Performance evaluation of INDECT security architecture

Evaluación del rendimiento de la arquitectura de seguridad INDECT

Petr Machník
Ph. D. Telecommunication Technology
VSB-Technical University of Ostrava
Ostrava, Czech Republic
petr.machnik@vsb.cz

Marcin Niemiec
Ph. D. Telecommunications
AGH University of Science and Technology
Krakow, Poland
niemiec@kt.agh.edu.pl

Manuel Urueña
Ph.D. Telecommunication Technologies
Universidad Carlos III de Madrid
Leganés (Madrid) Spain
muruenya@it.uc3m.es

Nikolai Stoianov
Ph.D. Information Security
Bulgarian Defense Institute
Sofia, Bulgaria
n.stoianov@di.mod.bg

Abstract– **This paper evaluates the performance of the key elements of the security architecture developed by the INDECT project. In particular it first evaluates three different concurrent error detection mechanism (parity check, Berger code, and cyclic redundancy check) developed in software- and hardware-based implementations of the INDECT block cipher. It also analyses the performance hit in secure web servers when enabling TLS/SSL with mutual authentication. Finally, it evaluates the throughput and delay of traffic in the virtual private network based on the OpenVPN software package with the implemented INDECT block cipher. The results of these evaluations demonstrate that the proposed mechanisms, and by extension the whole INDECT security architecture, are viable and can be used in high-performance Police information and communication systems.**

Keywords– **INDECT project; INDECT security architecture; performance evaluation; Police ICT systems; security.**

Resumen– **Este artículo evalúa el rendimiento de los principales elementos de la arquitectura de seguridad desarrollada por el proyecto INDECT. En particular, en primer lugar evalúa tres mecanismos diferentes de detección concurrente de errores (comprobación de paridad, códigos Berger y verificación por redundancia cíclica) desarrollados en las implementaciones *software* y *hardware* del algoritmo de cifrado de bloque INDECT. También se analiza el impacto en el rendimiento de los servidores web seguros cuando se activa TLS/SSL con autenticación mutua. Por último, evalúa el rendimiento y el retardo del tráfico en una red privada virtual, basada en el *software* OpenVPN con el algoritmo de cifrado INDECT. Los resultados de estas evaluaciones demuestran que los mecanismos propuestos, y el algoritmo de cifrado INDECT, son viables y pueden usarse en sistemas de información y comunicaciones de alto rendimiento para la Policía.**

Palabras Clave– **Arquitectura de seguridad INDECT; evaluación de prestaciones; proyecto INDECT; seguridad; sistemas TIC de la Policía.**

1. INTRODUCTION

Nowadays ICT (Information and Communication Technology) systems require a constantly growing degree of security, because criminal activities in the cyber space represent an increasingly higher threat. Especially the law enforcement agencies (LEAs) have very strict demands for the security and privacy of data in their information systems.

INDECT (Intelligent information system supporting observation, searching and detection for security of citizens in urban environment) is a research project funded by the EU 7th Framework Program that is developing cost-effective tools for helping European Police services to enforce the law and guarantee the protection of their citizens [1]. Thus, the so-called INDECT system is composed by a set of novel applications and ICT services designed to help Police forces in their current investigations, as well as to fight new forms of cyber-crime.

In particular, INDECT is also designing a security architecture that was already described in [2]. The goal of this paper is to evaluate performance

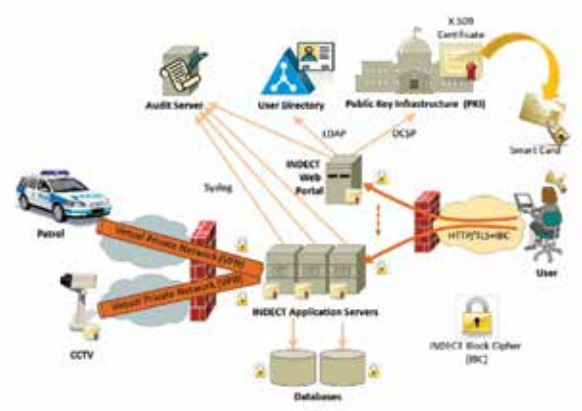
Recibido: 07/07/2017 Aceptado: 07/11/2017 ITECKNE Vol. 15 Número 1 • ISSN 1692-1798 • ISSN Digital 2339-3483 • junio 2018 • 34 - 42
http://dx.doi.org/10.15332/iteckne.v15i1.1962

of the main components of the INDECT security architecture to confirm that the newly proposed tools and mechanisms do not have any negative impact on the efficiency of the Police information systems that will be secured in this way.

2. INDECT SECURITY ARCHITECTURE t2

The INDECT security architecture [2] provides a set of common security services, including authentication, authorization (access control), non-repudiation, privacy/auditing, communication security, data confidentiality and integrity, by using standardized protocols and mechanisms. Fig. 1 shows a simplified view of the integrated INDECT security architecture for Police ICT systems.

Fig. 1. INDECT SECURITY ARCHITECTURE



Source: The authors.

The main components of proposed INDECT security architecture are:

- **Public key infrastructure (PKI)** – It issues, manages, stores and revokes X.509 certificates used in INDECT systems. Certificates are issued to all INDECT users and ICT systems for authentication, as well as for securing their communications.
- **LDAP user directory** – It stores all users' contact data and credentials for legacy systems that do not support certificate-based authentication. The user directory also stores general authorization information, such as users' clearance level or the specific applications they can access.
- **Audit server** – All relevant user actions (e.g. accessing applications or requesting classified information) are logged, both locally and in a secure centralized system. These logs are constantly reviewed by security personnel and Police auditors in order to detect suspicious behaviors.
- **INDECT web portal** – It is the homepage of INDECT users, allowing them to access the different services and applications available to them, according to particular scenarios (e.g. in a crisis). User authentication is based on X.509 certificates and/or user credentials stored in the LDAP user directory.
- **Application servers and databases** – They execute the INDECT services, applications and tools, and store their associated information. They also authenticate users by means of user certificates, although they can also handle application-specific user authorization attributes (e.g. which CCTV cameras a given user may access). We assume that most INDECT applications provide a web-based interface, and most ICT services will also be web-based, implementing SOAP or REST interfaces and using TLS/SSL for secure communications, featuring mutual client-server authentication.
- **Virtual private networks (VPNs)** – They are employed for protecting communications with external Police users and devices. Only encrypted traffic is allowed to go through the Police data center firewall, which blocks all external traffic by default and should feature additional security mechanisms such as intrusion detection/protection systems (IDS/IPS).
- **Smart cards (SC)** – The store users' certificates, which are issued by the INDECT PKI and used for access control by the central INDECT web portal, as well as for encrypting and signing e-mails and documents.

2.1 INDECT cryptographic algorithms

Nowadays, encryption is used to ensure the confidentiality of digital data. It is the most important task of modern cryptography. Encryption relies on transforming confidential data into another encrypted form, unreadable to anyone except of users which possess the cryptographic key. The encryption is realized by using an appropriate algorithm, called cipher.

Some novel cryptographic algorithms were developed in the Work Package 8 of EU INDECT project. The main algorithm in this group is a symmetric cipher, called INDECT Block Cipher (IBC). In general, IBC cipher consists of nonlinear transformations, which are dependent on the key [3]. These functions ensure the protection of confidential data. The crucial transformation during the encryption process in IBC is substitution by secure S-boxes. These S-boxes makes each encryption secure and highly unique. The construction of this cipher is based on substitution and permutation functions that are used in each round of the IBC cipher. This structure ensures a high performance and a fast data encryption.

The IBC algorithm is a block cipher, where each 256-bit block of data is divided into 64 sub-blocks. Sub-blocks are transformed by the appropriate S-box and output values are concatenated back into the 256-bit block. At the end of the round, the permutation – also based on S-box – is performed. This function modifies the 256-bit block of data. All these steps are repeated for a number of iterations (minimum 8 times).

One of the most novel ideas provided in the IBC cipher is unique approach to cryptographic key. The key is a pseudo-random sequence; however it is used to create new S-boxes. These S-boxes are based on the AES (Advanced Encryption Standard) S-box, and ensure high-level of security. This approach ensures that 5.35·1018 new S-boxes is used in IBC cipher. All new S-boxes are used as a non-linear transformation: substitution or permutation.

2.2 INDECT Public Key Infrastructure

One of the main characteristics of INDECT project is that it is composed by multiple heterogeneous systems that exchange sensitive information among them. Therefore it is necessary to fulfill all requirements for information security: access control, authentication, non-reputation, data confidentiality, communication security, data integrity, availability and privacy [4]. One of the main elements of the security infrastructures being deployed to provide these security properties is the INDECT Public Key Infrastructure (PKI). This PKI is the base for creating a heterogeneous and secure environment, based on X.509 certificates, public keys and asymmetric cryptographic algorithms. The INDECT PKI architecture has a hierarchical, two-level structure [4].

PKI is a common way to solve the problems related to the distribution of public keys, because it offers the scalability that is required for large communication and information infrastructures. A PKI is usually used to create policies and mechanisms for asymmetric key management, where public keys are distributed in the form of the so-called digital certificates. However, in INDECT the information that is included in certificates is more than just a public key, since they are also employed for authentication and authorization purposes. Certificates are digitally signed to ensure the integrity and validity of contained information [5].

For issuing a certificate the following data are required and registered:

- Username and password - used by RA to access user's certificate data. RA can edit certificate data until sends request to CA to issue user's certificate.
- User's a-mail address - obligatory information for issuing the certificate.
- Common name (CN) - this value is used as a short textual description of certificate. Usually the name of Police officer can be used.
- Country - in a national LEA this field will be same for all users, but in an international LEA organization this field may have of the country of origin of the user.
- Organization - this field should be the user's organization name. In our case, it can specify the LEA office (for example LEA-EU, LEA-BG, etc.).
- Given name – the first name of the user.
- Surname – the last name of the user.
- UID - this is an extension field of X.509v3 certificates that is employed as a unique identifier for the Police user (for example the officer number).
- Lvl - this field shows maximum clearance level of the user. This field is also an extension to the standard certificate structure, and the five possible values are from 0 (Unclassified) to 4 (Top Secret).

Another important thing is key usage. Based on specific functions that INDECT PKI users have, we define following possibilities for using of certificate separated in two main areas – key usage and extended key usage Table I:

TABLE I
KEY USAGE AREAS

INDECT PKI user certificate key usage	Digital signature
	Non-repudiation
	Key encipherment
	Data encipherment
INDECT PKI user certificate extended key usage	Client authentication
	Email protection
	Time stamping
	SSH client
	MS smart card logon
	MS document signing
	MS Encrypted File System (EFS)
	MS EFS recovery (only for administrators)
	PDF signing

Source: The authors.

2.3 INDECT communications security

Given the distributed nature of the INDECT system, one of the main components of the secure communication infrastructure is a virtual private network (VPN) framework that will enable secure communications among multiple remote nodes and servers interconnected over public networks. Nowadays VPNs are mostly based on two different technologies – SSL (Secure Socket Layer) and IPsec (Internet Protocol Security).

The most suitable open-source SSL VPN solution is the OpenVPN software package [6]. OpenVPN can be installed in computers or smartphones with almost any of current operating systems (Linux, Windows, Mac OS X, Android, iOS). OpenVPN offers two basic modes that run either as a layer 2 or layer 3 VPN. Within the INDECT system, users will employ mainly OpenVPN to securely communicate between their remote terminals (desktop, laptop, tablet, smartphone, etc.) and servers located in the police headquarters. The INDECT Devices CA will be employed to authenticate the individual terminals.

The Strong Swan software package [7] provides an open-source IPsec VPN solution. StrongSwan can be installed in computers or smart phones with various operating systems (Linux, Mac OS X, FreeBSD, Android). It is fully compatible with other standard IPsec VPN implementations, and thus can be used in networks with mixed equipment. StrongSwan implements both IKEv1 and IKEv2 (internet key exchange) protocols, and it fully supports IPv6.

Both types of VPN software can employ pre-shared keys, X.509 certificates, or smart cards for the device authentication.

3. ENCRYPTORS' ERROR DETECTION PERFORMANCE EVALUATION t2

Currently, INDECT Block Cipher (IBC) has a few software implementations. One of them was presented in [8]. The next step towards using this application in practice was implementation of error detection methods to increase the reliability of encryption software. It ensures the higher level of data security. The console-based interface of a new software implementation – IBC cipher with error detection algorithms [9] – is presented in Fig. 2.

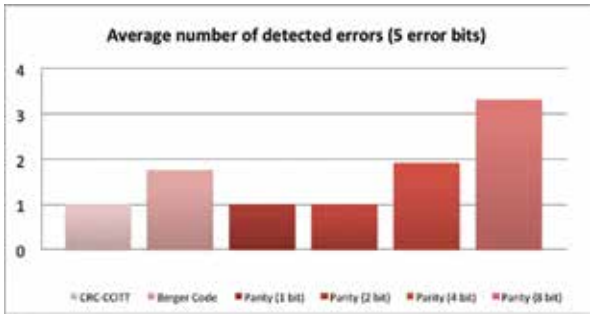
Fig. 2. INTERFACE OF SOFTWARE IMPLEMENTATION OF IBC CIPHER WITH ERROR DETECTION METHODS [9]

```
$ ./IBCipher.exe
Not enough parameters!
-i [Input File Name] -o [Output File Name] -k [Keys File Name] -e [Encryption St
rength Level] -m [ecb/cbc mode] --[enrypt/decrypt]
Optional (for testing): --cedtype [0/1/2/3] [--paritybits bits] --errortesting [
number of errors] --debug
CED Type: 0 - none
          1 - parity (1, 2,4 or 8 parity bits)
          2 - Berger Code
          3 - CRC-16-CCITT (16 bits)
Encryption Strength:  low      - 128 bits key with 8 cycle rounds
                      medium   - 256 bits key with 10 cycle rounds
                      high     - 320 bits key with 12 cycle rounds
                      veryhigh - 128 bits key with 14 cycle rounds
Example: IBCipher -i text.txt -o secret.txt -k key.txt -e low -m ecb --encrypt
```

Source: The authors.

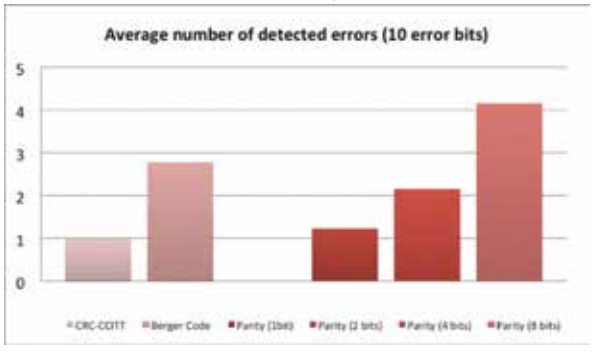
Three different error detection algorithms were implemented in the software IBC encryptor: parity check, Berger code and cyclic redundancy check (CRC). Parity checks if a number is even or odd – in this implementation refers to the evenness or oddness of a particular set of bits. The implementation contains four types of parity checks – depending on the amount of parity bits (i.e. 1, 2, 4 or 8 bits). A Berger code is an error-detecting code that computes the number of '0' (or '1') in the given set of bits. In the implementation every single bit of 32 byte data block is checked for being '0'. A cyclic redundancy check (CRC) is a popular error-detecting code based on polynomial division.

Fig 3. AVERAGE NUMBER OF DETECTED ERRORS FOR DIFFERENT METHODS (5 ERROR BITS WERE INSERTED DURING THE ENCRYPTION PROCESS)



Source: The authors.

Fig. 4. AVERAGE NUMBER OF DETECTED ERRORS FOR DIFFERENT METHODS (10 ERROR BITS WERE INSERTED DURING THE ENCRYPTION PROCESS)



Source: The authors.

Each implemented error detection method detects different amount of errors during the encryption process. The average number of detected faults for two different numbers of inserted errors (5 bits and 10 bits) is presented in Fig. 3 and Fig. 4. It was proved that CRC algorithm successfully detects any occurred fault. However it does not provide any information about the amount of errors. Parity check based on 1-bit has worse error detection success rate, because any even number of errors were not detected. Berger code is much better method than 1-bit and 2-bit parity: however for a small amount of errors, the parity check based on 4-bits is able to detect faults with better precision. Although, the parity check based on 8-bits provides the significant overhead, this method detects almost a half of provided errors in the encrypted data.

For the hardware implementation of IBC cipher with error detection methods, a Xillinx Spartan-3AN board –based on 90 nm technology – was chosen [10]. It has a relatively low cost and enough efficiency, which were the main reasons why this platform was chosen.

The Spartan-3AN board – presented in Fig. 5 – consists of a few basic programmable function elements: configurable logic blocks (build logic and storage elements), input/output blocks, RAM block (data storage), multiplier blocks, and Digital Clock Manager (DCM) blocks. The Spartan-3AN board consists of various ports and input/output controls like LCD, LEDs or switches. For IBC implementation the XC3S700AN version of Spartan-3AN board has been used. It is characterized by memory's parameters and I/O specifications:

- System gates: **700K**
- Logic cells: **13,248**
- Dedicated multipliers: **20**
- Block RAM blocks: **20**
- Block RAM bits: **360K**
- Distributed RAM bits: **92K**
- Flash size bits: **8M**
- User flash bits: **5M**
- Clock speed: **50MHz**

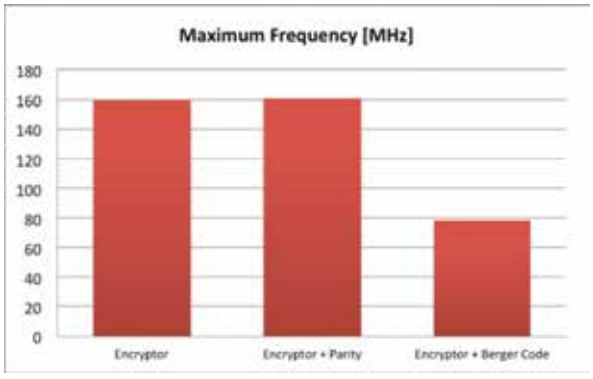
In the hardware implementation of IBC cipher Concurrent Error Detection (CED) functionality was also added. CED is a method of detecting errors while the algorithm is being processed. It allows verifying the results of encryption – if some random errors appear, the CED module will inform about this fact. Hardware encryptors should be equipped with CED methods, because some kind of faults in the internal structure of hardware encryptor may cause specific errors in

the encrypted cipher text. This may be used by potential attackers to give them some information about the key used in the encryption process or the encrypted message.

Fig. 5. XILINX SPARTAN 3AN DEVICE

Source: The authors.

Fig. 6. MAXIMUM FREQUENCY FOR HARDWARE ENCRYPTOR WITH AND WITHOUT ERROR DETECTION METHODS



Source: The authors.

Even a single internal error may cause multiple errors in the output data because of the specific functions used several times during the encryption process (i.e. substitutions and permutations).

Implemented CED methods provide some kind of redundancy, i.e. in hardware resources or operation time [9]. Therefore, proper CED design and implementation are crucial. In Fig. 6 the maximum frequency for three kinds of encryptors were shown: without any CED technique, with parity check and with Berger code. The parity check does not have any negative impact to frequency. However it requires more hardware elements: slices, BRAMs and Look-up Tables (LUTs) which build logic and storage elements. The implementation of Berger code requires even more hardware elements and significantly reduces maximum frequency. However, the maximum frequency for both CED methods do not exceed the clock speed of Spartan device (50 MHz), thus these two solutions can be used in practice.

4. HTTPS AND MUTUAL AUTHENTICATION PERFORMANCE EVALUATION

A key protocol of the INDECT security architecture is HTTPS, since most INDECT applications, including the INDECT web portal, provide a web interface and/or REST or SOAP ones, and HTTPS has been specifically designed to secure this kind of web sessions. Actually, HTTPS just refers to the hypertext transport protocol (HTTP) running on top of the transport layer security/secure sockets layer (TLS/SSL) protocol, which is the one encrypting and protecting the integrity of communications.

However these extra security mechanisms do not come without costs. The performance impact of HTTPS has been studied along time in the literature. In 1999, Apostolopoulos et al. [11] reported that serving web pages over TLS/SSL was two orders of magnitude slower than regular ones, and increased the latency above 300 ms, mainly due to the TLS handshake protocol. Later, in 2006, Coarfa et al. [12] reported that the TLS/SSL impact was reduced to less than one order of magnitude, and was mainly due to RSA decryption of master key. More recently, it has been reported [13] that deploying TLS/SSL may only incur in 12-40% penalty degradation thanks to session reuse. Therefore, the technological evolution has greatly improved the deployment of TLS/SSL in web servers, and thus seems ready to be employed in secure architectures.

However the INDECT security architecture employs one little-known feature of TLS/SSL: mutual authentication, which has not been considered in the mentioned performance studies. With mutual authentication, both the server and the client provide their certificates, so both are mutually authenticated, instead of only the server as in regular TLS/SSL sessions. This way, users can authentica-

te against any web-based INDECT service automatically by using the certificates stored in their smart cards, instead of typing a username and password.

In order to evaluate the performance penalty of client authentication in TLS/SSL, we have deployed a small testbed composed by an Apache 2.2 web server running in a Dell PowerEdge 1950 server (4x Intel Xeon CPU E5420 @2.50GHz, 8GB RAM, Linux 2.6.32-64 bits) and the Apache JMeter 2.11 benchmark tool running on a Dell Latitude E6330 laptop (Intel Core i5-3320M CPU @2.60GHz, 8GB RAM, Linux 3.8.13-64 bits) interconnected with a Fast Ethernet LAN. Table II shows the results of the performance tests, where a small static HTML page (177 bytes) was downloaded 10,000 times consecutively by 16 parallel threads, either using HTTP, HTTPS with server authentication or HTTPS with mutual authentication. To study the effect of the full TLS/SSL handshake protocol, session reuse was enabled/disabled in all scenarios. The server certificate (2048 bit RSA key) has an associated chain with two CAs (4096 bit RSA keys), while the client certificate (2048 bit RSA key) chain has a single CA (4096 bit RSA key).

The test results in Table II show that TLS/SSL still has some impact on the performance of web servers nowadays (i.e. 39% throughput penalty between HTTP and HTTPS with session reuse). As previous works in the literature have already identified, the main source of overhead is the TLS handshake protocol and the required public key cryptography operations. Disabling session reuse reduces the performance of a HTTPS web server by 60% and increases the latency 2.65 times. However, once TLS/SSL is enabled, the further effect of mutual authentication is minimal both in latency and throughput, either with or without session reuse.

Therefore these results show that it is feasible to employ TLS/SSL to protect web traffic within the INDECT security architecture, as well as to employ client certificates to implement a secure user authentication mechanism.

5. VIDEO STREAMING OVER OPENVPN PERFORMANCE EVALUATION

The INDECT block cipher (IBC), which was described in Section 2.1, has also been implemented into the OpenSSL 0.9.8v library. Since OpenVPN software package uses OpenSSL to perform all cryptographic operations, it is also possible to protect INDECT VPN infrastructure with IBC. A OpenVPN package that uses the modified OpenSSL 0.9.8v cryptographic library supports the following IBC cipher modes: INDECT-128-CBC, INDECT-192-CBC, and INDECT-320-CBC supporting 128, 192 and 320 bit long keys with cipher-block chaining (CBC) mode of operation.

One of the scenarios where OpenVPN can be employed is to secure the communications between a CCTV camera and its INDECT application server. Because video streaming requires high throughput and low delay, it is necessary to carry out a performance test to confirm the applicability of OpenVPN with IBC for this scenario.

To analyze the performance of OpenVPN cryptographic operations, a benchmark using iperf and ping tools has been performed. iperf has been used to measure the throughput and ping to measure the delay. Two directly connected desktop computers with Ubuntu 10.04 LTS operating system, a Pentium 4 processor running at 3.06 GHz and Gigabit Ethernet network interface cards were used in this test. The measured results of the throughput and delay are shown in Table III. In addition to the three IBC cipher variants, other cipher algorithms were also tested for comparison purposes. TCP window size was set to 85.3 KB during all measurements. It is also worth mentioning that OpenVPN uses LZO (Lempel-Ziv-Oberhumer) compression to reduce the amount of transmitted traffic.

The results of these measurements in Table III show that the performance of the new IBC cipher is significantly worse than the other more mature ciphers. This is due to fact that the IBC code is not yet optimized, especially when compared with the AES cipher that is the most popular symmetric cipher nowadays, and subject of continuous optimizations in past years. However, the measured throughput for all IBC ciphers (more than 68 Mbps with a software implementation) is sufficient for the transmission of high quality video. This assumption was confirmed by a test, in which a video stream was transported from the IP camera (D-Link DCS-2100+) to the computer via the OpenVPN tunnel. A more thorough test would require transmission of several video streams from multiple cameras to the destination in the OpenVPN server.

TABLE II
PERFORMANCE OF MUTUAL AUTHENTICATION IN TLS/SSL

	HTTP	HTTPS -client +reuse	HTTPS -client -reuse	HTTPS +client +reuse	HTTPS +client -reuse
Throughput (op/s)	561,37	341,49	133,71	350,32	116,01
Avg. latency (ms)	25	43	114	41	133

Source: The authors.

TABLE III
RESULTS OF OPENVPN THROUGHPUT AND DELAY TESTS

Cipher	Throughput (Mbit/s)	Delay (ms)
INDECT-128-CBC	78.3	0.154
INDECT-192-CBC	72.6	0.156
INDECT-320-CBC	68.7	0.153
AES-128-CBC	125	0.153
AES-192-CBC	123	0.156
AES-256-CBC	119	0.155
DES-CBC	126	0.153
DES-EDE-CBC	116	0.155
DES-EDE3-CBC	118	0.158
DESX-CBC	121	0.153
IDEA-CBC	122	0.151
RC2-CBC	119	0.157
RC2-40-CBC	120	0.157
RC2-64-CBC	124	0.152
BF-CBC	124	0.152
CAST5-CBC	123	0.156

Source: The authors.

6. CONCLUSION

This paper has presented a performance evaluation of three important components of the INDECT security architecture.

Firstly, the performance evaluation of three encryptors' error detection methods – parity check, Berger code, and cyclic redundancy check – was performed. Further, the performance of IBC cipher hardware implementation with concurrent error detection functionality was tested.

Secondly, the performance penalty of TLS/SSL with mutual authentication was assessed, which is used by the HTTPS protocol, to protect the web-based communications of most INDECT systems, including the INDECT web portal.

Thirdly, the throughput and delay of traffic, which was transmitted via the OpenVPN tunnel with the IBC cipher, were measured.

All these performed tests confirmed that, although the proposed security tools and mechanisms in the INDECT security architecture could have a negative influence on the performance of the communication, this performance penalty is not too high to prevent their use in LEA systems and networks.

ACKNOWLEDGMENTS

This work has been funded by the EU Project INDECT (Intelligent information system supporting observation, searching and detection for security of citizens in urban environment) — grant agreement number: 218086, and the project No. CZ.1.07/2.3.00/20.0217 (The Development of Excellence of the Telecommunication Research Team in Relation to International Cooperation) within the frame of the operation programme "Education for competitiveness" financed by the European Structural Funds and from the state budget of the Czech Republic.

REFERENCES

[1] INDECT Project, http://www.indect-project.eu

[2] M. Urueña, P. Machník, M. Niemiec, N. Stoianov, "INDECT Security Architecture," *Multimedia Communications, Services and Security*, CCIS, vol. 368, pp. 273-287, 2013. Heidelberg: Springer

[3] M. Niemiec, L. Machowski, "A new symmetric block cipher based on key-dependent S-boxes," International Congress on Ultra-Modern Telecommunications and Control Systems, ICUMT 2012, pp. 474-478, Saint Petersburg, 2012.

[4] N. Stoianov, M. Urueña, M. Niemiec, P. Machník, G. Maestro, "Integrated security infrastructures for law enforcement agencies," *Multimedia Tools and Applications*, vol. 74, pp. 4453-4468, 2015. Springer.

[5] C. Adams, S. Lloyd, *Understanding PKI: Concepts, Standards, and Deployment Considerations*, 2nd ed., Addison Wesley, 2002.
[6] OpenVPN, http://openvpn.net/index.php/open-source.html
[7] StrongSwan, http://www.strongswan.org
[8] N. Stoianov, M. Urueña, M. Niemiec, P. Machník, G. Maestro, "Security Infrastructures: Towards the INDECT System Security," *Multimedia Communications, Services and Security*. CCIS, vol. 287, pp. 304-315, 2012. Heidelberg: Springer.
[9] INDECT Consortium. *D9.44: New methods of error detection*, February, 2014.
[10] M. Niemiec, J. Dudek, L. Romański, M. Święty, "Towards hardware implementation of INDECT Block Cipher," *Multimedia Communications, Services and Security*. CCIS, vol. 287, pp. 252-261, 2012. Heidelberg: Springer.
[11] G. Apostolopoulos, V. Peris, D. Saha, "Transport layer security: how much does it really cost?" Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99), vol. 2, pp. 717-725, New York, 1999.
[12] C. Coarfa, P. Druschel, DS Wallach, "Performance Analysis of TLS Web Servers," *ACM Transactions on Computer Systems*, vol. 24, no. 1, pp. 39-69, 2006.
[13] H. Kleppe, "Performance impact of deploying HTTPS," Technical Report. Universiteit van Amsterdam, 2011.