRM corresponding to the faulty CPU replica is started based on the PRM error vector generated by TMR voters. A test application executed by triplicated CPUs periodically checks the digital input for activation of the synchronization enable request. This signal is generated by the GPDRC after the reconfiguration is finished. Afterwards, repaired CPU is restarted. During its startup, the test application reads the digital inputs and checks if request for synchronization is active. Since the request was activated by GPDRC, the CPU switches into the SLEEP mode. When the application executed by other operating CPUs is in a state suitable for synchronization, it will indicate readiness for the hardware synchronization through processor digital output to a synchronization controller. This is special circuit responsible for parallel addressing of all synchronized registers and their copying from the correctly working CPUs to the recovered one. Then, operating CPUs go into the SLEEP mode as well. In this state, CPUs are waiting for an external IRQ generated by the synchronization controller which will activate normal operating mode. In parallel, the synchronization controller performs synchronization of all internal registers while CPUs are idle in the SLEEP mode. After the hardware synchronization phase is finished, the external IRQ signal is triggered to bring CPUs again into the operating mode. Since that moment, all CPUs continue in synchronized program execution and with consistent data stored in the internal registers. By this FT design, we demonstrated possibility to implement a fault recovery mechanism for soft-core processor with the state synchronization logic embedded into the processor architecture and with the non-blocking CPU execution aware of fault recovery phases.

Paper origin

This paper has been accepted and presented at the 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems in Cluj-Napoca [1].

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science – LQ1602 and the BUT project FIT-S-17-3994.

References

[1] Szurman, K.; Kotasek, Z.: Run-Time Reconfigurable Fault Tolerant Architecture for Soft-Core Processor neo430. 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems. Cluj-Napoca: IEEE Computer Society, 2019, pp. 136-140. ISBN 978-1-72810-072-2.

[2] Kretzschmar, U; Gomez-Cornejo, J.; Astarloa, A.; Bidarte, U.; Del Ser, J.: Synchronization Of Faulty Processors In Coarse-Grained TMR Protected Partially Reconfigurable FPGA Designs. Reliability Engineering & System Safety, 2016.

[3] Morillo, A.; Astarloa, A.; Lazaro, J.; Bidarte, U.; Jimenez, J.: Knownblocking synchronization method for reliable processor using tmr & dpr in sram fpgas. VII Southern Conference on Programmable Logic (SPL), April 2011, pp. 57-62.

Linear cryptanalysis and recovery of key bits in Baby Rijndael

Josef Kokeš, Róbert Lórencz
Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, Praha 6, 16000, Czech Republic

josef.kokes@fit.cvut.cz, robert.lorencz@fit.cvut.cz

Keywords. AES, Rijndael, Simplified AES, Baby Rijndael, Linear cryptanalysis, Key recovery.

Abstract

The Rijndael cipher, published in 1998 and standardized as AES in 2001 [1], is the most widely used symmetric block cipher in the world. It can be found in many practical applications, including disk and file encryption or messaging and communication. A very important are for AES is its appearance in various layers of network protocols such as WiFi or TLS/SSL where it serves to protect the integrity and confidentiality of transferred data.

To ensure the security of any cipher, cryptanalytic techniques are used to study the cipher and evaluate its resistance against various kinds of attacks, each based on specific assumptions. One of the basic techniques is the algebraic cryptanalysis, which tries to express the cipher as a set of equations, and its two special cases, differential cryptanalysis and linear cryptanalysis. Each modern cipher is designed with resistance to these techniques in mind, and AES is no exception [2], but even then these techniques may be successfully used to weaken the security of a cipher, as has been shown by the Biclique attack by Bodganov et al against full AES [3].

Unfortunately, the size of the full AES, which is an important part of its security as it prevents attacks using brute force, also makes it difficult to study in detail. We can, however, make use of the fact that unlike AES, which is highly constrained in its parameters [1], the description of Rijndael is quite open and allows modifications to many of its aspects. This has led to the introduction of several versions of simplified AES variants such as S-AES [4] or Baby Rijndael [5]. While these AES-variants cannot be practically used in real-world scenarios due to their reduced complexity, they strive to preserve as much of the original cipher's structure as possible while allowing a more detailed research due to their much more manageable sized.

In our past research of the effects of linear cryptanalysis on Baby Rijndael [6] we discovered several very interesting properties of the cipher. Particularly, we found out that when performing linear cryptanalysis using Matsui's Algorithm B [7], there are almost 8000 linear approximations with the theoretically best properties, which, however, achieve significantly different levels of success in recovering the cipher's key: For example, linear approximations which terminate in alternating active/inactive S-boxes in the second-to-last round of the cipher are on average much more successful than any other approximations. More, even within the same class of approximations, some approximations lead to better results than others: while the average rank of a recovered key for "bad" approximations is about 114 (out of 256), "good" approximations' correct key rank is about 49, with the actual rank varying between 40 and 57.

Furthermore, it appears that if we consider the bits of the recovered key individually, some bits tend to be more prone to errors than others; for example, bits 3 and 11 of the key can be successfully recovered

with a probability of as much as 70 %. Furthermore, some approximations tend to provide better results for some bits than the others, as has been shown by an exhaustive search of all the possibilities. With this fact in mind, we propose a set of algorithms which would leverage the varying success levels of key-bit recovery of different approximations to achieve an improved ratio of correct key recovery. We start with algorithms focused on the best recovery of a single bit of a key, leading up to a composite algorithm which would recover a set of bits with a higher-than-expected probability. The latest results have shown that we can correctly recover one bit of the key with a probability of more than 81 %. Four bits of the key can be correctly recovered with a probability of more than 49 %, which is a significant improvement over the theoretical 6,25 % of pure guessing.

We expect that a further improvement to these algorithms is possible, and that's where our current research is focused.

Acknowledgment

This research is supported by the the European Union[1], The Ministry of Education, Youth and Sports of the Czech Republic, Czech Technical University in Prague and The Research Center for Informatics.

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MŠMT
MINISTRY OF EDUCATION,
YOUTH AND SPORTS

CTU
CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

RESEARCH
CENTER FOR
INFORMATICS
rci.cvut.cz

References

[1] FIPS: Advanced Encryption Standard (AES), Federal Information Processing Standards, 2001, doi:10.6028/NIST.FIPS.197

[2] Daemen, J., Rijmen, V.: The design of Rijndael: AES — the Advanced Encryption Standard, Springer-Verlag, 2002, ISBN 3-540-42580-2.

[3] Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES, Advances in Cryptology – ASIACRYPT, 2011.

[4] Musa, M., Schaefer E., Wedig S.: A simplified AES algorithm and its linear and differential cryptanalyses, Cryptologia 27, 2003.

[5] Bergman C.: A Description of Baby Rijndael, Iowa State University, 2005.

[6] Kokeš, J., Lórencz, R.: Linear Cryptanalysis of Baby Rijndael, IMACC, 2015.

[7] Matsui, M.: Linear Cryptanalysis Method for DES Cipher, Lecture Notes in Computer Science, Vol. 765, 1994.

[1] European Structural and Investment Funds, Operational Programme Research, Development and Education

Multiprecision Microcontroller-optimized ANSI C Library for Exotic Cryptosystems

Jan Říha, Jakub Klemsa, Martin Novotný
Czech Technical University in Prague

rihaja11fit.cvut.cz

Keywords. Cryptography, microcontroller, arithmetic.

Abstract

Current cryptographic algorithms work with operands that are several times wider than the machine word, e.g., the still popular RSA algorithm shall use at least 2 048-bit keys. Such algorithms therefore require libraries that implement multiprecision arithmetic. Existing libraries are either not tailored for microcontrollers, or they implement an incomplete set of multiprecision operations, which limits the implementation of some unusual cryptographic algorithms on microcontrollers.

In this work, we present a novel ANSI C library that implements also some less common operations like, e.g., multiprecision integer division. The library was designed with respect to the use on microcontrollers and has been tested on ARM M4-based microcontroller Microchip CEC1302.

Paper origin

This research has been presented at the 8th Mediterranean Conference on Embedded Computing (MECO 2019).

Acknowledgment

This work was supported by the Grant Agency of the Czech Technical University in Prague, grants No. SGS17/213/OHK3/3T/18 and SGS19/109/OHK3/2T/13

Recognition of Semi-trailers on the basis of the image

Tomasz Czech, Małgorzata Mazurkiewicz,
Piotr Mróz, Anna Pławiak-Mowna
Faculty of Computer, Electrical and Control Engineering, University of Zielona Góra
Address Prof. Z. Szafrana Street 2, 65 – 516 Zielona Góra, Poland

E-mail feralnex@gmail.com; m.mazurkiewicz@issi.uz.zgora.pl;
p.mroz@imei.uz.zgora.pl; a.mowna@issi.uz.zgora.pl

Abstract. Due to the need of automatically determine the semi-trailer's identifier, a real-time identification system based on the camera image was developed. The article presents two methods of recognizing objects in an image. Both methods were tested and one of them was selected for implementation. The system was made and correctness tests were carried out. The system works properly and is used in tractor models with semi-trailers at the Faculty of Computer Science, Electrical Engineering and Automation at the University of Zielona Góra.

Keywords. Image recognition, object detection, vehicle control, ArUco markers.

1 Introduction

For many years, the world has been working on the development of various types of vision systems. These systems are used, among others, for technological processes, monitoring of cities, air space, etc. One part of the process is to recognize objects in an image that can be performed on static image or from the moving camera. Face recognition, identification of license plates in cars, searching for cells on medical images are just a few of many examples of this rapidly growing branch of technology.

At the Faculty of Computer Science, Electrical Engineering and Automation of the University of Zielona Góra, a control system for the model of a truck with a semi-trailer made in 1:14 scale has been developed by students. The control is carried out using the manipulator shown in Figure 1. The manipulator allows you to control all the functions available in the saddle tractor model (speed, turn, selection of the gear, lighting, acoustic signal, etc.). Signals are sent to the truck (Figure 2). The control of the semi-trailer shown in Figure 3 is carried out from the manipulator via a truck tractor.

Currently, the department has four sets of vehicles (Figure 4) that can drive with any semi-trailer. All trucks and semi-trailers have the same addresses on the wireless bus, which means that the control is carried out with the currently turn on set. Ultimately, a modification of the manipulator program is planned, which will allow the operator to choose any truck. One of the problems that appeared during the implementation of this task is to indicate to the tractor the semi-trailer address to which he has driven and in which he controls the lighting, lifting the supports and read the battery voltage. In the described system, the identification should be carried out automatically, without the participation of the operator.

There was born an idea to identify automatically the semi-trailer that was connected to the truck, basing on the image from the camera placed on the back of the tractor cab (Figure 5).

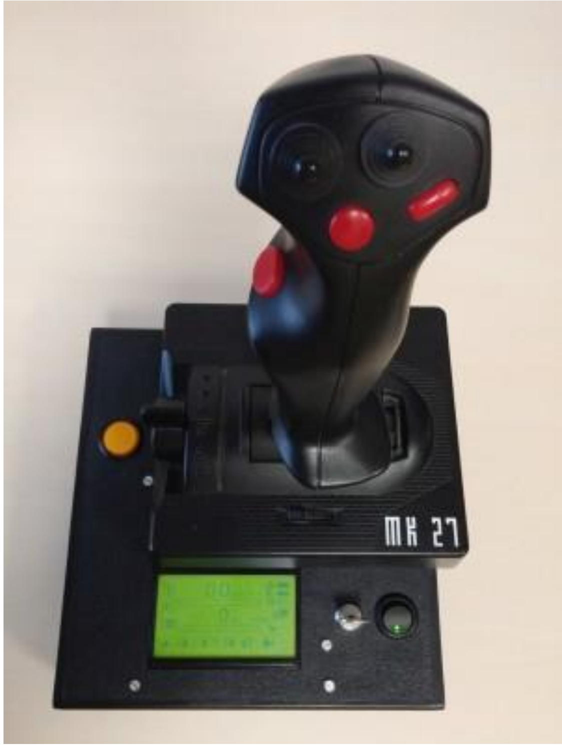


Figure 1. The view of the manipulator

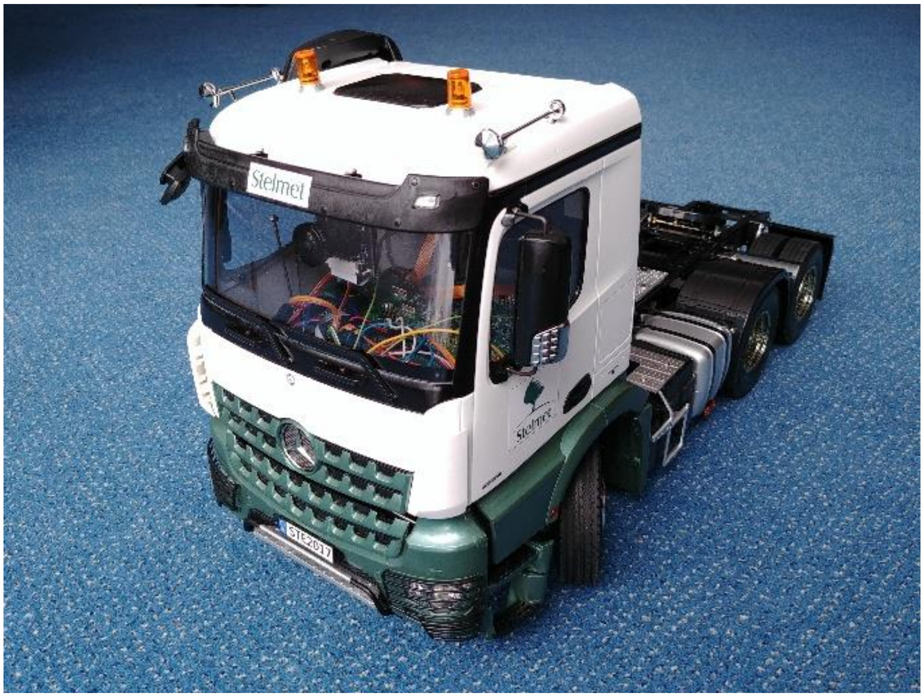


Figure 2. View of a tractor

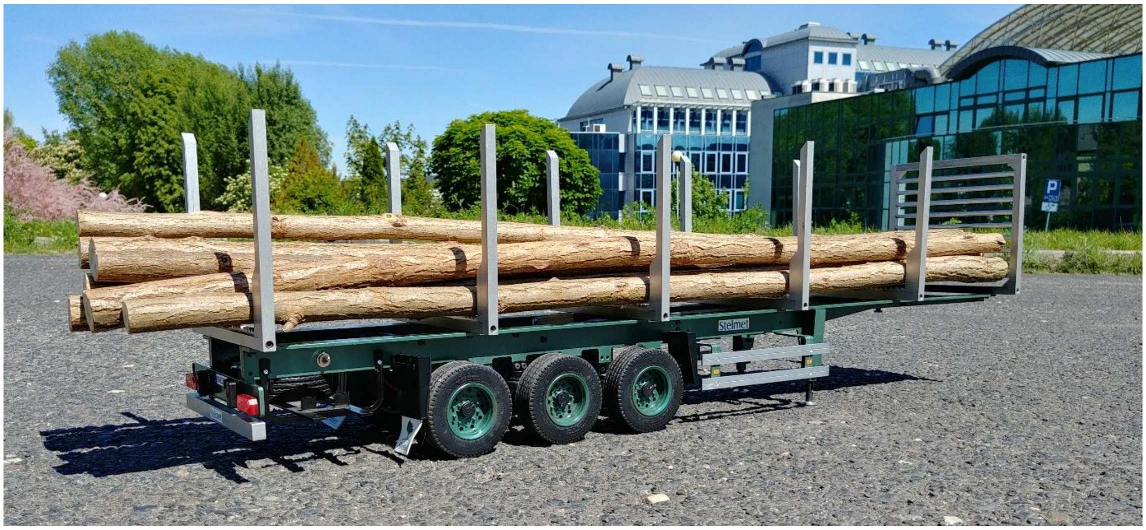


Figure 3. View of the semi-trailer

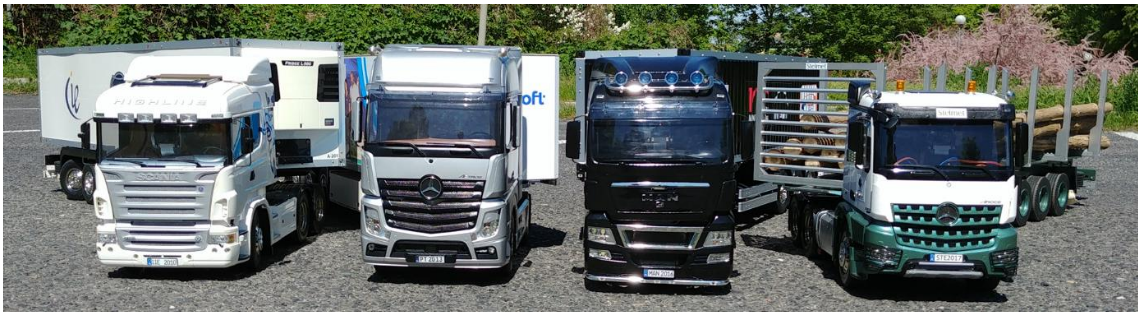


Figure 4. Tractor units with semi-trailers located
at the Faculty of Computer Science, Electrotechnics and Automation

2 Design assumptions

The tractors are equipped with two computers: STM32F4 Discovery - used to control the model, and Raspberry Pi 3 - used to acquire and transfer the image from one of the two cameras placed in the cabin of the vehicle. First camera - with wide angle - is placed at the front of the cabin and transmits an image seen from the front of the truck, the second one - ordinary - is placed in the back of the cabin and sends the image of the rear of the tractor and that what is behind it.

In order to build a system that identifies the kind of semi-trailer, it was assumed that:

- the system is operating on a Raspberry Pi computer built in the truck model,
- all operations used for the objects identification should be carried out without the participation of external data processing and analysis services,
- the data source with which the program will recognize the object is the image from the camera mounted in the rear of the truck cab (Figure 5),
- the system has to work correctly in a computer environment with limited hardware resources,
- it should be resistant to external conditions, eg. brightness of lighting,

- should identify the semi-trailer as soon as possible,
- beginning the recognition process takes place on a command sent via the UART interface from the truck control computer,
- the identification result has to be sent via the UART interface to the truck control computer.

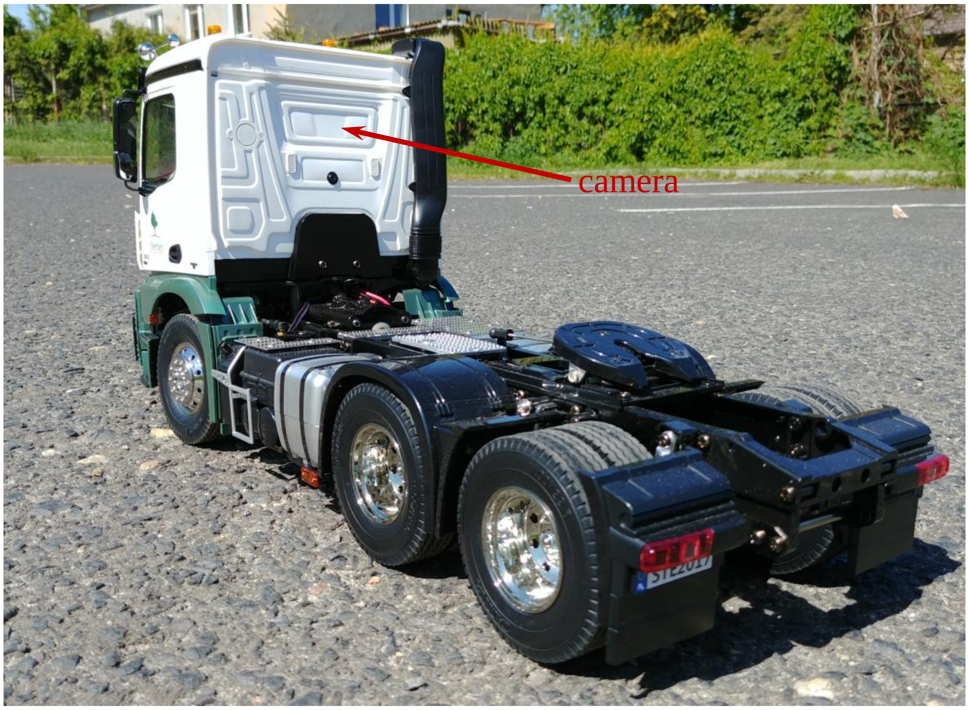


Figure 5. View of the camera for recognizing the semi-trailer

3 Used method of Image analysis.

3.1 Template matching method

The template matching method [3] is a model-based object recognition method. In this method it is necessary to create the so-called "repository of objects" (models base), which is used to compare the acquired image. The picture downloaded by the camera is compared with the pattern in the repository.

The object repository should contain all possible images that the recognition system will have to deal with. The reference images should be grouped according to the features of the object under study and then grouped into classes. During processing, firstly image classes are selected, and then specific pictures using matching method. This approach is aimed at reducing the number of comparisons [1, 2].

The main disadvantage is a huge number of patterns contained in the model database, what is necessary to use presented method. It leds to significant extention of the recognition time of the object.

For the needs of the implemented project, a simple database of reference images containing photos of semi-trailers was created. Exemplary reference images of semi-trailers are presented in Fig. 6.

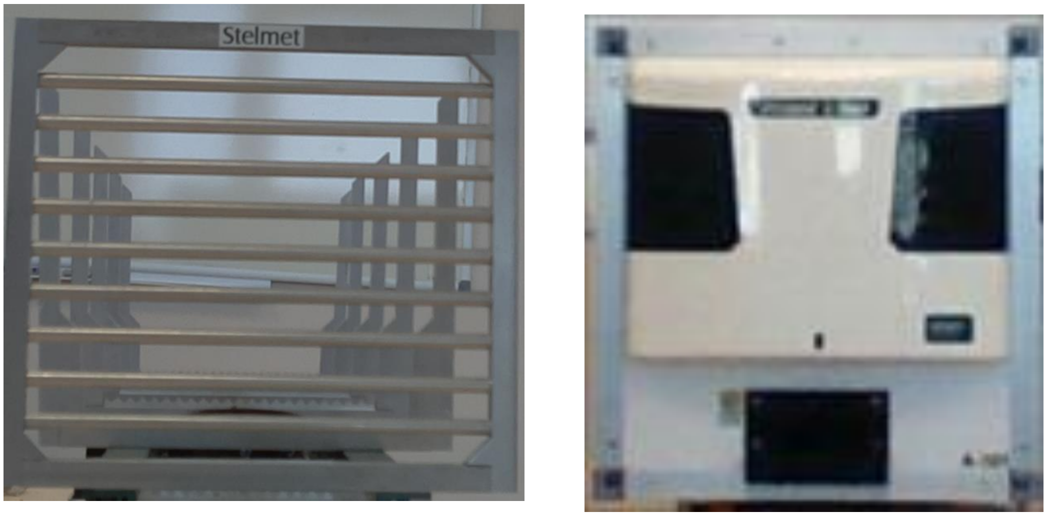


Figure 6. Exemplary reference images of semi-trailers

The program dedicated for Raspberry PI realizing the recognition of semi-trailers has been written in Python. The OpenCV (Open Source Computer Vision) library was used to process the image. It offers a lot of useful functions, which is using during image processing [5]. According to its authors, OpenCV has over 2,500 optimized algorithms.

During the tests, a program is run from the command line and the user can enter two parameters. The first one is responsible for accuracy and is given as a number between 0 and 1, for example the number 0.65 equals the accuracy of 65%. The accuracy is increasing with increase of value of this parameter. With an accuracy of more than 80-90%, for the program it is much harder to recognize the semi-trailer, while in the case of a set value below 50%, the program can recognize semi-trailers in other objects. The value from the range from 0.65 to 0.7 is assumed to be the optimal value, which allows to recognize the semi-trailer in an effective way. The second value entered by user determines the maximum timeout to recognize the semi-trailer. Exceeding this time interrupts the recognition and returns the "timeout" value.

The program can be run in two variants: the first one enables to display of the recognized model of the semi-trailer on the screen (Figure 7); the second variant allows to obtain the text identifier of the semi-trailer without displaying model on a screen (Figure 8). Research have shown that the time of recognition of a semi-trailer is from 2. This time depends of many factors, such as the brightness of the lighting, the angle between the tractor and the semi-trailer or the distance of the semi-trailer from the tractor.

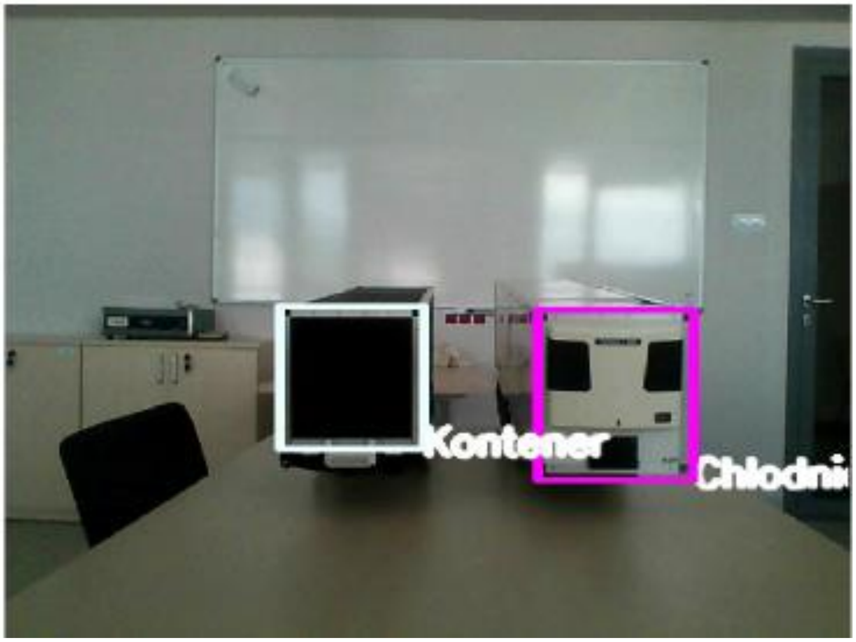
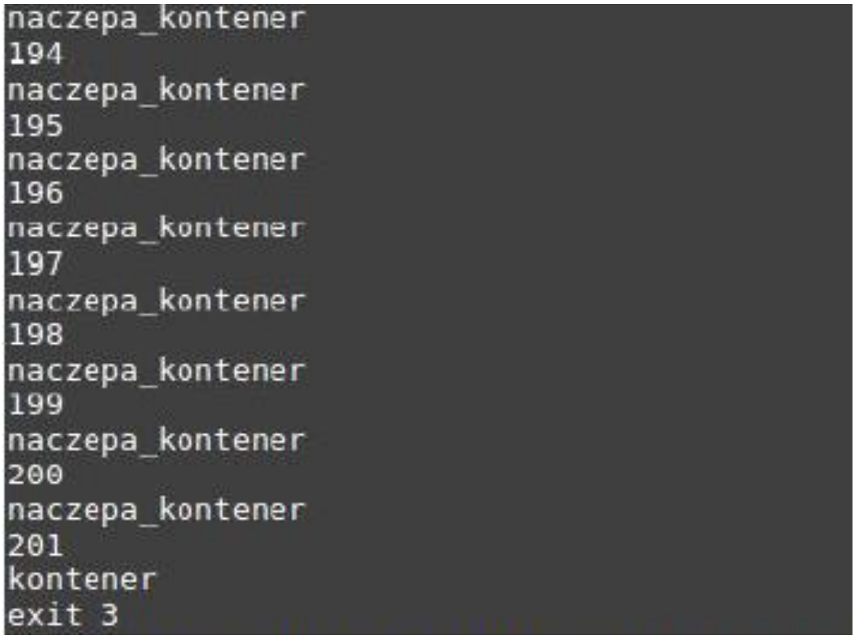


Figure 7. Visual presentation of recognized semi-trailers



```
naczepa_kontener
194
naczepa_kontener
195
naczepa_kontener
196
naczepa_kontener
197
naczepa_kontener
198
naczepa_kontener
199
naczepa_kontener
200
naczepa_kontener
201
kontener
exit 3
```

Figure 8. Textual identifier of the semi-trailer

3.2 ArUco markers

The ArUco marker (Figure 9) is a square tag composed of a wide black border and an internal binary matrix that specifies its identifier (id). The black border allows for quick image detection, and the binary codification enables its identification and lets to apply of error detection and correction techniques. The size of a marker determines the size of the inner matrix. For example, a 4x4 tag consists of 16 bits.

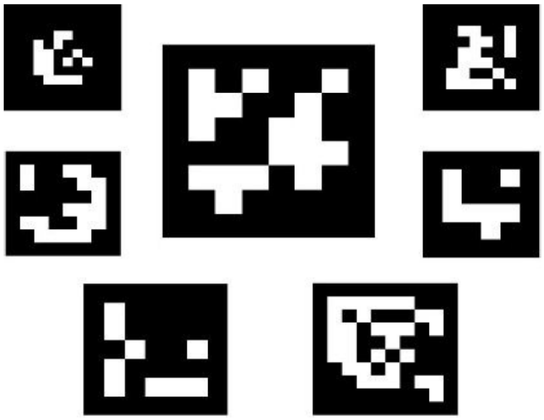


Figure 9. Examples of ArUco markers [7]

There are several types of markers, and each of them belongs to the dictionary. The design of the dictionary is important because its markers should be as different as it is possible to avoid misunderstandings.

The ArUco OpenSource library written in C ++ is used to detect ArUco markers on images [6]. Furthermore, if the camera is calibrated, you can estimate the camera's orientation relative to the markers. The result of identification of the semi-trailer by means of an exemplary mark is presented in Figure 10.

Application of the ArUco library in the program enabled real-time detection of marks placed on semi-trailers. In addition, drive of tractor to the semi-trailer can be made at different angles and from different distances. Studies have shown that the time of the semi-trailer recognizing ranges between 1 s to 5 s and it independent of the brightness and angle between the tractor and semi-trailer.

4 System construction and testing

Based on the research described in p. 3, it was decided that the system being developed will use a program that uses ArUco marks. The system operation algorithm is shown in Fig. 11. It was implemented in Python using the OpenCV library and uploaded to the Raspberry Pi computer. In the next step implemented in the STM32F4 Discovery processor the truck control program was modified. After starting the Raspberry Pi processor, a system that waits for the order to start the recognition from the truck control processor sent via the UART interface is started. This order is sending when the semi-trailer is placed in the tractor's saddle. During the recognition of the semi-trailer, a counter controlling conversion time is also started. The process ends when the semi-trailer is recognized or after exceeding the prescribed time. Depending on the reason of ending the recognition by the UART interface, there is sent the code of the recognized semi-trailer or the 0xffff code, which informs that semi-trailer has not been recognized. To the STM32F4 Discovery processor was also added support for this command. When the ID of the detected semi-trailer is obtained, the address of the semi-trailer in the tractor is corrected and it is possible to control the created set. From now on, the system waits again for an order to start another conversion.

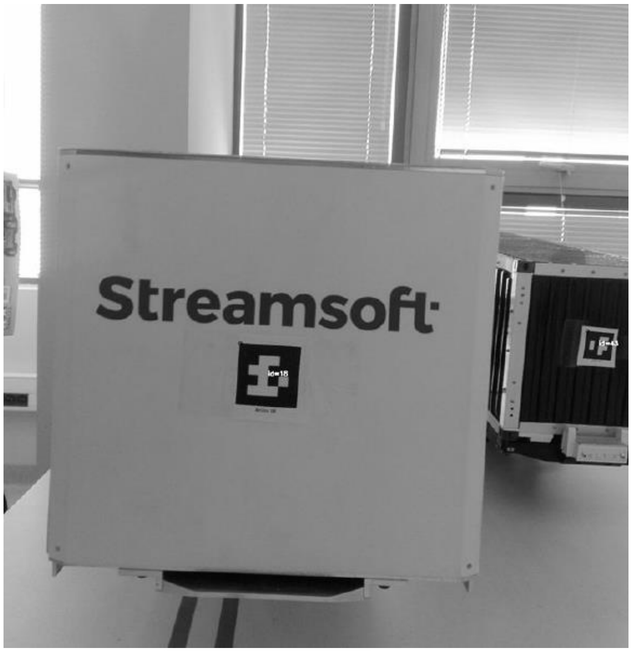


Figure 10. Identification of the semi-trailer with the use of a marker

5 Conclusion

Presented in the article methods of recognizing objects enabled the recognition of trucks semi-trailers. In the case of the pattern matching method, due to the limited capacity of Rasphberry Pi memory, it may be problematic to create a sufficiently large database of patterns. Also the time of recognition of the object may be significantly extended due to the large number of patterns. Tests have shown that this method works the best when the tractor approaches the semi-trailer in line as near as possible to the straight line. Too much deviation can lead to anomalies in the objects recognition, eg. recognition of a semi-trailer in an object that is not. Furthermore, the brightness of the semi-trailer lighting has a very big impact on the recognition time.

The use of ArUco tags allowed to reduce the number of patterns, which significantly shortened the time of identification. In addition, it has been reduced influence of the lighting brightness. The developed system has been implemented and is used in truck models owned by the Faculty of Computer Science, Electrotechnics and Automation at the University of Zielona Góra.

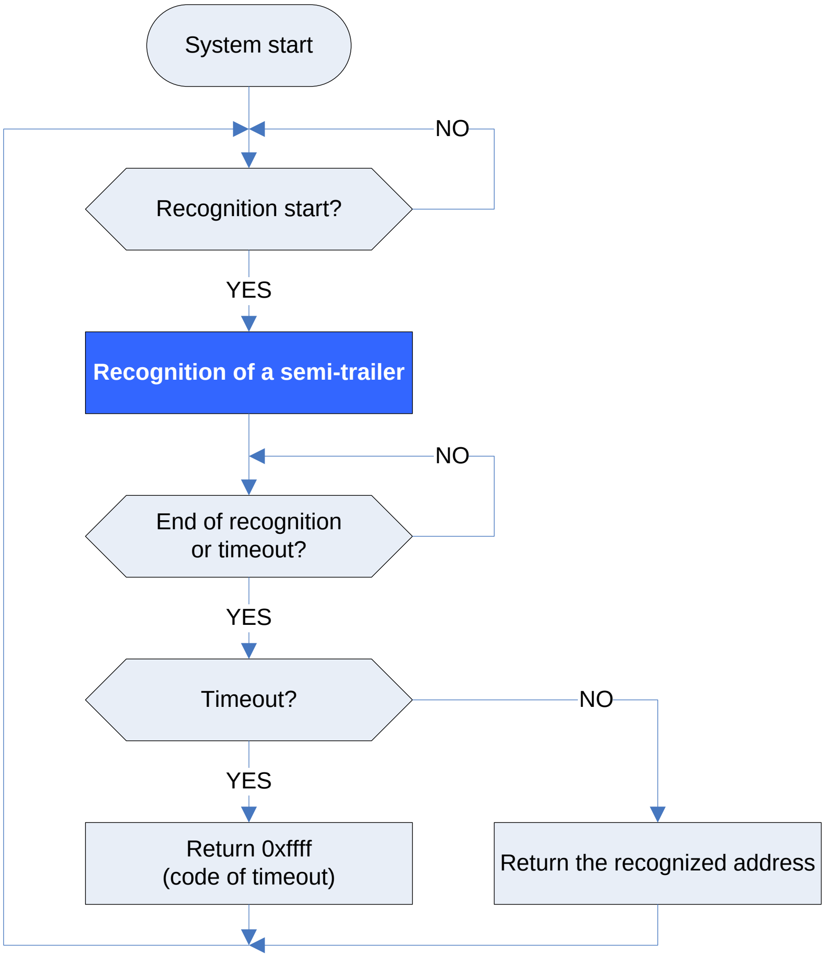

System start
NO
Recognition start?
YES
Recognition of a semi-trailer
NO
End of recognition or timeout?
YES
Timeout?
NO
YES
Return 0xffff (code of timeout)
Return the recognized address

Figure 11. System operation algorithm

References

[1] Szymczyk T., Metoda dopasowania wzorców w rozpoznawaniu obrazów – ograniczenia, problemy i modyfikacje metody, Automatyka tom 12, zeszyt 2, UWND 2008, str. 449-462.

[2] Choraś R. S.: Komputerowa Wizja Metody interpretacji i identyfikacji obiektów, EXIT Warszawa, 2005

[3] T. Mahalakshmi, R. Muthaiah, P. Swaminathan, Review article: an overview of template matching technique in image processing, Res J Appl Sci, Eng Technol, 2012

[4] Mai L. C.: Introduction to computer vision and image processing, Department of Pattern Recognition and Knowledge Engineering Institute of Information Technology, Hanoi, Vietnam eeexplore.ieee.org/iel5/10500/33260/01572091.pdf?arnumber=1572091.

[5] OpenCV: http://sourceforge.net/projects/opencv/

[6] ArUco: a minimal library for Augmented Reality applications based on OpenCV; https://www.uco.es/investiga/grupos/ava/node/26

[7] Detection of ArUco Markers:

[8] https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html

Multidimensional Pareto Frontiers Intersection: Processor Optimization Case Study

Jakub Podivinsky, Ondrej Cekan, Martin Krcma, Radek Burget,
Tomas Hruska, Zdenek Kotasek
Brno University of Technology, Faculty of Information Technology,
Centre of Excellence IT4Innovations
Božetěchova 2, 612 66 Brno, Czech Republic

{ipodivinsky,icekan,ikrcma,burgetr,hruska,kotasek}
fit.vutbr.cz

Keywords. Pareto optimization, Pareto frontier, processor optimization, ASIP.

Abstract

Almost all today's electronic devices are equipped with a processor. Different applications require and depend on different properties of the processor. For example, the fast-growing field of Internet of Things depends on a long operation time of the devices when powered with batteries. Using general purpose processors has proved ineffective which led to a growing usage of Application-Specific Instruction-Set processors (ASIPs) which can be optimized for specific applications using different modifications of their properties (such as the number of registers, cache sizes, instruction set modifications, etc.).

A suitable processor configuration can be hand-picked by a designer or by an automatic tool. Such a tool was developed in our previous research. It is able to find a set of Pareto-optimal processor configurations for a specific application which can be a significant help in a device design. The cost of the design process can be cut significantly when a processor is used in multiple designs. The goal of this paper is to introduce a tool able to find a suitable processor configuration for multiple applications by constructing a compromise Pareto-optimal frontier of processor configurations. The paper describes this problem on a theoretical level and it also introduces a practical implementation and experimental evaluation of constructing a compromise Pareto frontier of processor configurations for a set of applications. The experiments are based on a parametrizable RISC-V processor and example of compromise Pareto-optimal frontier is shown in Fig. 1.

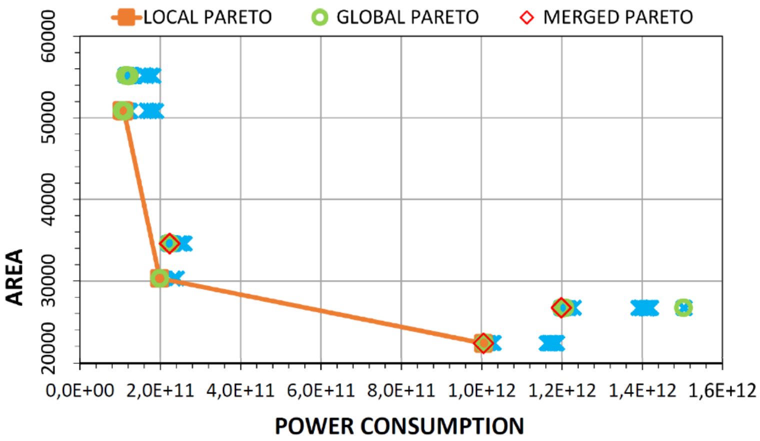


Fig. 1: An example of all configurations (blue marks) with the original local and global Pareto frontiers together with merged Pareto frontier.

Paper origin

The original paper has been accepted at 22nd Euromicro Conference on Digital System Design in Kallithea, Chalkidiki, Greece [1].

Acknowledgment

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602.

References

[1] J. Podivinsky, O. Cekan, M. Krcma, R. Burget, T. Hruska and Z. Kotasek. Multidimensional Pareto Frontiers Intersection Determination and Processor Optimization Case Study. In: 2019 Euromicro Conference on Digital System Design. Kallithea: IEEE Computer Society, 2019, accepted for publishing.

Future Directions in Network Flow Monitoring

Petr Velan
Masaryk University, Institute of Computer Science
Botanická 68a, 60200, Brno

velan@ics.muni.cz

Keywords. Network, Flow Monitoring.

Abstract

Flow monitoring has been used for accounting and security for more than two decades. This paper describes how it was developed, what is its current status and what challenges can be expected in this field in the following years.

1.1 The Past

The first mention of a flow export can be found in RFC 1272 [1] published in 1991 by IETF Internet Accounting (IA) Working Group (WG). The goal of the document was to provide background information on Internet accounting. The authors describe methods of metering and reporting network utilisation. The goal at the time was to provide a framework for traffic accounting. However, the common belief was that internet should be free and any form of traffic capture, even for the accounting purposes, is undesirable. This, together with the lack of vendor interest, resulted in the conclusion of the working group in 1993. Note that the negative attitude towards the monitoring returns more than 20 years later [2].

In 1995, Claffy, Braun, and Polyzos showed a methodology for internet traffic flow profiling based on packet aggregation [3], which started a revival of flow monitoring efforts. The Realtime Traffic Flow Measurement (RTFM) Working Group was active since 1996 and was conclude in 2000 by publishing several RFCs describing new traffic flow measurement framework with increased flexibility and even provided bi-directional flow support [4]. Since these documents fulfilled the objectives of the RTFM WG, the group was concluded in 2000. However, no flow export standard was developed as the vendors showed no interest in this area.

Meanwhile, Cisco realised that similar kind of flow information is already stored in a flow cache of their packet switching devices. The purpose of this cache is to speed up packet switching by making a forwarding decision only for the first packet of each flow. Unlike the RTFM flow measurement framework, the primary purpose of flow cache is not accounting nor monitoring. Therefore the configuration of measurement process using a flow cache in a switch is severely limited. Despite the limitations, once Cisco introduced its flow export technology called NetFlow, it achieved widespread adoption. The main reason for the wide adoption was the fact that it was readily available on most Cisco devices with little effort. The NetFlow was patented in 1996 and the first version that became available to the general public around 2002 was NetFlow v5 [5], albeit Cisco newer released any official specification. The NetFlow v5 format simply specified a single set of fields that should be exported from each flow record.

NetFlow v5 was soon obsoleted by NetFlow v9 which remedied some of the deficiencies of the previous version. The state of NetFlow v9 is described in [6]. It allowed defining an arbitrary set of fields for export using templates. It also introduced support for new protocols, such as IPv6, Virtual Local Area Networks (VLAN), Multiprotocol Label Switching (MPLS), Border Gateway Protocol (BGP) or Multicast.

Other vendors created their own versions of flow exporting protocols, although they retained some level of compatibility with NetFlow. There are JFlow by Juniper, CFlow by Alcatel-Lucent, RFlow by Ericsson, and other protocols. When the potential of flow monitoring for security purposes became realised in 2005 [7], more effort was devoted to extending flow records with information not directly associated with switching. Cisco presented Flexible NetFlow technology [8] in 2006 which allows to dynamically define and export new types of information, such as parts of payloads or traffic identification.

In 2001, it was clear that exporting flow information from switching devices was going to be supported by vendors. However, no standard flow export protocol existed at the time and NetFlow v5 was not yet released to general public. For that reason the IETF started IP Flow Information Export (IPFIX) WG [9]. The charter was updated over the years to match current requirements. Several vendors were engaged in the IPFIX WG's activities, most notably Cisco, which significantly contributed from the start. The WG defined a set of requirements for the IPFIX protocol [11] and evaluated existing candidate protocols [12] to decide the most suitable approach to defining the new protocol. The NetFlow v9 specification (RFC 3954) was designed with IPFIX requirements in mind [13] and was released in order to compete in this evaluation (RFC 3955). After the evaluation, the NetFlow v9 was chosen as a basis of the new IPFIX protocol. For this reason, IPFIX is sometimes called NetFlow v10 and even starts with protocol version 10 in its header. However, the IPFIX protocol supports many new features and is not completely backwards compatible with NetFlow.

The IPFIX WG did more than just design the IPFIX protocol. In the 29 RFCs published before its conclusion, the WG paid attention to, for example: bidirectional flow export, architecture for IP flow information export, reducing redundancy in flow and IP flow mediation framework. The IPFIX protocol specification is described by *"Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information"* [24] which became an Internet Standard. The working group was concluded in 2014, however, IPFIX related Internet-Drafts are still being created by involved parties. Further information about IPFIX development is provided by Brownlee in [25].

The importance of flow monitoring for security purposes was recognized by Cisco engineers in 2005 who proposed to use NetFlow for anomaly detection and traffic analysis [7]. Creation of dedicated flow monitoring probes allowed to easily extend the set of collected flow features and add application information to the flows. Pioneers in this area were Cisco, ntop, Masaryk Unviersity, and CESNET. Applications such as HTTP, DNS, and SMTP were being analysed. Cisco published a tool called *joy* [**Cisco–Joy**] in 2016 which allows to collect a rich set of information about network connections.

1.2 The Now

A concern for privacy of users has been rising in recent years, which led to an extensive deployment of encryption of network traffic. It is more and more difficult to monitor network applications as most traffic is protected by TLS of other encryption protocols. HTTP/2 is supported only together with encryption by mainstream browsers. A recent push for addition of WireGuard VPN to Linux kernel has triggered its increasing adoption. However, despite the use of encryption, the need to monitor the traffic has not decreased. The challenge that we are facing is monitoring analysis of encrypted traffic.

Fortunately, machine learning algorithms are increasingly available as well; therefore statistical analysis of encrypted data can be performed with relative ease. There is a large body of research encrypted traffic classification and malware detection in encrypted traffic. The most recent results from Cisco show that information from TLS protocol together with per packet metrics can be used to achieve high accuracy in malware detection. However, flow records need to be extended with additional information to provide enough features for the machine learning algorithms.

1.3 The Future

The level of encryption can be only expected to grow. There is an RFC draft called *Encrypted Server Name Indication for TLS 1.3* which proposes to encrypt even Server Name Indication in TLS protocol. Combined with increasing deployment of DNS over TLS and DNS over HTTPS protocols, most of the current visibility into network traffic will soon be lost. This will result in higher demand for statistical analysis of network traffic.

To obtain accurate results for encrypted traffic classification, an annotated dataset of high quality is needed. There are two approaches to obtain such datasets. The first is to observe and capture normal network traffic and manually or semi-automatically annotate it. The second approach is to generate the traffic manually and label the observed traffic based on the known traffic patterns. However, both approaches are time-consuming and error prone. Moreover, such datasets become obsolete in time and might not contain the necessary traffic mix that is seen in real networks. Therefore, the most of the research should be focused on generating and obtaining datasets that will enable us to perform encrypted traffic classification with high accuracy.

A promising way to obtain such datasets is to combine information from multiple sources, such as DNS resolvers, server logs, and application logs. This will allow us to assign labels to flow data with high accuracy and create datasets that are both real and of high quality. Once the data sets are available, machine learning can be used to find correlations and relations in the data, which can be used to analyse even non-labelled traffic. However, masquerading network traffic as a different category is just a next step that attackers are likely to be examining.

Apart from the encrypted traffic classification, there is also the question of quality of the generated data. For example, will the machine learning methods work well if flow generation parameters, such as timeouts, are changed? How are the flow exporters behaving under heavy load, are the exported flows incomplete? These and similar questions need to be answered, especially when machine learning is relied upon.

Paper origin

The first part of this paper has been accepted as a dissertation of the author.

Acknowledgment

This research was supported by ERDF ”CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence” (No. CZ.02.1.01/0.0/0.0/16 019/0000822).

References

[1] C. Mills, D. Hirsh, and G.R. Ruth. *Internet Accounting: Background*. RFC 1272 (Informational). RFC. Fremont, CA, USA: RFC Editor, November 1991. URL: https://www.rfc-editor.org/rfc/rfc1272.txt (page 1).

[2] S. Farrell and H. Tschofenig. *Pervasive Monitoring Is an Attack*. RFC 7258 (Best Current Practice). RFC. Fremont, CA, USA: RFC Editor, May 2014. URL: https://www.rfc-editor.org/rfc/rfc7258.txt (page 1).

[3] Kimberly C. Claffy, Hans-Werner Braun, and George C. Polyzos. “A Parameterizable Methodology for Internet Traffic Flow Profiling”. In: *IEEE Journal on Selected Areas in Communications* 13.8 (October 1995), pp. 1481–1494. ISSN: 0733-8716. DOI: 10.1109/49.464717 (page 1).

[4] N. Brownlee, C. Mills, and G. Ruth. *Traffic Flow Measurement: Architecture*. RFC 2722 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 1999. URL: https://www.rfc-editor.org/rfc/rfc2722.txt (page 1).

[5] Cisco Systems, Inc., San Jose, CA and USA. *NetFlow Services Solutions Guide*. January 2007. URL: http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.html (Accessed on April 27, 2017) (page 2).

[6] B. Claise. *Cisco Systems NetFlow Services Export Version 9*. RFC 3954 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3954.txt (page 2).

[7] Cisco Systems, Inc., San Jose, CA and USA. *Cisco IOS NetFlow and Security*. February 2005. URL: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6642/prod_presentation0900aecd80311f49.pdf (Accessed on April 27, 2017) (page 2).

[8] Cisco Systems, Inc., San Jose, CA and USA. *Cisco IOS Flexible NetFlow*. December 2008. URL: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/flexible-netflow/product_data_sheet0900aecd804b590b.html (Accessed on April 27, 2017) (page 2).

[9] The Internet Engineering Steering Group. *IP Flow Information Export (ipfix) Charter*. URL: http://datatracker.ietf.org/wg/ipfix/charter/ (Accessed on April 27, 2017) (page 2).

[10] The Internet Engineering Steering Group. *IP Flow Information Export Charter*. September 2001. URL: https://www.ietf.org/mail-archive/web/ipfix/current/msg00213.html (Accessed on April 27, 2017) (page 2).

[11] J. Quittek et al. *Requirements for IP Flow Information Export (IPFIX)*. RFC 3917 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3917.txt (page 2).

[12] S. Leinen. *Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX)*. RFC 3955 (Informational). RFC. Fremont, CA, USA: RFC Editor, October 2004. URL: https://www.rfc-editor.org/rfc/rfc3955.txt (page 2).

[13] Brian Trammell and Elisa Boschi. "An Introduction to IP Flow Information Export (IPFIX)". In: *IEEE Communications Magazine* 49.4 (April 2011), pp. 89–95. ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5741152 (page 2).

[14] B. Trammell and E. Boschi. *Bidirectional Flow Export Using IP Flow Information Export (IPFIX)*. RFC 5103 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, January 2008. URL: https://www.rfc-editor.org/rfc/rfc5103.txt (page 2).

[15] G. Sadasivan et al. *Architecture for IP Flow Information Export*. RFC 5470 (Informational). RFC. Updated by RFC 6183. Fremont, CA, USA: RFC Editor, March 2009. URL: https://www.rfc-editor.org/rfc/rfc5470.txt (page 2).

[16] E. Boschi, L. Mark, and B. Claise. *Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports*. RFC 5473 (Informational). RFC. Fremont, CA, USA: RFC Editor, March 2009. URL: https://www.rfc-editor.org/rfc/rfc5473.txt (page 2).

[17] T. Dietz et al. *Definitions of Managed Objects for IP Flow Information Export*. RFC 5815 (Proposed Standard). RFC. Obsoleted by RFC 6615. Fremont, CA, USA: RFC Editor, April 2010. URL: https://www.rfc-editor.org/rfc/rfc5815.txt (page 2).

[18] T. Dietz et al. *Definitions of Managed Objects for IP Flow Information Export*. RFC 6615 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, June 2012. URL: `https://www.rfc-editor.org/rfc/rfc6615.txt` (page 2).

[19] P. Aitken et al. *Exporting MIB Variables Using the IP Flow Information Export (IPFIX) Protocol*. RFC 8038 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, May 2017. URL: `https://www.rfc-editor.org/rfc/rfc8038.txt` (page 2).

[20] A. Kobayashi and B. Claise. *IP Flow Information Export (IPFIX) Mediation: Problem Statement*. RFC 5982 (Informational). RFC. Fremont, CA, USA: RFC Editor, August 2010. URL: `https://www.rfc-editor.org/rfc/rfc5982.txt` (page 2).

[21] A. Kobayashi et al. *IP Flow Information Export (IPFIX) Mediation: Framework*. RFC 6183 (Informational). RFC. Fremont, CA, USA: RFC Editor, April 2011. URL: `https://www.rfc-editor.org/rfc/rfc6183.txt` (page 2).

[22] E. Boschi and B. Trammell. *IP Flow Anonymization Support*. RFC 6235 (Experimental). RFC. Fremont, CA, USA: RFC Editor, May 2011. URL: `https://www.rfc-editor.org/rfc/rfc6235.txt` (page 2).

[23] G. Muenz, B. Claise, and P. Aitken. *Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols*. RFC 6728 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, October 2012. URL: `https://www.rfc-editor.org/rfc/rfc6728.txt` (page 3).

[24] B. Claise, B. Trammell, and P. Aitken. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*. RFC 7011 (Internet Standard). RFC. Fremont, CA, USA: RFC Editor, September 2013. URL: `https://www.rfc-editor.org/rfc/rfc7011.txt` (page 3).

[25] Nevil Brownlee. “Flow-Based Measurement: IPFIX Development and Deployment”. In: *IEICE Transactions on Communications* E94.B.8 (September 2011), pp. 2190–2198. DOI: `10.1587/transcom.E94.B.2190` (page 3).

Future approaches to monitoring in high-speed backbone networks

Karel Hynek, Tomáš Beneš, Tomáš Čejka, Hana Kubatová
FIT Czech Technical University in Prague   CESNET a.l.e.
Thákurova 9, Prague 6   Zikova 4, Prague 6

hynekkar@fit.cvut.cz, benesto3@fit.cvut.cz,
tomas.cejka@fit.cvut.cz, kubatova@fit.cvut.cz

Keywords. Network, Monitoring, Analysis, Encryption, Security, Hardware

Abstract

Network monitoring features has been always a challenge in high-speed networks. Some of them like detailed traffic analysis and packet inspection are not suited or simply not feasible even on modern hardware. The challenges are becoming even greater with an uprise of encrypted traffic. This leaves large opportunity for threat actors to take advantage of. Therefore, it is necessary to develop a new generation of monitoring tools that can deal with the current issues for security purposes. This research aims to improve traffic analysis techniques to handle encrypted traffic, and also to adapt hardware accelerated monitoring components for processing.

2 Introduction

Personal privacy became one of the crucial features of modern applications in recent years. Due to surveillance, fraudulent attempts and data leaks scandals companies are forced by the public opinion to strengthen their services by use of encryption.

Gartner predicts (1) that by the end of 2019, 80 percent of internet traffic will be encrypted. There is simply no reason NOT to encrypt anymore. This prevents any surveillance possible and allows users to have their privacy secured. However, this enables threat actors to hide malicious activities on the network.

This introduces the new field of research, where a traditional threat inspection with bulk decryption, analysis and re-encryption is not always practical or feasible, for performance and resource reasons.

Detection of malicious and non-standard activities in encrypted communication is one of the new challenges. Most of the businesses do not have tools capable of analyzing these threats in encrypted connections at their disposal. This leaves these businesses vulnerable to many threats.

There are several proprietary solutions targeted on detecting anomalies in encrypted connections (1). These solutions aims to be deployed at large businesses and are not intended to be used in the open source form. This gives us a great opportunity to introduce new types of threat detection algorithms aimed at encrypted communication.

3 Monitoring infrastructure for future

Traditional infrastructure for network monitoring is composed of monitoring probes, also known as observation points, that aggregate packet-level information and extract protocol headers. These information are sent in a form of IP flow records into a flow collector, where the data are processed and stored.

Flow data usually contain information about transport layer protocols, i.e., IP addresses, transport layer protocol, and transport protocol ports. Additional information can be extracted from the application layer in case of not encrypted traffic. Recently, there have been developed tools (2) for this purpose. However, the amount of encrypted traffic is increasing, where the application layer is not available.

Evolution in encrypted traffic analysis starts with Joy flow exporter (3), that is designed to extract data features, i.e., inter packet gaps, the sequence of lengths and arrival times of TLS records etc., from network traffic. The computed features can be used with machine learning tools for classification and anomaly detection. However, this approach is very dependent on training datasets that are generally not easily available.

Monitoring probes for high-speed network links need to be hardware accelerated. Development of such high-performance devices is a non-trivial task. The devices must be intensively optimized for specific purposes, such as header field extraction or computation of traffic-related statistics. On the other hand, there is a software-defined networking concept that describes the functionality with more abstraction using either configuration mechanisms or even application specific high-level languages. Benáček et al. presented a generation of high-speed (at least 100 G) devices from the P4 (4) language in (5). Also, Havranek et al. proposed enhanced flow exporter described in P4 capable of processing multi-layer encapsulation in (6).

The aim of our research is to improve the flow-based network monitoring concept for encrypted traffic. It will based upon the current state-of-the-art, which is represented by the latest Cisco systems activities related to botnet detection in encrypted traffic using machine learning. The theoretical part of the work lies in traffic analysis and derivation of models that can be used for the recognition and classification of network traffic. Theoretical findings, evaluated by experiments with real network traffic, will be applied to design and develop a new generation of hardware-accelerated monitoring probes that can provide required traffic features for classification and detection models. The main goal of the hardware scope of the work aims to explore promising technology of high-level synthesis using P4 language to describe a hardware-accelerated monitoring probe. However, the synthesis from P4 language into sufficient high-throughput hardware design (e.g., in VHDL) is not currently not available and must be elaborated.

4 Conclusion

This paper presents a first insight into our planned research related to a new generation of flow-based network monitoring system consisting of hardware-accelerated monitoring probes and classification/detection software capable of analysis of encrypted traffic in high-speed backbone networks.

The planned solution will be a continuation of the current research activities at Faculty of Information Technology in collaboration with CESNET association, which is the operator of the Czech national research and education network.

Acknowledgment

This research has been supported by the CTU grant projects SGS17/017/OHK3/1T/18 ”Dependable and attack-resistant architectures for programmable devices” and SGS17/212/OHK3/3T/18 ”Research and development of tools for security analysis of computer network traffic”

References

[1] Encrypted Traffic Analytics, Cisco Systems 2019. [Online]. Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en.pdf

[2] Čejka, T.: Stream-wise Parallel Anomaly Detection in Computer Networks, Ph.D. thesis, Faculty of information technology, CTU in Prague, 2018.

[3] Anderson, B., McGrew, D.: Joy. 2016. [Online]. Available: https://github.com/davidmcgrew/joy

[4] P4 language, https://p4.org/

[5] Benáček, P., Puš, V., Kubátová, H., Čejka, T.: P4-to-VHDL: Automatic Generation of High-Speed Input and Output Network Blocks. In: Microprocessors and Microsystems journal (MICPRO, Volume: 56), DOI: 10.1016/j.micpro.2017.10.012., Elsevier, 2018.

[6] Havránek, J., Velan, P., Cejka, T., Benáček, P.: Enhanced Flow Monitoring with P4 Generated Flexible Packet Parser. In proceedings AIMS 2018, 2018.

L7 capable flow exporter described in P4

Jiří Havránek   Tomáš Čejka   Pavel Benáček
FIT CTU in Prague   CESNET, a.l.e.
Thákurova 9, 160 00 Prague 6, Czech Republic

{havraji6,tomas.cejka}@fit.cvut.cz   benacek@cesnet.cz

Keywords. P4, compiler, backend, code generation, flow exporter, application protocols

Abstract

Current flow exporters are the essential source of information for monitoring systems. They usually create aggregated information as flow data and, additionally, it is possible to extract headers from higher layer protocols (L7). Due to requirements on high throughput, the flow exporters use hardware acceleration to handle high packet rate at link speed (aiming at least 100 Gb/s). However, manually created design of such high-performance devices is very complex and complicated. Therefore, we propose to use a high-level P4 language for description of network traffic processing device that will be capable of handling L7 information. As our recent works show, it is possible to generate high-performance firmware design automatically based on P4 description. Since P4 is not primarily intended for processing L7 data, this paper proposes a feasible way to overcome limits of P4.

2 Introduction

As computer networks grows rapidly, the infrastructure needs technology that is more flexible. In the recent history a software defined networking concept was introduced by Open Networking Foundation. It is based on flexible configuration of network active devices at run-time by a special element — controller — that has a global overview about network traffic.

The next evolution step brought more flexibility to network devices itself. P4 consortium has defined an universal architecture of network active devices (L2–L4 switches) that is composed of a parser, match action tables, and deparser. To describe functionality of all three components, a novel high-level application specific language called P4 was proposed. Based on P4 description, which is much simpler than low-level programming languages (such as C/C++) or hardware definition languages (such as VHDL). Naturally, the primary goal of P4 is to translate the description into real compiled or synthesized functional blocks usable in the device.

P4 consortium develops and maintains a compiler that is able to generate code for target platform using so called backend, i.e., part of the compiler. Section 3 describes several existing backends. However, we see the limits of P4 that are related to the capabilities of the language and standardized architecture. Our aim is to enhance possible use-cases of P4 to be able to process L7 protocols. As a proof-of-concept, we have created a P4 description of flow exporter, a system that aggregates packet-level information into flow data in IPFIX [1], and is able to extract even L7 information. The description is compilable using the developed backend for P4 compiler.

3 Related Work

There are few existing compiler backends for various P4 applications at the moment. The example of such backends can be the compiler from P4 to HDL language which is used for the description of digital circutis. This can be beneficially used for fast development of network accelerators because HDL languge is much more complicated compared to the P4 or C language. One of the emerging works from this area was described in [5]. There are also backends for different languages like transformation from P4 to C. However, none of currently known backends deals with automatic generation of flow exporter, but there are ones that can be used to generate flow exporter parts.

One of the existing backends is Flexible Packet Parser (FPP) backend [2] that was developed as our recent work. It deals with generation of packet parser described in P4 and generates C code with parsing function. This generated parser is capable of parsing unlimited number of protocol headers and tunnel encapsulation. The output of the parser are extracted headers in form of linked list. Headers to extract can be specified in special P4, architecture defined, structure.

Other backends with parser include generation of eBPF [3] and XDP [4] filters. Compiler generates C code of filters that can be compiled into filter program and loaded into kernel.

4 L7 processing

Our research is a continuation of the development L7 capable flow exporter that was written manually. Based on the analysis and our experiences, we have identified the following requirements to be able to process L7 headers:

1. Extract string with moving payload cursor
2. Match string in payload without moving payload cursor
3. Match extracted string
4. Convert extracted string to number
5. Copy strings between variables

All the listed function must be supported by the device in order to extract the needed information that should be put into IPFIX records. Luckily, P4 has a construct that represent an external block (*extern block*) with functionality “hidden” inside. The *extern blocks* are a standard way to express counters in P4 language (version 16). Therefore, the functions we listed are represent as *extern blocks* and are supported by our P4 backend.

5 Flow Exporter Generated from P4

During flow exporter analysis, key architecture components were identified. These components are necessary to describe packet headers processing, application protocol processing, flow creation and export process of generated exporter in P4 language. They include packet parser, flow record create/update/export functions and application protocol parsing plugins.

C code with flow exporter is generated by P4 compiler backend developed by this work. Code generation of exporter requires P4 program and exporter source code templates. C source code templates contain placeholders for variable rendering like {{ plugin/name }}. These placeholders are replaced by backend with generated code when compiling P4 program with described exporter components. Generated source codes of exporter can be later compiled into executable application.

Architecture of P4 program consist of P4 parser and control blocks, used to describe functionality of each component of exporter. For every parser and control block in the program, there is an apropriate extern block available to use. These extern blocks help to represent exporter specific functionality like creation of flow key, creation of IPFIX templates, adding flow fields to IPFIX packet, parsing L7 protocols etc.

For the description of packet parser in P4, parser block was used. Generated C code is similar to generated code in FPP backend and its functionality is slightly improved. Parser is compiled as sequence of goto statements, labels, block of codes and switch statements. Output of the generated parser is linked list with extracted headers. Headers that can appear in linked list are specified in special structure as member variables.

Flow cache of the flow exporter accepts mentioned linked list with extracted headers as input. Packet headers processing is programmed via two control blocks. One control block is used to program creation of flow key and fills the flow record with packet protocol fields. Key can be used to find flow record in flow cache and in case record does not exits, new one is created and filled with fields. Second control block is used to program update of the flow record in case record exists in flow cache. Because P4 language does not support loops, control blocks are executed for each header present in the input linked list.

When flow record expires, flow exporter is supposed to export record to flow collector. Records are sent to the collector using IPFIX protocol. With IPFIX protocol, it is needed to specify output templates containing record fields and specify filling of the IPFIX packet. To describe this exporting component, two control block are also used. First control block specifies used templates in IPFIX exporting process and second control block is used to program filling of fields to packet. With second block, only basic flow record can be filled, each application protocols plugin have its own control block to fill packet with extended flow record fields.

Protocol parsing plugins consist of two parsing blocks and one control block. First parser block of each active plugin is executed after flow record is created (post create) and second block is executed before existing flow record is updated. Parsed protocol fields can be added to existing flow record by transition to accept state in parser, when flow extension record exists, it is updated. P4 language does not support handling of text protocol payload. This work solves this problem by adding *extern block* with string extraction and basic manipulation functions. Regular expressions are used to describe text protocols and POSIX capture groups serves to designate substring extraction into a variable. Each plugin contains control block to program how extended flow records are filled into IPFIX packet.

6 Conclusion

By design, P4 language supports expressions that can describe functionality of a network device that is able to process network packets based on at most L4 layer protocols. This paper and our recent work presents a proof-of-concept application (flow exporter) that is capable of processing even L7 protocol information. To achieve this goal, we have identified a set of required functionality that is currently missing in the P4 language. Using a standard construct of P4 language (i.e., *extern blocks*) we were able to describe the whole flow exporter. The description can be currently translated to compilable C source codes using a created P4 backend.

7 Acknowledgement

This work was supported by the CTU grant SGS17/212/OHK3/3T/18.

References

[1] Claise, B., Ed., Trammell, B., Ed., and P. Aitken: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, STD 77, RFC 7011, September 2013.

[2] Havranek, J., Velan, P., Cejka, T., and P. Benacek: Enhanced Flow Monitoring with P4 Generated Flexible Packet Parser. In Proceedings of The International Conference on Autonomous Infrastructure, Management, and Security (AIMS2018), 2018, pp. 21-32, ISBN: 978-3-903176-12-6.

[3] Barefoot Networks, Inc.: P4 compiler eBPF backend, https://github.com/p4lang/p4c/tree/master/backends/ebpf

[4] VMware, Inc.: XDP backend, https://github.com/vmware/p4c-xdp

[5] Benacek, P.: Generation of High-Speed Network Device from High-Level Description, Ph.D. thesis, Czech Technical University in Prague, 2016.

A New Generation of an IPFIX Collector

Lukas Hutak
CESNET z.s.p.o.
Zikova 1903/4, 160 00 Praha 6

lukas.hutak@cesnet.cz

Keywords. IPFIX, collector, network traffic processing, modularity.

Abstract

The level of network traffic significantly grows every year and with it also grows amount of cyber-attacks. Monitoring of network traffic is therefore one of key aspect of network security and capacity planning. Since traditional packet-based traffic analysis is very performance demanding in high speed networks, flow monitoring has become a common method in these types of networks. Instead of individual packet analysis, the packets are aggregated into flows based on a set of common properties and exported using NetFlow or IPFIX protocol to a collector for storage and further analysis.

Typical flow monitoring architecture [1] contains a collector that gather flows from one or more flow exporters. Its conventional purpose is to provide an overview of network usage and store the flow records for later inspection. However, the flows contain valuable information that can be utilized for automated detection of anomaly (and potentially harmful) behavior [2]. Nevertheless, different anomaly detection tools which can receive data from the collector could be based on various input formats and a long-term flow storage could be realized using different technologies, such as traditional databases or specialized flat files. Therefore, a generally deployable flow collector should be easily extensible and configurable to support various use-cases required by network administrators.

This work is based on analysis of the current implementation of open-source collector *IPFIXcol*, whose main feature is effective modular design based on extensibility by a set of plugins. Nevertheless, the collector went through historical development and many impetuous design modifications caused that the further development is no longer feasible. Examination of the existing collector revealed its strengths as well as weaknesses, which played important role in shaping the future generation of the collector.

In this work, I proposed an improved design of the collector founded on an interlinked set of plugins in form of pipeline. As Figure 1 indicates, the new *IPFIXcol2* collector consists of input, intermediate, and output plugins. Briefly speaking, the input plugins are responsible for receiving flow data from exporters, intermediate plugins perform optional flow modification and enrichment, and the output plugins ensure that flow records are stored or forwarded to 3rd party tools for further processing. Combination of plugins used in the collector always depend on a user specified configuration, which makes sure that only required components are running and processing flows. Because one of the goals was also deployment on backbone networks, which can produce hundreds of thousands records per second, considerable emphasis has been placed on performance and effective usage of system resources. For this purpose, an improved communication mechanism between adjacent plugins in the pipeline has been designed and developed.

The proposed design was implemented, and a selected part of plugins have been adapted for the new collector. Existence of the old and new generation allowed to compare flow processing performance and

impact of design changes. In all performed tests, the new collector is significantly faster. Moreover, in some configurations it outperformed the previous generation more than 2 times. Achieved results and internal design improvements provide the basis for further project development.

The work summarized in this abstract has been defended as MSc. Thesis at Faculty of Information Technology, Brno University of Technology in summer 2018.

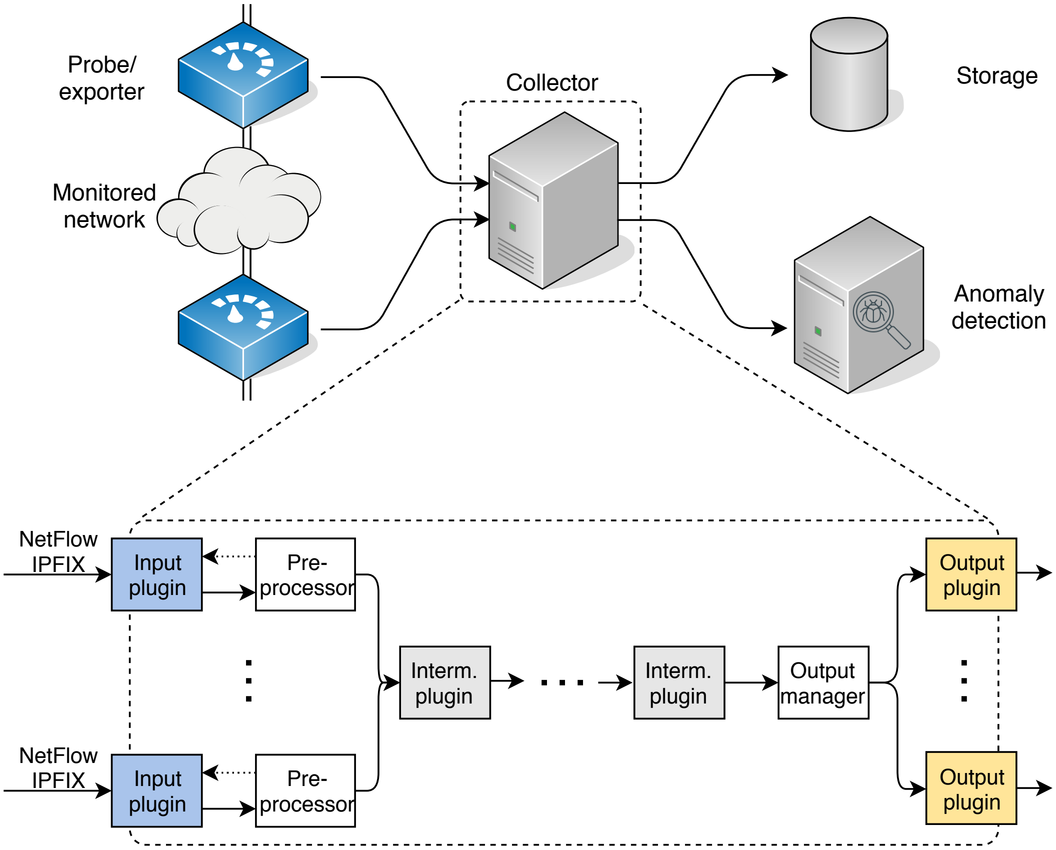


Figure 1: Flow monitoring architecture and the internal design of *IPFIXcol2*

References

[1] Hofstede, R., Čeleda, R., Trammell B., et al.: Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX, IEEE Communications surveys and tutorials, 2014, vol. 16, No 4, p. 2037-2064. ISSN 1553-877X

[2] Čejka, T. and Bartoš, V., et al.: NEMEA: A Framework for Network Traffic Analysis, 12th International Conference on Network and Service Management, 2016

Trace-Share: Towards Provable Network Traffic Measurement and Analysis

Milan Cermak
Masaryk University, Institute of Computer Science
Brno, Czech Republic
cermak@ics.muni.cz

Keywords. Network traffic analysis, Semi-labeled dataset, Research provability.

Abstract

Research in network traffic measurement and analysis is a long-lasting field with growing interest from both scientists and the industry. However, even after so many years, results replication, criticism, and review are still rare. We face not only a lack of research standards, but also inaccessibility of appropriate datasets that can be used for methods development and evaluation. Therefore, a lot of potentially high-quality research cannot be verified and is not adopted by the industry or the community.

The aim of our research is to overcome the mentioned controversy with focus on the whole issue covering all areas of data anonymization, authenticity, recency, publicity, and their usage for research provability. We believe that these challenges can be solved by utilization of semi-labeled datasets composed of real-world network traffic and annotated units with interest-related packet traces only. While the real-world traffic capture needs to be kept private, the annotated units can be freely shared since they only contain the interest-based trace of traffic with a minimum of private information. Our approach enables to insert such annotated events to an unlabeled real-world network traffic dataset and create semi-labeled dataset providing a ground truth used for the development of analytical methods as well as their validation. We do not claim that semi-labeled datasets provide a universal solution to all problems related to dataset usage. However, we aim to show, that it offers more benefits than other current approaches.

The most crucial part in creating a semi-labeled dataset is the adjustment of inserted annotated units so that their features, such as TTL values or packets delay, are indistinguishable from features of the real-world network traffic dataset. For this purpose, we are developing Trace-Mix[1] tool based on ID2T toolkit [2]. The tool analyzes real-world network traffic capture and calculates all necessary features of each connection. Based on these features, it allows to insert selected annotated unit at the specified time and adjust it to match the original dataset (e.g., according to common characteristics of all connections with the target IP address). The IP addresses of the annotated unit can be fully adjusted to match addresses in the original capture. Alternatively, the original IP address distribution of annotated units can be preserved. In this case, original IP addresses from annotated units can serve as a natural label of the inserted traffic. Our approach makes possible to insert any annotated unit into a real-world network traffic and create datasets for development and verification of various measurement and analysis methods.

Semi-labeled datasets can be used for the development of new analysis methods, or adaptation of deployed methods to specifics of a given network, as well as for the verification of their correctness. Figure 1 demonstrates the use of the semi-labeled dataset on the example of a development of network threat detection inspired by PDSA methodology. A similar approach can be used for tweaking of measurement

[1] Trace-Mix tool is publicly available at https://github.com/Trace-Share.

and analysis methods so that they work properly within a given network. In this case, the annotated units serve as ground truth, which must always be recognized while false positives are reduced. In the case of method validation, it is possible to use different annotated units containing a similar event, for example various types of DDoS attack, and test whether they are correctly recognized by the tested method.

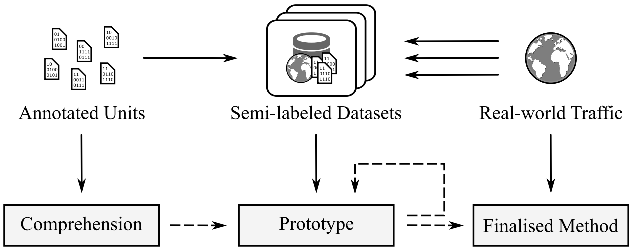


Figure 1: Usage of annotated units and semi-labeled datasets for analysis method development [1].

Sharing of annotated units and cooperation represents a vital component of the whole approach. We are currently developing a sharing platform called Trace-Share intended as a community hub. The platform is built upon essential functions of uploading, searching, downloading, and mixing of annotated units. The unified procedures for normalization with anonymization included will narrow the heterogeneity of the shared data and, at the same time, build trust in the sharing platform. Furthermore, the community-based approach, such as commenting or tagging of units, will help to alleviate some workload from the hub managers and ensure project sustainability with regular updates.

This article is a brief introduction of the concept of semi-labeled datasets, and we are aware that many challenges need to be addressed in further research. Our goal is not to deal with all identified problems at this point, but to present a general solution to start a discussion of its usability. We hope that the follow-up discussions will help us to move forward to a solution that will be accepted by the research community, help us to establish better research conditions, and make research more accessible to other researchers and the industry as well.

Paper origin

The original paper has been accepted and presented at the Network Traffic Measurement and Analysis Conference (TMA 2018) [1].

Acknowledgment

This research was supported by ERDF ”CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence” (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] M. Cermak, T. Jirsik, P. Velan, J. Komarkova, S. Spacek, M. Drasar and T. Plesnik. “Towards Provable Network Traffic Measurement and Analysis via Semi-Labeled Trace Datasets,” in *2018 Network Traffic Measurement and Analysis Conference (TMA)*. Vienna, Austria: IEEE, 2018. doi:10.23919/TMA.2018.8506498.

[2] E. Vasilomanolakis, C. G. Cordero, N. Milanov, and M. Muhlhauser, “Towards the creation of synthetic, yet realistic, intrusion detection datasets,” in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, apr 2016, pp. 1209–1214.

Multi-level Anomaly Detection in IoT Networks

Dominik Soukup, Tomas Cejka, Simon Stefunko
CTU in Prague
Thákurova 9, 160 00 Prague, Czech Republic

soukudom@cesnet.cz, {cejkato2, stefusim}@fit.cvut.cz

Keywords. IoT, anomaly detection, encrypted traffic

Abstract

The number of cyber attacks grows rapidly every year. The percentage of encrypted traffic is still growing [1] too, and thus security analysis becomes much more challenging. Also, the security is not topic just for network perimeter, but it must cover visibility across the whole network. Lots of data are exchanged locally on the network edge, and we must be able to analyze this traffic efficiently. Currently used approaches to detection based on simple pattern matching and deep packet inspection are not feasible without data decryption. Nevertheless, decryption without knowledge of private keys is a computationally intensive problem that is unsolvable in practice.

Nowadays, access control based on predefined static rules at the perimeter is not secure enough. Systems should be prepared to understand how communication looks like and detect unexpected patterns. Instead of decrypting the traffic, we need to find a way to analyze the encrypted traffic as it is and identify the behavior of connected devices based on visible characteristics of data flows. Moreover, this problem is more complicated in the Internet of Things (IoT) networks because we have many different devices with discrepant traffic patterns. From the security perspective, we need a solution that is able to learn the behavior of different devices with respect to distributed fog and edge architecture [2] that has limited resources and to assist users in order to sustain the level of security. For this challange, machine learning (ML) brings new options to recognize a type of communication despite the heterogeneity of encrypted IoT traffic right at the network edge.

This paper is primarily focused on IoT networks that contain IP devices, such as gateways, sensors, and mobile phones. Our aim is to design scalable monitoring system and its features for IoT gateways, to analyze the behavior of IoT devices, classify them according to trained classes, and to detect anomalies at the network edge. This approach to securing infrastructure brings better visibility and improves threat detection because there is the biggest insight without any obstacles at the network edge. Our goal is to create a system that can notify owners of IoT gateway about suspicious behavior observed even in the encrypted traffic. In our case, anomalous traffic represents some change in the behavior of a device that can be occur after infection by malware or after some configuration changes.

In this paper, we propose an approach to detect anomalies in modern IoT networks. We show that it is feasible to monitor the IoT network, to learn automatically standard behavior (having at least a short training dataset at the beginning), and detect anomalous activities using a ML approach. We describe multi-level architecture that is necessary for proper scaling. However, the implementation covers just the first layer. Based on our domain knowledge and related work, we selected ML features that are independent of network hosts and describe flow behavior even in encrypted traffic. In total, the feature set creates a vector of 32 items. Combination of two existing semi-supervised techniques that we used

ensures higher reliability of anomaly detection and improves results achieved by a single method. We describe conducted classification and anomaly detection experiments allowed thanks to existing and our training datasets. With the selected dataset covering more types of traffic and 20 IoT devices, we have achieved F1-score 86 % for Local Outlier Factor model and 77 % classification accuracy using Random Forest. The created dataset and source code are publicly available at our repository [3].

The achieved results are satisfiable because our models are very lightweight, so they are resource-economical. The correctness of results was verified on the dataset with up to 400 flows. A small input data set and low resource requirements are the key features for the first layer of our multi-level architecture.

Note: The work summarized in this abstract is currently under review for the International Conference on Network and Service Management.

Acknowledgment

This work was supported by the CTU grant No. SGS17/212/OHK3/3T/18 funded by the Ministry of Education, Youth and Sports of the Czech Republic, CESNET and *Secure Gateway for Internet of Things (SIoT)* project No. VI20172020079 funded by the Ministry of the Interior of the Czech Republic.

References

[1] G. Gebhart, "We're Halfway to Encrypting the Entire Web," 2017.

[2] Statista, "Fog computing and the internet of things: Extend the cloud to where the things are."

[3] D. Soukup and T. Cejka, "Nemea-siot," 2019.

Low-Cost CMOS Power Consumption Data Dependency Demonstrator Concept

Jan Bělohoubek[1] **and Robert Vik**[2]
[1]Czech Technical University in Prague
[2]University of West Bohemia in Pilsen

[1]jan.belohoubek@fit.cvut.cz, [2]rvik@ket.zcu.cz

Keywords. CMOS, data dependency, combinational logic, demonstrator

Abstract

As digital devices penetrate into many areas important for the present society, it is important to analyze even potential threats to mitigate device vulnerability during the lifetime of a digital device. Our research is targeted on the illuminated CMOS combinational logic, whose power consumption is data dependent [1].

The power consumption data dependence may be potentially used to obtain secret values from the secured digital devices.

In this contribution, we follow concepts described in [1] and we introduce the low-cost data dependency demonstrator concept based on the CMOS 4011 circuit. The circuit decapsulation process will be described and the power consumption data dependence principle will be explained and demonstrated.

Acknowledgment

The author[1] acknowledges the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16 _019/0000765 “Research Center for Informatics” and the CTU grant SGS17/213/OHK3/3T/18.

References

[1] Bělohoubek, Jan and Fišer, Petr and Schmidt, Jan, in *22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS 2019).*

Analysis of the Dummy Rounds Scheme Optimizations

Stanislav Jerabek, Jan Schmidt
Faculty of Information Technology, Czech Technical University in Prague
Thakurova 9, 160 00 Prague

{jerabst1, schmidt}@fit.cvut.cz

Keywords. Hiding, FPGA, DPA, Hiding in time, Dummy rounds.

Abstract

In [1], we proposed a scheme to make hardware implementations of Feistel Networks [2] and Substitution-Permutation Networks [3] more resistant against Side Channel Attacks (SCA) such as Differential Power Analysis (DPA) [4] [5]. Further analysis of the scheme has been published in [6].

The Dummy Rounds scheme employs the fact that the cipher networks consists of similar *rounds*. It further assumes that the implementing hardware can execute $M > 1$ rounds in a clock cycle. The actual M for every clock cycle is chosen randomly.

The randomness of the execution is supposed to hide the real computation from an attacker. To prevent redundant rounds from leaking data, they process random data rather than the real data from preceding rounds.

Experimental evaluation on the PRESENT cipher [7] in [1] did not give approving results. After refining the accuracy measuring process, much better and almost satisfactory results were obtained. The biggest weakness was in the first clock cycle.

The states of the algorithm, together with transition probabilities, form a Markov chain. Using the state probabilities, we can calculate the probability that the round r was executed as active in a given state. In general, these probabilities vary with clock cycle number for any given round number. The clock cycle with the maximum round execution probability offers the best point for an attack on the given round. The gist of this contribution is to *design* the transition probabilities so that the probability of round execution remains the minimum possible over the entire computation.

The problem with the first and last rounds follows directly from the fact that $m > 0$. There is no freedom and no randomness in the first and last clock cycle. Therefore, we have to fix $m = 0$ in all cases.

As a remedy to the leak in clock cycle 0, the original proposal suggest to randomly postpone the beginning of the computation. This is precisely what can happen with $m = 0$: there can be a random number of redundant rounds at the beginning, and then some active rounds can occur. Therefore, any scheme with $m = 0$ fulfills this request as a special case.

In the above mentioned Markov chain, the transitions have a regular structure. Let $S_{n,r}$ be the state that has executed rounds $1 \ldots r$ in the clock cycle n. Possible transitions from this state are the transitions to states $S_{n+1,r+m}, \ldots S_{n+1,r+M}$.

In the original proposal, M rounds are executed serially, and a random output is chosen. We have to suppose that the attacker can distinguish the execution of a particular round. Then, instead of N clock cycles, we model $K = MN + 1$ *slots*. Then, a yet simpler (but larger) model can be constructed.

Let $S_{k,r}$ be the state that has executed rounds $1 \ldots r$ in the slot k. From this state, only two transitions are possible. Either, the next round will be taken as active, which leads to the state $S_{k+1,r+1}$. Or, the round is redundant, which transits to the state $S_{k+1,r}$. An example is in Figure 1.

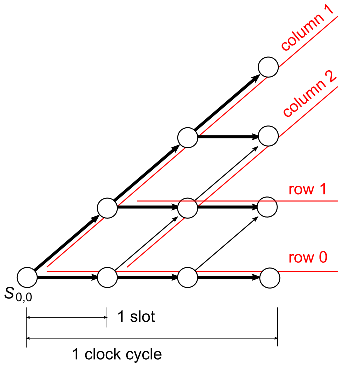


Figure 1: A part of a slot-level model with $m = 0$, $M = 3$.

This model is more general than the round control in the original proposal. That controller takes $m \dots M$ active rounds, and the rest is discarded, so that only thick lines in Figure 1 can be followed. Practically at no hardware cost, we can obtain finer control, more random operation and simpler analysis.

The optimum protection executes a number of redundant rounds first. Then, it executes all rounds as active, and finally executes redundant rounds to the required number of slots.

An attack to any round must collect more traces to achieve certain probability, that the desired round has been executed with a given probability in the collected traces. The amount of protection depends on work effort only. The function is, unfortunately, almost linear in the practical range of work effort. With an average work effort, around 40 times the number of traces are required to collect compared with the unprotected circuit.

Paper origin

This paper has been accepted and presented at the conference 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS) in 2019.

Acknowledgment

We thank again Prof. Dr.-Ing. Tim GĂźneysu for his suggestions, which, not surprisingly, also follow from the analysis in this contribution.

References

[1] S. Jeřábek, J. Schmidt, M. Novotný, and V. Miškovský, “Dummy rounds as a DPA countermeasure in hardware,” in *2018 21st Euromicro Conference on Digital System Design (DSD)*, Aug 2018, pp. 523–528.

[2] H. Feistel, “Cryptography and computer privacy,” *Scientific American*, vol. 228, no. 5, pp. 15–23, 1973.

[3] C. E. Shannon, “Communication theory of secrecy systems,” *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct 1949.

[4] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[5] M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart, "Power analysis, what is now possible..." in *Advances in Cryptology — ASIACRYPT 2000*, T. Okamoto, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 489–502.

[6] S. Jeřábek and J. Schmidt, "Analyzing and optimizing the dummy rounds scheme," in *2019 22nd International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, Apr 2019.

[7] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, P. Paillier and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466.

Health Monitoring and Fault Detection Using Memristive Switching Behavior in DSC Array

Vishal Gupta[*], Jimson Mathew[$], and Marco Ottavi[*]
[*]Department of Electronic Engineering, University of Rome, Tor Vergata
Roma, Italy
[$]Department of Computer Science and Engineering, Indian Institute of
Technology Patna,
Patna, India
[*]vishal.gupta@students.uniroma2.eu, ottavi@ing.uniroma2.it
[$]jimson@iitp.ac.in

Keywords. Dye Solar Cell, Memristor, Health Monitoring, Fault Analysis, Switching Behavior.

Abstract

Photovoltaic (PV) technology plays a vital role in green energy revolution and emerging as an important area of study due to its fault tolerance based system design. Dye solar cell is a promising candidate due to its low cost of fabrication and high energy conversion efficiency. It has been already reported that dye solar cell also shows the memristive behavior with the same production process. Resistive switch device, also known memristor, has potential to replace CMOS device for the switching application. Memristors have also been proposed as a promising candidate for numerous other applications such as logic design, non-volatile storage, Content-addressable memory (CAM), sensing, neuromorphic computing, Physically Unclonable Functions (PUFs) and reconfigurable computing. In this paper, we investigate health of the PV cell by using the switching behavior of memristor device and also provide a mechanism for the detection of various types of faults in the PV array. For this experiment, we used DSC solar cell array, in which a cell acts as a memristor device and used as a sensor for the fault detection.

A memristor is a two terminal device with two stable states of its resistance value, where 1 and 0 are associated with the stable resistance values. These stable states are termed as high resistance state and low resistance state respectively [1]. The resistance of the memristor depends upon the applied voltage i.e. Vset and Vreset. Memristor can be changed from state 1 to state 0 by applying Vreset or negative voltage sweep, and changed from state 0 to state 1 with a voltage applying Vset or positive voltage sweep. This type of device exhibits a pinched-hysteresis in the I-V curve and also shows switching from one stable state to another stable state as shown in Figure 1(a) [2].

To monitor the status of a PV array we capture the state change behavior of TiO_2 based memristor device. This TiO_2 based memristor could be embedded inside a photovoltaic array (Dye-sensitized solar array [3] because the fabrication process is identical for both photovoltaic cell and memristor device. This time analysis of memristor device provides valuable information for behavior of the system under measurement both in terms of power generated in normal conditions, and for fault detection.

A photovoltaic (PV) cell or solar cell is a device that converts solar energy (photons) in to DC supply. The circuit for an ideal solar cell modeled by a current source with a parallel connection of diode, but for the practical device a shunt resistance and a series resistance component are added. Figure 1(b) shows the equivalent circuit for the DSC solar cell.

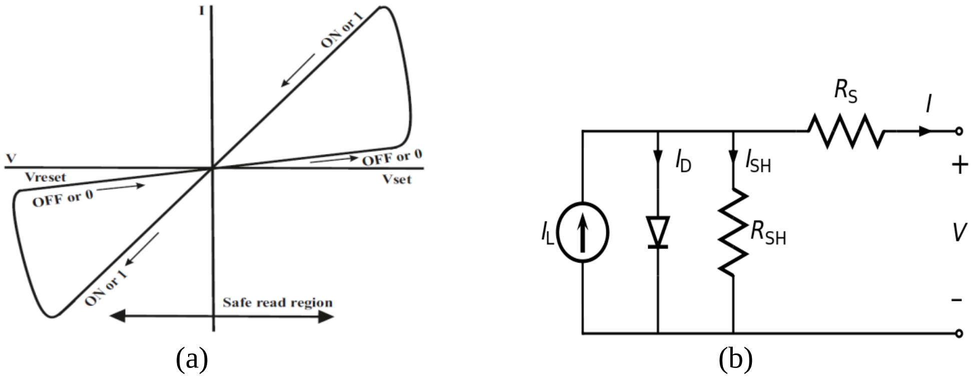


Figure 1. (a) I-V characteristic of memristor [2] (b) Equivalent circuit of a PV cell.

As an example of this approach, in the following, we analyze various configuration of solar cell arrangements: single and four series connected solar cells with a memristor element. The configurations are shown in Figure 2(a) & 2(b) respectively where is added a small wire resistance. In our simulations, a healthy dye solar cell has open circuit voltage of 708.67mV and the short circuit current of 15mA. Behavior of the memristor were characterized by switching the device from high impedance state to low impedance state. Initially, we set our TiO_2 based memristor into the high resistance state, then we triggered this by the dye-sensitized solar cell configuration.

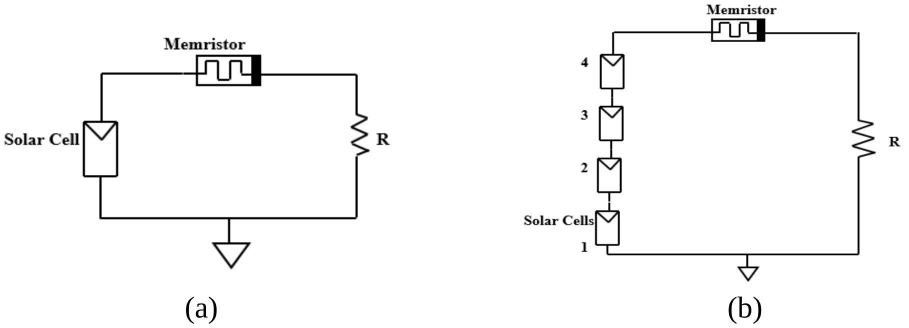
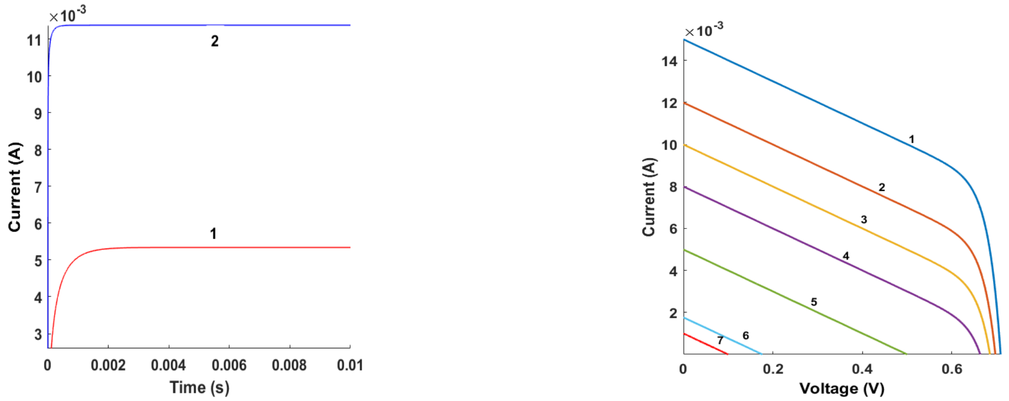


Figure 2. Schematic of experimental circuit with memristor element: (a) single solar cell setup. (b) Series configuration of four solar cells.

In Figure 3(a), curve 1 (single solar cell setup) and curve 2 (configuration of four series connected solar cells) show the variation in current flow through the memristor with respect to the time. This initial variation in current with respect to time represent the switching mechanism for the device from high resistance state to the low resistance state. After certain time interval current goes saturate which shows the minimum current required to switch the memristor from one stable state to the another stable state. The time taken by the memristor device is 1.98ms and 0.29ms for the single solar cell setup and the configuration of four series connected solar cells respectively. The output current of the series connected cell is higher than the single cell configuration, which also shows the faster switching as compared to the single solar cell setup. The output power and current of the solar cell is related to the irradiance, (i.e. the amount of solar power per unit are that is shining on the array itself) and this is modeled by the I_L value in the Figure 1(b). Therefore, while in the following we will use this mechanism to detect degradation and faults in the array, it is worth noticing that the memristor switch time could be used to provide a runtime estimation of the power output of the array in nominal operating conditions.

Degradation of solar power plays a crucial role in the PV system. Solar cell experience degradation due to the unavoidable circumstances like UV exposure, thermal cycling, damp heat, humidity freeze and weather cycle. This degradation may cause faults in the system or responsible for the system failure. We consider the family of defects in a cell that results in one of the solar cells is stuck permanently not healthy or partially healthy and full degradation or partial degradation cell voltage/current generated [4]. These faults can be categorized are as ground fault, open-circuit (OC) and short circuit (SC) fault in the solar cell.



(a)

(b)

Figure 3. Simulation results: (a) Curve 1 is for single solar cell setup and curve 2 for series configuration of solar cell setup. (b) Effect of solar cell degradation and faulty cell on I-V Curve.

We also analyzed the state change behavior of memristor device for various cases of unhealthy and faulty cells. Figure 3(b) shows current-voltage characteristics and Table 1 shows the corresponding numerical data for various current generator and corresponding maximum

Solar cell current generator (mA)	**Solar cell power generation(W)**	**Time (ms)**
0	0	Infinite
1	24.975µ	Infinite
1.765	77.800 µ	38.76
5	624.375µ	5.97
8	1.598m	2.99
10	2.496m	2.21
12	3.566m	2.15
15 (Healthy Cell)	5.320m	1.98

Table 1. Time Analysis for the state change behavior of memristor device for various source of current generators.

output power for the solar cell. Curve 1 (solar cell current generator for 15mA) represents the behavior for healthy cell and have maximum output power is 5.320mW. Curve 2-6 (solar cell current generator for 12mA to 1.765mA) show the degradation in the maximum output power for the solar cells, are considered as an unhealthy solar cell. Curve 7 represents the solar cell current generator for 1mA and switching time for the memristor device is infinite and this cell considered as a faulty solar cell. From these results as shown in Table 1, we conclude that switching time for the memristor device increases as the generated power by the solar cell decreases. This shows that switching time for unhealthy cells is more than the healthy cell. If generated maximum power for solar cell is lower than 77.8 µW, memristor device is unable to change their state and considered as a faulty cell.

Detection of faults in the solar array is essential to prevent the system failure. We also examined the state change behavior of memristor for the faulty cells in the series configuration of four cells as shown in Figure 2(b). Table 2 shows time taken by the memristor to switch their state from high resistance state to the low resistance state for the number of faulty cells in this configuration. As the number of faulty cells increases the switching time of the device increases. This time analysis of device in the array is helpful for finding the faults. This series configuration of PV cells allow the short circuit (SC) fault and ground fault in the array and can be found by the switching time of the memristor device. We can also find the number of faulty cells in the series configuration. If one of the cell has open circuit (OC) fault in this configuration, the maximum output power will be zero and system will be failed.

No of faulty cell	**Time (ms) in the series combination of 4 cell configuration**
1	0.35
2	0.63
3	1.98

Table 2. Time Analysis for the state change behavior of memristor device for faulty cell in the series combination of the 4 cells.

Paper origin

This paper has been extracted from the accepted paper in 25th IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS) 2019, Greece and will be presented soon.

References

[1] Palson, C. L., Krishna, D. D., Mathew, J., Jose, B. R., Ottavi, M., and Gupta, V.: Memristor based adaptive impedance and frequency tuning network, 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), April 2018, pp. 1-2.

[2] Mathew, J., Yang, Y., Ottavi, M., Brown, T., Zampetti, A., Carlo, A. D., Jabir, A. M., and Pradhan, D. K.: Fault detection and repair of DSC arrays through memristor sensing, IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS) 2015, pp. 7-12.

[3] Sastrawan, R., Renz, J., Prahl, C., Beier, J., Hinsch, A., and R Kern.: Interconnecting dye solar cells in modules i–v characteristics under reverse bias, Journal of Photochemistry and Photobiology A: Chemistry, 2006, Vol 178, No. 1, pp. 33–40.

[4] Mathew, J., Ottavi, M., Yang, Y., and Pradhan, D. K.: Using memristor state change behavior to identify faults in photovoltaic arrays, IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), 2014, pp. 86–91.

PBO-Based Fault Selection for Compact Test Generation

Robert Hülle, Petr Fišer, Jan Schmidt
Faculty of Information Technology, Czech Technical University in Prague
Prague, Czech Republic

{hullerob,fiserp,schmidt}@fit.cvut.cz

Keywords. ATPG, test compaction, SAT, PBO, fault ordering.

Abstract

The length of a test is of vital importance both during manufacturing and in-application testing. Methods used to compact a test set can be divided between *static compaction* and *dynamic compaction*. Static compaction is used on a pre-generated test set to reduce the test pattern count while keeping desirable coverage [1–3]. Dynamic compaction is used during test pattern generation itself [4–9]. Dynamic compaction can produce more compact test sets, at the expense of computational resources.

Multiple-target test generation (MTTG) [6–8] is further improving dynamic compaction by explicitly targeting multiple faults in one test pattern. To successfully generate a test pattern, the targeted faults need to form independent faults set.

In this abstract, we present two novel methods of dynamic compaction to generate a compact test set by utilizing an optimizing SAT solver (PBO). We describe a heuristic to increase the probability of additional fault detection. Next, we describe a method to select optimal fault for the ATPG process, forming an implicit fault ordering.

Paper origin

The original paper has been submitted to the Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems 2019. At the time of this writing, the paper is under review.

Acknowledgment

The authors acknowledge the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16 019/0000765 “Research Center for Informatics” and the CTU project SGS17/213/OHK3/3T/18. Computational resources were supplied by the Ministry of Education, Youth and Sports of the Czech Republic under the Projects CESNET (Project No. LM2015042) and CERIT-Scientific Cloud (Project No. LM2015085) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

References

[1] L. N. Reddy, I. Pomeranz, and S. M. Reddy, “ROTCO: a reverse order test compaction technique,” in *Proceedings Euro ASIC ’92*, June 1992, pp. 189–194.

[2] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, “Fast algorithms for static compaction of sequential circuit test vectors,” in *Proceedings. 15th IEEE VLSI Test Symposium (Cat. No.97TB100125)*, April 1997, pp. 188–195.

[3] Xijiang Lin, J. Rajski, I. Pomeranz, and S. M. Reddy, “On static test compaction and test pattern ordering for scan designs,” in *Proceedings International Test Conference 2001 (Cat. No.01CH37260)*, Nov 2001, pp. 1088–1097.

[4] I. Pomeranz, L. N. Reddy, and S. M. Reddy, “Compactest: a method to generate compact test sets for combinational circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 7, pp. 1040–1049, July 1993.

[5] S. Remersaro, J. Rajski, S. M. Reddy, and I. Pomeranz, “A scalable method for the generation of small test sets,” in *2009 Design, Automation Test in Europe Conference Exhibition*, April 2009, pp. 1136–1141.

[6] Jau-Shien Chang and Chen-Shang Lin, “Test set compaction for combinational circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 11, pp. 1370–1378, Nov 1995.

[7] G. Tromp, “Minimal test sets for combinational circuits,” in *1991, Proceedings. International Test Conference*, Oct 1991, pp. 204–.

[8] S. Eggersglüß, R. Krenz-Baath, A. Glowatz, F. Hapke, and R. Drechsler, “A new SAT-based ATPG for generating highly compacted test sets,” in *15th IEEE Design and Diagnostics of Electronic Circuits and Systems*, April 2012, pp. 230–235.

[9] S. Eggersglüß, K. Schmitz, R. Krenz-Bååth, and R. Drechsler, “On optimization-based ATPG and its application for highly compacted test sets,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 12, pp. 2104–2117, 2016.

Author Index

Sponsors

STMicroelectronics

STMicroelectronics is a world leader in providing the semiconductor solutions that make a positive contribution to people's lives, today and into the future. ST is a global semiconductor company with net revenues of US$ 8.35 billion in 2017. Offering one of the industry's broadest product portfolios, ST serves customers across the spectrum of electronics applications with innovative semiconductor solutions for Smart Driving and the Internet of Things. By getting more from technology to get more from life, ST stands for life.augmented.

ASICentrum

ASICentrum, established in 1992 in Prague is a design center of EM Microelectronic and a competence center of ETA, belonging to the Swatch Group. EM Microelectronic is one of the most innovative IC providers. It developed and manufactured the smallest and the lowest power consuming Bluetooth chip on the market, the top performing optical sensors for optical office as well as gaming mice and it was the first to release the award-winning world-first dual-frequency NFC + RAIN RFID em|echo.

CZ.NIC

CZ.NIC, interest association of legal entities, was founded by leading providers of Internet services in 1998. The association currently has 114 members. The key activities of the association include operation of the domain name registry for the .CZ domain, operation of the CZ top-level domain and public education in the area of domain names. The association is now intensively working on development of the DNSSEC technology and mojeID service, extension and improvements of the domain administration system and support of new technologies and projects beneficial to the Internet infrastructure in the Czech Republic. CZ.NIC is a member of the EURid association, managing the European domain – EU, and other similarly oriented organizations (CENTR, ccNSO etc.).

CESNET

CESNET is an association of universities of the Czech Republic and the Czech Academy of Sciences. It operates and develops the national e-infrastructure for science, research and education which encompasses a computer network, computational grids, data storage and collaborative environment. It offers a rich set of services to connected organizations.

ISECO.CZ

We are information security. Our mission is to be a trusted partner to our clients in the field of information and IT security, to understand the client‘s needs and deliver the best technical solutions and services with an individual approach.

Czech Technical University in Prague

The conference has been sponsored by the CTU grant SVK 50/19/F8.

Partners

IEEE Student Branch at Czech Technical University in Prague

IEEE Young Professionals

Computer (C) Society Chapter of the Czechoslovakia Section of IEEE