Stoianov, N., Urueña, M., Niemiec. M., Machnik, P. & Maestro, G. (2015). Integrated security infrastructures for law enforcement agencies. *Multimedia Tools and Applications*, 74 (12), pp. 4453–4468.

DOI: 10.1007/s11042-013-1532-7

Integrated security infrastructures for law enforcement agencies

Nikolai Stoianov - Manuel Urueña - Marcin Niemiec
Petr Machnik - Gema Maestro

This paper provides an overview of the security architecture for Law Enforcement Agencies (LEAs) designed by the INDECT project, and in particular the security infrastructures that have been deployed so far. These security infrastructures can be organized in the following main areas: Public Key Infrastructure (PKI) and user management, communications security, and new cryptographic algorithms. This paper presents the new ideas, architectures and deployed testbeds for these areas. In particular, it explains the inner structure of the INDECT PKI employed for federated identity management, the different technologies employed in the VPN testbed, the INDECT Block Cipher (IBC) – a novel cryptographic algorithm that has being integrated into OpenSSL library, and how IBC-enabled TLS/SSL sessions and X.509 certificates are employed to protect INDECT applications. All proposed mechanisms have been designed to work in an integrated fashion as the security foundation of all systems being developed by the INDECT project for LEAs.

Keywords
Law enforcement agency (LEA)
Public key infrastructure (PKI)
Virtual private network (VPN)
INDECT block cipher (IBC)
X.509 certificates
Smart card (SC)
Federated identity management
Transport layer security (TLS)

This paper is an improved version of "Security Infrastructures: Towards the INDECT System Security" from the same authors, presented in the 5th International Conference on Multimedia Communication Services & Security (MCSS 2012), Krakow (Poland), May 31- June 1, 2012.

N. Stoianov (✉)
Technical University of Sofia, INDECT Project Team, 8, KlimentOhridski St., 1000 Sofia, Bulgaria
e-mail: nkl_stnv@tu-sofia.bg

G. Maestro
APIF Moviquity SA Madrid, Madrid, Spain
e-mail: gmm@moviquity.com

M. Niemiec
AGH University of Science and Technology,
Mickiewicza 30 Ave., 30-059 Krakow, Poland
e-mail: niemiec@kt.agh.edu.pl

M. Urueña
Department of Telematic Engineering, Universidad Carlos III de Madrid,
Avda.de la Universidad, 30, 28911 Leganes, Madrid, Spain
e-mail: muruenya@it.uc3m.es

P. Machnik
Department of Telecommunications, VSB-Technical University of Ostrava,
17.Listopadu 15, 708 33 Ostrava, Czech Republic
e-mail: petr.machnik@vsb.cz

1 Introduction

Nowadays the requirements of any ICT (Information and Communication Technologies) system regarding data protection and information security are constantly increasing. The expectations of Law Enforcement Agencies (LEAs) from their ICT systems are even higher, given that the security of their citizens is at stake. INDECT (*Intelligent information system supporting observation, searching and detection for security of citizens in urban environment*) [5] is a collaborative research project funded by the 7th EU Framework Program whose objective is to develop advanced tools for Law Enforcement Agencies. In particular the Work Package 8 (WP8) of the INDECT project is focused on increasing the security of the information stored, exchanged or accessed by INDECT systems and users.

One of the main characteristics of today's ICT is system interconnection. Based on different protocols and standards, LEA systems exchange a lot of information related to users, citizens, suspects, criminals, system status, patrol vehicles, etc. All these data are sent and received in different ways and stored in dedicated systems. This huge amount of information raises two main problems: the management of the so-called "big data", and the security of this information. For assuring confidentiality, integrity, accessibility, access control and non-repudiation it is necessary to employ different methods and techniques - from technical to organizational ones. For a complex ecosystem the first step to create a secure environment is to define the security foundations of such a system.

2 INDECT security architecture

Figure 1 shows a simplified view of the integrated security architecture for Law Enforcement Agencies being designed by the INDECT project.

The main components [11] of the proposed security architecture are:

- Public Key Infrastructure (PKI)—to issue, mange, store and revoke X.509 certificates used in the system. They are issued to all users and ICT systems to authenticate them as well as to secure their communications.
- LDAP User Directory—to store 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 the user's clearance level or the applications she can access to.
- Audit Server—all relevant user actions (e.g. accessing an application or requesting classified information) is logged both locally and in the secure centralized system. The logs are constantly being reviewed by LEA auditors to detect suspicious behaviors.
- INDECT Portal—the homepage of LEA users. It allows them to access the different services and applications available to them, according to particular scenarios (e.g. during a crisis). The INDECT portal also acts as the Identity Provider (IdP) for all INDECT federated systems.
- INDECT Application Servers—execute the different applications, services and tools being developed by the INDECT project. They act as Federated Service Providers (SP), authenticating the users through the INDECT Portal (IdP), although they may handle application-specific user's authorization attributes (e.g. which CCTV cameras a given user may access to). Most applications provide a web-based interface, and most services are also web-based,

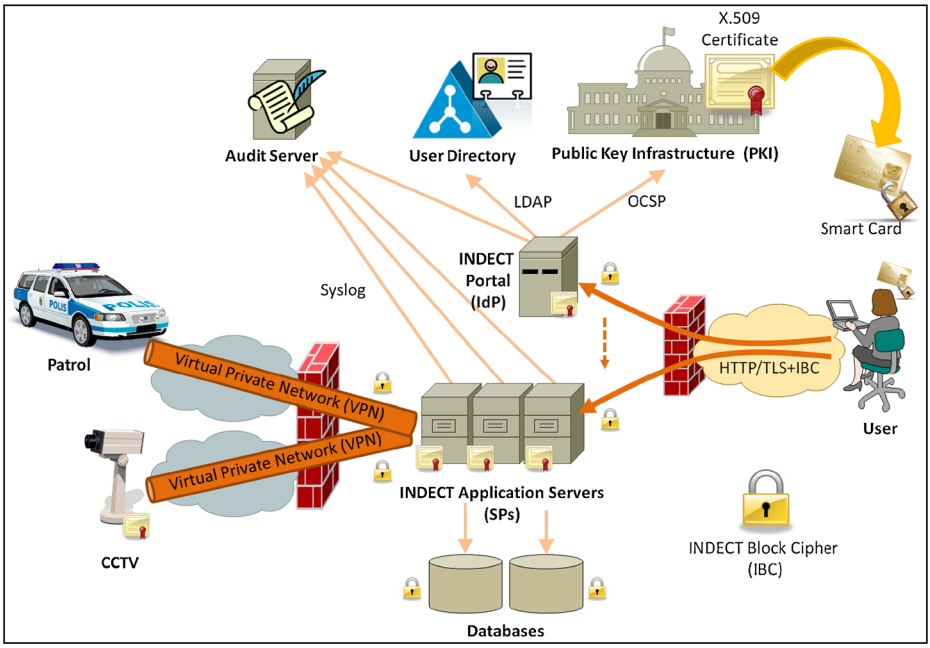

Fig. 1 INDECT security architecture

implementing SOAP or REST interfaces. Therefore SSL/TLS is employed for secure communications with users as well as among themselves.

- INDECT Databases—although stored deep inside the LEA data center, they should communicate in a secure way with IDECT application servers and being encrypted, for by instance using the novel INDECT Block Cipher (IBC) presented later.
- Virtual Private Network (VPN)—protect the communications with external LEA users and devices. Only encrypted traffic is allowed to go through the LEA Data Center firewalls, which block all external traffic by default and should feature additional security mechanisms such as Intrusion detection Systems (IDS).
- Smart Cards—storing users' certificates are issued by the INDECT PKI and used for access control by the central INDECT web portal, as well as encrypting and signing e-mails and documents.

The following sections study in more detail the different security technologies and protocols employed inside this integrated architecture.

It is worth noticing that, in order to guarantee the robustness of the security architecture and a wide support by applications, standard security protocols like TLS/SSL or IPSec have been preferred to custom ones. Nonetheless the INDECT security architecture also includes novel mechanisms such as the new INDECT Block Cipher (IBC) that may be employed to encrypt TLS/SSL sessions and VPN tunnels.

3 INDECT public key infrastructure

One of the main characteristics of the 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 [1]. 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. The INDECT PKI architecture has a hierarchical, two-level structure:

- Level I—Only the Root Certification Authority (Root CA) operates at this level. This CA is always offline to prevent attacks to the root key of the INDECT PKI.
- Level II—There are two CAs at this level: one for issuing certificates to users (Users' CA), and other CA for issuing certificates to devices (Devices CA). A trusted connection is established between these two CAs. Thus, the Root CA only issues certificates to these two CAs (Users and Devices CAs).

Figure 2 shows the different levels of the INDECT PKI and the relationship between the different CAs.

The Users CA manages (create, issue, revoke etc.) all the certificates related to INDECT users. Users may employ these certificates to log into the INDECT web portals, sign documents or encrypt connections and e-mails. These X.509 certificates are securely stored in smart-cards as described in the next section.

The Devices CA manages all aspects of certificates issued to devices (PCs, PDAs, CCTVs, etc.). Each X.509 certificate is assigned to a specific device, thus each device can be uniquely identified and managed based on its certificate. Devices' certificates are used for creating secure communication channels (e.g. TLS/SSL or SSH), for signing data streams and exchanging documents, and for authentication to avoid man-in-the-middle attacks.

Table 1 shows the appropriate key sizes for the proposed CA's and the certificates they issue.

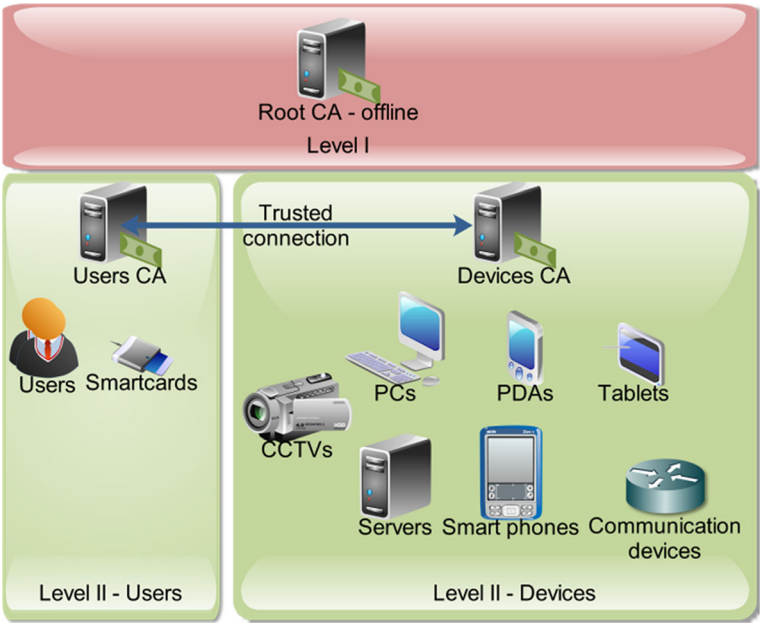

Fig. 2 Sample presentation of PKI infrastructure

In order to test the feasibility of the proposed INDECT PKI, a testbed first based on OpenCA [8] and EJBCA [3] has been deployed[1]. A sample user certificate issued by this PKI is shown below:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4 (0x4)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: emailAddress=nkl_stnv@tu-sofia.bg,CN=Nikolai
Stoianov,OU=WP8,O=INDECT Test CA,C=EU
        Validity
            Not Before: Apr 11 02:08:33 2013 GMT
            Not After : Apr 10 02:08:33 2015 GMT
        Subject: serialNumber=4,CN=Nikolai Stoianov,OU=Technical University of
Sofia,O=INDECT Test CA,C=BG
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:ea:7f:24:5b:a7:7f:e2:36:f6...
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Certificate Policies:
                Policy: 1.2.3.3.4
                  CPS: http://www.indect-project.eu/cps
            Netscape Cert Type:
                SSL Client, S/MIME
            X509v3 Key Usage:
                Digital Signature, Non Repudiation, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, E-mail Protection, Microsoft
Smartcardlogin
            Netscape Comment:
                User Certificate of INDECT Test CA
            X509v3 Subject Key Identifier:
                B2:6A:DA:...
            X509v3 Authority Key Identifier:
                keyid:BC:10:...
                DirName:/C=EU/O=INDECT Test CA/OU=WP8/CN=Nikolai
Stoianov/emailAddress=nkl_stnv@tu-sofia.bg
                serial:00
            X509v3 Subject Alternative Name:
                email:nkl_stnv@tu-sofia.bg
            X509v3 Issuer Alternative Name:
                email:nkl_stnv@tu-sofia.bg
            Netscape CA Revocation Url:
                http://www.indect-project.eu/pub/crl/cacrl.crl
            Netscape Revocation Url:
                http://www.indect-project.eu/pub/crl/cacrl.crl
            X509v3 CRL Distribution Points:
                URI:http://www.indect-project.eu/pub/crl/cacrl.crl
    Signature Algorithm: sha1WithRSAEncryption
        83:87:57:...
```

[1] Initially we planned to employ OpenCA for the proposed INDECT Public Key Infrastructure (PKI). However we stopped using OpenCA because of the key length limitations (up to 4096 bits). It is possible to issue keys with length up to 8192 bits by using EJBCA which covers INDECT CA requirements.

Table 1 Suggested size of the private RSA keys

CA role	Key length
Root CA	8192 bits
User CA	4096 bits
Device CA	4096 bits
User certificate	2048 bits
Device certificate	1024 bits

Currently the work is focused on creating a fully functional infrastructure based on EJBCA, implementing specific extensions for X.509 certificates, and finally to employ the certificates issued by the PKI in other INDECT systems and in the remaining security infrastructures.

4 Federated identity based on certificates and smartcards

In a Federated Identity scenario, different Service Providers (SPs) delegate the identity management of these users to a centralized entity referred as Identity Provider (IdP) within a controlled environment called Circle of Trust (CoT). This concept has been the starting point for the user management solution implemented for the INDECT project. The system formed by (at least) one IdP and (at least) one SP is called a Federation, and it is characterized by having a trust relationship among its members, simplifying data communication and validation of the user in a secure way.

Usually Federated Identity solutions are focused on managing and protecting end user's information employed by different services [13]. However the main emphasis of INDECT security framework must be on providing a controlled and secure access to the different INDECT services. To maximize the compliance of these requirements so as to meet this scenario, the Federated Identity paradigm has been extrapolated to a secure model that also relays on a Public Key Infrastructure (PKI) for two-factor authentication.

Within INDECT, the IdP role is played by the INDECT web portal, although part of the user management is also performed by the PKI that issues INDECT certificates. This way they are able to guarantee the identities of all INDECT elements: users, devices and servers, pointing out their unique identifier and attributes as well as managing their validity. The CA is then a trustworthy element in which all other INDECT elements have to trust in. Based on certificates, any exchange is carried out between authenticated and trustworthy elements through secure channels as explained in next section.

The certificate generation, and therefore the PKI, is linked to the user registration process. When a user is registered into the system, the User CA creates the corresponding certificate, including part of the information provided in the registration process. The information stored in the certificate should be relevant for the user management process and must be not updated frequently (i.e. the user identifier).

This X.509 certificate enables checking the identity of the user. When a user accesses any INDECT systems through the INDECT Portal, its certificate is required by means of TLS/SSL. The authentication solution is able to check the validity of the certification with the corresponding CA and process its data, informing the INDECT Service Provider about the users' identity and attributes (i.e. role).

The access control solution is then composed by four main elements:

- INDECT Security Local Tools, which allow to generate, a key pair (private and public) as the previous step to the generation of a certificate, and on the other hand, the certificate container (PKCS#12), necessary to load the certificate in a smart card or in the browser.
- INDECT PKI Certification Authority, which issues the X509 users', service providers' and access devices' certificates and manages their validity. The certificates, through their extensions, contain the information that has been considered relevant for the access control process, such as the security clearance of the user.
- A Certification Holder, is where the certificates are stored, like a Smart Card.
- Authentication Filter Libraries, that, once integrated in the INDECT application servers (Service Providers), will allow INDECT applications to handle the user authentication information, such as validating the certificate and recovering any certificate information, like user's identity or any other extension.

This design also supports legacy LEA applications employed by users without a certificate. In this case, a login and password will be required to access these low-security applications. These user credentials are stored in the LDAP User Directory to ease its management. Moreover, depending on the levels associated to the users' attributes, access to high-security applications may require both a valid certificate and an additional password, thus enabling multi-factor authentication.

Based on the user's attributes (contained in the certificate and in the LDAP repository) the general users' rights are built. According to this information, the user will access the INDECT Portal dashboard through which the authorized INDECT services will be accessible.

5 Communications security

A major challenge is how to protect in a secure manner the diverse set of applications and systems being developed by the INDECT project without designing a specific security mechanism for each system. The main design insight is that most networked[2] INDECT applications have got either a web interface or are based on web services, although other systems employ a completely different set of protocols and only have in common the fact that they run on top of the Internet Protocol (IP).

Therefore we should start studying standard security mechanism for these two different sets of applications. On the one hand, the security of INDECT web-based applications and web services is based on the so called "Secure HTTP" (HTTPS). On the other hand, applications based on protocols different to HTTP or remote systems running outside the security of the Data Center protect their communications by means of Virtual Private Networks (VPNs).

5.1 Virtual private networks (VPNs)

One of the main components of the secure communication infrastructure within INDECT system is a Virtual Private Network (VPN) framework that will enable the secure communication among

[2] The INDECT project is also developing standalone, non-networked applications. We won't consider it security here, since its usage is confined to particular systems and the information that can be disclosed by a security breach is limited to the local data of the application.

multiple remote nodes and servers interconnected over public networks. Nowadays virtual private networks are usually based on two different technologies: IPSec and SSL.

For the implementation of VPN infrastructures in the INDECT system, only open-source solutions will be used due to its flexibility and proven robustness. In particular, the StrongSwan software package seems to be a convenient open-source IPSec VPN solution. StrongSwan is intended primarily for devices using Linux, although it is fully compatible with other standard IPSec VPN implementations, and thus can be used in networks with mixed equipment.

As an open-source SSL VPN solution, the best option appears to be the OpenVPN software package. OpenVPN can be installed in computers with major operating systems, and it is a very flexible and scalable VPN package. For example, it works nicely with Network Address Translators (NAT) in contrast to IPSec VPNs. On the other hand, it has compatibility problems with VPN solutions from other vendors.

Both VPN packages, StrongSwan and OpenVPN, support PKI and authentication based on X.509 certificates. Each VPN client obtains a certificate from a certification authority, which is subsequently used to authenticate the client when a secure tunnel has to be created between the client and the VPN gateway. To support Authentication, Authorization and Auditing (AAA) services, an additional LDAP/RADIUS server can be employed, which must be located inside the private network.

Within the INDECT system, users will employ mainly OpenVPN to securely communicate between their terminals (desktop, laptop, PDA, smart phone, etc.) and servers located in the Police headquarters. The INDECT Devices CA will authenticate the individual terminals.

5.2 Security and mutual authentication of INDECT web applications

The secure version of the Hypertext Transfer Protocol (HTTP) employed by web applications is commonly known as “HTTPS” since this is the protocol name that appears at the beginning of the Uniform Resource Locator (URL) of secure web sites. However “HTTPS” is not a new protocol itself, but specifies that HTTP protocol runs on top of a secure session protocol. This secure protocol is called TLS/SSL, and it is the foundation of the common security mechanism of web-based INDECT applications.

The Transport Layer Security (TLS) [2] and its predecessor, and probably more popular, Secure Sockets Layer (SSL) [4], are client–server protocols that provide communications security on top of the Transmission Control Protocol (TCP). Although SSL was originally designed for the web, they are application-agnostic, meaning that any application protocol running on top of TCP may run on top of TLS/SSL. It is an advanced security protocol featuring symmetric-cryptography encryption, asymmetric-cryptography key exchange, end-point authentication based on X.509 certificates, and integrity protection by means of message authentication codes. Moreover, TLS/SSL is an extensible protocol since peers are able to negotiate which version of the protocol and what cipher suite (e.g. TLS_RSA_WITH_INDECT_320_CBC_SHA) will be used during the communication session. However it is worth noting that TLS/SSL does not provide digital signature or non-repudiation services, thus these security mechanisms must be implemented by the applications that require it.

Usually in a web TLS/SSL session only the server is authenticated, that is, it is the web server who send its X.509 certificate to the client. After validating the certificate (i.e. checking server’s name, the expiration date, the whole certificate chain, the revocation list, etc.) the client encrypts the session key exchange message with the public key of the certificate, thus the communication can only progress if the server has the private key. This way, web browsers can check that they are actually communicating with the intended web server (e.g. the original bank website instead of a *phising* clone), thwarting all kinds of man-

in-the-middle attacks. However INDECT web applications will also authenticate the client of the TLS/SSL session by means of the X.509 certificate stored in the Smart Card of the user. Figure 3 details the setup of a TLS/SSL session with mutual authentication.

The advantage of using TLS/SSL for INDECT web applications is that it is already implemented by all major web browsers and web servers, and it is enforced even before the application is called. Therefore, when TLS/SSL is properly configured in the server, the application is certain that the client has been authenticated by means of the X.509 certificate stored in a pin-protected Smart Card. Thus, it provides from the start a two-factor authentication service based on a "something you have" and/or "something you know" credentials.

From a cryptographic point of view, and after several revisions, TLS version 1.2 is considered a secure protocol, although many attacks have been proposed against practical details of its implementation. Recently the security breaches of some trusted Certificate Authorities, including the issue of fraudulent server certificates, have called into question the security of PKI, and thus the TLS/SSL authentication security. However, we argue that these attacks do not pose a threat to the use of TLS/SSL by INDECT systems, since secure terminals do not trust other Certification Authorities (CAs) than INDECT's one. Moreover INDECT web servers do only request and accept client certificates issued by the INDECT Users' CA. Therefore, even if the certificate of an INDECT device (e.g. a node station) is compromised, it cannot be employed neither to supplant an application server, nor a user.

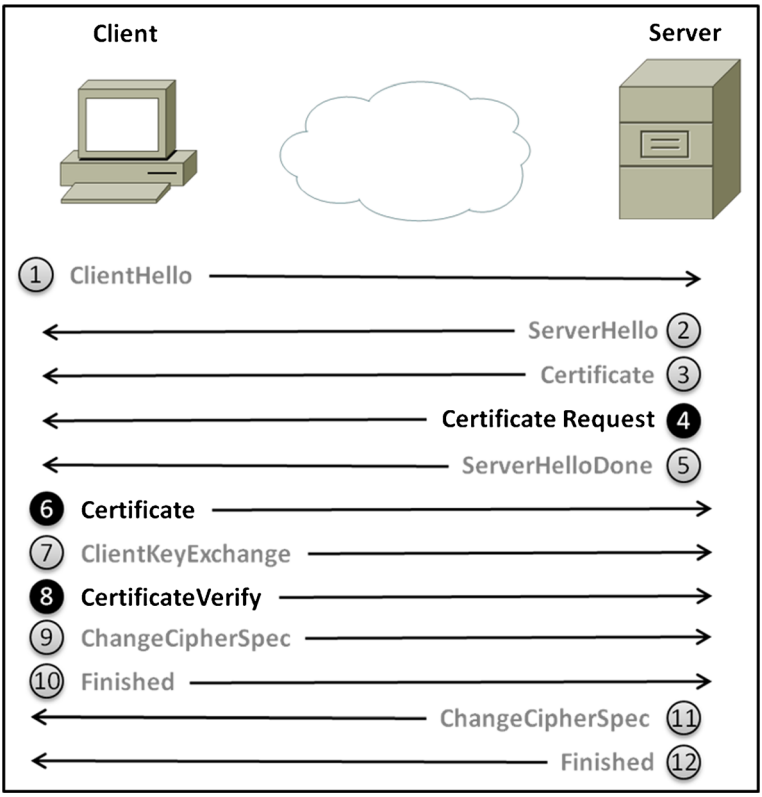

Fig. 3 TLS/SSL secure session setup with mutual authentication [12]

6 INDECT block cipher's basics

One of the research activities performed by INDECT project refers to modern cryptographic techniques. This includes developing new algorithms and protocols which ensure high-level of data confidentiality. New algorithms are evaluated by means of proper simulators and tested to check the resistance on several attacks.

The INDECT Block Cipher (IBC) [7], is a novel algorithm to encrypt data, that meets the stringent confidentiality requirements of Law Enforcement Agencies. The cipher transforms a message in order to make it unreadable to anyone except some proper entities (i.e., the sender and recipient). It is a symmetric block cipher, so both encryption and decryption processes transform a message by means of the same key (secret key).

The proposed block cipher is based on two functions: substitution matrix (S-Box) and permutation. These operations are used in each round of the cipher. During each single round, the 256-bit block of plaintext/ciphertext is divided into 64 sub-blocks, 8-bit each. In the next step each sub-block is passed to the substitution box as the input value. Output values are concatenated into one 256-bit block (the merging method is adequate to the previous division schema). The last step is the use of a permutation function (based on S-Box) on the 256-bit block of data. The innovation of this cipher is based on the idea of "basic functions" and its linear combinations. This concept, allows us to modify the structure of cipher by means of a key and use huge number of non-linear S-boxes. Below, only the major features of IBC cipher are presented:

- A substitution-permutation structure
- The cipher architecture depends on the key
- Uses a huge number of non-linear S-boxes (about 5,35*1018)
- Block size: 256 bit
- Key lengths: 128, 192, 320, 576 bit
- Number of rounds: 8, 10, 12, 14

The IBC algorithm was initially tested by means of a new simulator. The simulator checks the main security features of the cipher, which decide the strength of cryptographic algorithm. All simulations confirmed that IBC ensures the high-level of data confidentiality. The following features were tested:

- Balancing,
- Non-linearity,
- Strict Avalanche Criterion (SAC),
- Completeness,
- Diffusion order, and
- Structure of XOR table.

The functionality of the new cipher was later verified by means of software and hardware implementations [6]. The C++ programming language was chosen for the software implementation of the algorithm itself, whereas the Graphical User Interface (GUI) was built by means of the C++/CLI language under the .NET platform. The visual interface of the IBC application showing an example of the encryption process is presented in Fig. 4. Additionally, the functionality of the IBC cipher was implemented in a Xilinx Spartan FPGA.

The next step towards the practical usage of IBC has been its integration with popular security libraries such as OpenSSL and OpenVPN. This integration greatly simplifies using

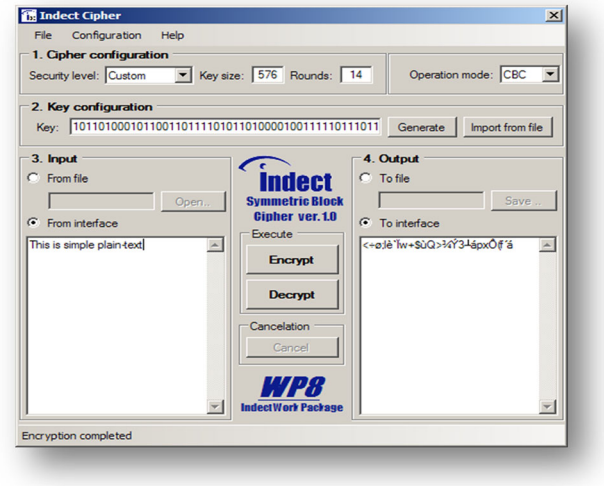

Fig. 4 The INDECT Block Cipher (IBC) application with an example of the encryption process

the developed algorithms in practice. By means of such integration in the OpenSSL library, the IBC cipher can be used to secure different INDECT subsystems as well as other applications outside the project.

The integration consisted of two steps: the implementation of the new cipher algorithm in the `libcrypto` library and the modification of the OpenSSL code to make the new cipher fully available in all OpenSSL utilities and in SSL/TLS connections. During the integration of the IBC cipher within the OpenSSL environment, many crucial files of the OpenSSL library were modified (i.e.: `apps/progs.h`, `apps/speed.c`, different files from `crypto/evp` and `crypto/objects` directories). Also, some new files which contain definition of IBC's functions and the code performing actual encryption and decryption were added.

The most important feature of integrating the IBC cipher in OpenSSL was to easy use IBC in existing applications (just by replacing the existing shared library files with the modified ones). This is possible when the binary compatibility is ensured between the original and modified libraries. By being aware of this requirement, now it is possible to deploy IBC on existing systems without changing the code of applications using OpenSSL or even without recompiling. The binary compatibility has been tested by the ABI compliance checker (an open source tool available for Linux operating systems).

6.1 OpenSSL implementation and testing results

In particular the INDECT Block Cipher, has been implemented into the *OpenSSL 0.9.8v software package*. OpenSSL Project [9] is a collaborative effort to develop a robust, commercial-grade, full-featured and open source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS) protocols as well as a full-strength general purpose cryptography library. The core library of OpenSSL is written in the C programming language and implements the basic cryptographic functions and provides various utility functions. OpenSSL can be used in many operating systems—Solaris, Linux, MAC OS X, open source BSD, OpenVMS, and Microsoft Windows.

The modified OpenSSL 0.9.8v supports the following IBC cipher modes—INDECT-128-CBC, INDECT-128-ECB, INDECT-192-CBC, INDECT-192-ECB, INDECT-320-CBC, and INDECT-320-ECB, supporting 128, 192 and 320 bit long keys with Cipher-Block Chaining (CBC) and Electronic Codebook (ECB) modes of operation.

To analyze the performance of OpenSSL cryptographic operations, a benchmark was performed using the *openssl speed* command. This test was carried out by a plain desktop computer with Ubuntu 12.04 LTS operating system and Intel Celeron processor running at 2.8 GHz. The measured results can be seen in Table 2. For comparison, it contains values for six variations of IBC cipher (modes and key length), six variations of AES cipher, two IDEA cipher's mode of operation and two CAST cipher's mode of operation. All values in the table are measured in kilobytes per second processed when encrypting a block of data. The size of the data block is listed in column headers.

Obviously, the measurements 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 implemented optimally, especially when compared with the AES cipher that is the most popular symmetric cipher nowadays and subject of continuous optimizations in the past years. Although IBC already has a reasonable performance for the communication needs of current applications (e.g. 25 Mbps), improving the performance of the IBC cipher is the next goal of our research in this area.

Besides these raw cipher benchmarks, the modified OpenSSL library has also been tested to check that IBC is also supported in TLS/SSL sessions. Each IBC cipher suite has been assigned an identifier in the `0x00FFxx` range (e.g. `DHE-RSA-INDECT320-SHA` is encoded as `0x00FF88`) to be negotiated in SSL sessions. Actually IBC with a 320 bits key is usually selected by default as the cipher to encrypt TLS/SSL sessions because OpenSSL prioritizes longer keys. However, when one of the peers does not support IBC,

Table 2 Results of cipher performance benchmark (in kilobytes per second)

Cipher	Block size 16 bytes	Block size 64 bytes	Block size 256 bytes	Block size 1024 bytes	Block size 8192 bytes
INDECT-128-CBC	4504	4541	4611	4605	4618
INDECT-128-ECB	4510	4536	4561	4575	4544
INDECT-192-CBC	3635	3654	3654	3663	3674
INDECT-192-ECB	3644	3683	3698	3672	3697
INDECT-320-CBC	3084	3097	3113	3113	3102
INDECT-320-ECB	3092	3118	3136	3124	3128
AES-128-CBC	45844	84438	109087	118111	117976
AES-128-ECB	95633	103068	106600	105506	105274
AES-192-CBC	42558	74902	93568	100929	100518
AES-192-ECB	83758	89517	92369	92161	91335
AES-256-CBC	39406	66725	82262	87411	88030
AES-256-ECB	73849	78735	80207	80800	80694
IDEA-CBC	31688	34899	36428	36279	37147
IDEA-ECB	32754	35358	35659	35449	35847
CAST5-CBC	39042	44645	46394	46419	47463
CAST5-ECB	37203	38195	38352	39078	39058

a standard cipher like AES256 is selected instead, therefore being fully compatible with other TLS/SSL implementations.

Regarding to the use of IBC for protecting VPN tunnels, OpenVPN [10] uses OpenSSL to perform all cryptographic operations, therefore it should be also possible to protect the INDECT VPN infrastructure with IBC, although this has not been tested yet. The performance of IBC-enabled OpenVPN clients and servers would mainly depend on the performance of encryption of the outgoing traffic and decryption of the incoming traffic. Therefore future improvements of IBC cryptographic performance would be especially important for OpenVPN servers that handle a large number of VPN tunnels.

7 Summary and final thoughts

A security architecture based on PKI functionality and Federated Identity is one of the most preferred and reliable solutions for data protection [14]. By building such a security infrastructure, the INDECT project sets ways of working with state-of-the-art security technologies.

A Federated Identity framework both simplifies user management and enhances its security. The INDECT Portal acts as the Identity Provider (IdP) that INDECT applications and Service Providers (SPs) employ to authenticate users employing multi-modal authentication, based on X.509 certificates stored in Smart Cards. Moreover, the separation of INDECT PKI in two hierarchical levels allows both, securing the main PKI element—the ROOT CA—and to manage and operate systems and users in different ways. For instance defining two CAs for users and devices enables the possibility of identifying each device in the systems and to manage each user individually. This type of PKI organization gives us the possibility to secure the data in different ways on different points of the creating, transmitting, editing and storing process.

Communication security is basic a tool for creating a secure distributed environment. Therefore VPNs, also based on X.509 certificates, have been selected for the INDECT security architecture. In this case key management and key negotiation do not require an additional secure channel. This way of creating a secure communication environment simplifies the cryptography key infrastructure and minimizes the number of secret keys. Furthermore, the TLS/SSL protocol is the foundation for the security of web-based INDECT applications and services. It enables communications security, including encryption, message integrity and mutual authentication between clients and servers. However TLS/SSL just provides a base security layer, INDECT web applications may implement further security mechanisms such as password-based authentication, or digital signatures for non-repudiation services.

Nowadays cryptography is the only way to guarantee the confidentiality of data. The development of new protocols and algorithms based on symmetric ciphers like the INDECT Block Cipher (IBC) gives users the possibility to use, exchange and store data in secure way. Thanks to its integration in the popular OpenSSL library, newly developed tools, protocols and applications may easily employ the new IBC algorithm to protect sensitive information when communicating with other entities, especially once the IBC implementation is optimized for performance.

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.

References

1. INDECT Consortium (2009) “D8.1: Specification of requirements for security and confidentiality of the system”, http://www.indect-project.eu/files/deliverables/public/INDECT_Deliverable_D8.1_v20091223.pdf/view. Accessed 20 December 2012.
2. Dierks T, Rescorla E (2008) “The Transport Layer Security (TLS) protocol version 1.2”, RFC 5246
3. EJBCA Enterprise PKI web site (2012) http://www.ejbca.org/. Accessed 1 April 2012.
4. Hickman K (1995) “The SSL Protocol”. Netscape communications corp.
5. INDECT project web site (2013) http://www.indect-project.eu. Accessed 13 January 2013.
6. Niemiec M, Dudek J, Romański Ł, Święty M (2012) “Towards hardware implementation of INDECT Block Cipher”. Proc. 5th International Conference of Multimedia Communications, Services and Security (MCSS 2012), Krakow, Poland
7. Niemiec M, Machowski L (2012) “A new symmetric block cipher based on key-dependent S-boxes”. Proc. 4th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2012), Saint Petersburg, Russia
8. OpenCA project web site (2012) http://www.openca.org/. Accessed 23 April 2012.
9. OpenSSL (2012) The open source toolkit for SSL/TLS. http://www.openssl.org. Accessed 21 December 2012.
10. OpenVPN Community Software. http://openvpn.net/index.php/open-source.html. Accessed 21 December 2012.
11. Stoianov N, Urueña M, Niemiec M, Machník P, Maestro G (2012) Security infrastructures: Towards the INDECT system security, MCSS 2012, CCIS 287, Springer-Verlag Berlin Heidelberg, pp. 304–315, ISBN 978-3-642-30720-1
12. Thomas SA (2000) “SSL and TLS Essentials: Securing the Web”. Wiley Computer Publishing.
13. Urueña M, Muñoz A, Larrabeiti D (2012) “Analysis of privacy vulnerabilities in single sign-on mechanisms for multimedia websites”, Multimedia Tools and Applications
14. Zhelyazkov D, Stoianov N (2009) PKI infrastructure in the BA – Prerequisite for minimization of the risk and enhancement of the information security, CIO journal, special issue communication & information technologies for the defense, pp. 19–20, ISSN 13112-5605