Multimed Tools Appl (2016) 75:10709–10732
DOI 10.1007/s11042-014-2386-3

Security architecture for law enforcement agencies

Manuel Urueña · Petr Machník · Marcin Niemiec · Nikolai Stoianov

Received: 13 January 2014 / Revised: 8 November 2014 / Accepted: 11 November 2014 /
Published online: 29 November 2014

Abstract In order to carry out their duty to serve and protect, law enforcement agencies (LEAs) must deploy new tools and applications to keep up with the pace of evolving technologies. However, police information and communication technology (ICT) systems have stringent security requirements that may delay the deployment of these new applications, since necessary security measures must be implemented first. This paper presents an integrated security architecture for LEAs that is able to provide common security services to novel and legacy ICT applications, while fulfilling the high security requirements of police forces. By reusing the security services provided by this architecture, new systems do not have to implement custom security mechanisms themselves, and can be easily integrated into existing police ICT infrastructures. The proposed LEA security architecture features state-of-the-art technologies, such as encrypted communications at network and application levels, or multi-factor authentication based on certificates stored in smart cards.

Keywords Law Enforcement Agency (LEA) · Security architecture · Police · ICT systems · Security

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

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

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

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

1 Introduction

The continuous evolution of information and communication technologies (ICT) has brought enormous changes, with the Internet being a prime example of this progress. However, organized crime has also embraced these new technologies and is increasingly employing them to perform criminal activities in order to be one step ahead of the law enforcement agencies (LEAs) that pursue them. Therefore police forces must keep up with the pace of evolving technologies, employ new tools to fight new high-tech crimes, and leverage ICT technologies to improve their investigations.

As such, police forces are interested in deploying new investigation tools as quickly as possible in order to keep up with the ongoing cyber-arms race against organized crime. However, since information handled by police forces during their investigations is extremely sensitive (e.g. names of informants, protected witnesses, etc.), any new system deployed as part of police ICT infrastructure must first fulfill a stringent set of security requirements [22]. Therefore, it is common that new tools must implement additional security mechanisms, which are usually custom made. This process may greatly delay the deployment of these new tools, and there is an important risk that such custom-made security mechanisms, added in at a later stage of the development process, do not provide adequate protection. Moreover, police administrators are required to manage a set of heterogeneous ICT systems with fairly different security mechanisms that cannot be easily integrated with other legacy applications, or even with their normal operations (e.g. user management). Therefore they need to be modified continuously in order to include each new application, further increasing the deployment delay.

In order to solve these problems, this paper presents an integrated security architecture aiming to provide common security services (i.e. authentication, authorization, confidentiality, integrity, non-repudiation, auditing, etc.) for police ICT systems that would be implemented with state-of-the-art security technologies. Although this architecture has been designed for new systems to be developed in the future, it can be also used for existing police ICT applications, including legacy ones.

2 LEA security architecture

The proposed LEA security architecture provides a set of common security services, which were previously defined in [22]. This includes authentication, authorization (access control), non-repudiation, privacy, auditing, communication security, data confidentiality, and integrity. Other security services such as efficiency, reliability and availability, as well as common ICT security best practices, are out of the scope of this paper because they greatly depend on the ICT environment in which they are deployed in.

Although each LEA application has its own security needs and characteristics, the proposed architecture includes a number of security infrastructures [20]. that provide common security services using standardized protocols and mechanisms. These security services are provided through a combination of novel and standard security protocols and mechanisms. Figure 1 shows a simplified view of the integrated security architecture for LEA ICT systems.

The main components of proposed LEA security architecture are:

- **Public key infrastructure (PKI)** – for issuing, managing, storing and revoking X.509 certificates used in LEA systems. Certificates are issued to all LEA users and ICT systems for authentication, as well as securing their communications.

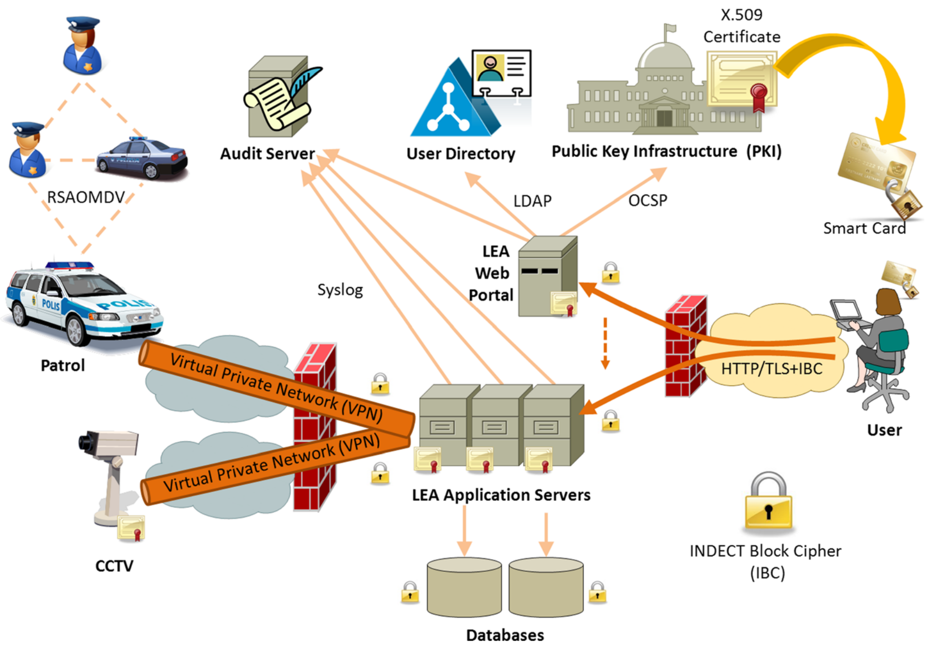

Fig. 1 Security architecture for law enforcement agencies

- **LDAP user directory** – for storing 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 LEA auditors in order to detect suspicious behaviors.
- **LEA web portal** – the homepage of LEA 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** – for executing the services, applications and tools used by the LEA. They can 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 LEA applications provide a web-based interface, and most ICT services will also be web-based, implementing SOAP or REST interfaces and using SSL/TLS for secure communications, featuring mutual client–server authentication.
- **Databases** – although located within the police data center, they should communicate in a secure way with LEA application servers as well as being encrypted, for instance using novel cryptographic algorithms presented later.
- **Virtual private networks (VPNs)** – 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)** – storing users' certificates, they are issued by the LEA PKI and used for access control by the central LEA web portal, as well as encrypting and signing e-mails and documents.

In order to guarantee the robustness of the security architecture and to support a wide range of applications, standard security protocols such as TLS/SSL and IPSec are preferred to proprietary or custom ones. Nonetheless, the proposed LEA security architecture also includes novel mechanisms such as the new INDECT block cipher (IBC) that can be used to encrypt TLS/SSL sessions and VPN tunnels.

For a complete description of the proposed LEA security architecture, the interested reader is referred to [8].

2.1 LDAP user directory

Although it is recommended that all LEA applications are based on the proposed TLS-enabled mutual authentication of users, it is still possible that some applications, including legacy police systems, do not implement certificate-based authentication. Therefore the LEA security architecture also includes a LDAP directory service that stores users' information, including user credentials (i.e. login/password) for such legacy applications. A LDAP-based directory service has been selected because nowadays it is commonly employed by enterprises, including law enforcement agencies, for user management (e.g. Microsoft Windows Active Directory is based on LDAP). The proposed LDAP schema has been designed to be as standard as possible in order to be easily integrated in existing LDAP systems.

LDAP user directory contains information about all LEA applications and users. This way it is possible to specify, in a centralized way, which applications can be accessed by each user, as well as to specify common authorization attributes of users. Legacy applications can then query the LEA user directory by means of LDAP commands in order to: (i) verify whether a user has access to that application, (ii) validate the authentication credentials provided by the user, and (iii) check the user's authorization attributes to enable only the allowed actions.

2.2 Audit server

Given the sensitive information that law enforcement agencies (LEAs) handle, and in order to protect the privacy of citizens, the operations of LEA agents are continuously monitored by a specialized department of auditors. However, the complexity of such system would require LEA auditors and security personnel to check the logs of a huge number of systems and applications. Therefore the proposed LEA security architecture also includes a centralized audit server that aggregates the logs of all LEA subsystems. This way, LEA auditors and ICT security personnel only have to monitor a single log stream, with the additional benefit of easing the correlation of events from different systems, which could be easily missed with separated logs.

This centralized log server also provides benefits from a security point of view, because logs are stored in two places, locally at the application server and remotely at the audit server, which can only be accessed by LEA auditors and that does not support deleting log records. This way, even if a server is compromised and the attacker is able to erase its actions from the local log, by then they would be already stored on the audit server and thus subject to auditors' scrutiny. Therefore all relevant user actions and system events must be logged locally by the application server and be sent to the centralized audit server using the syslog protocol. This also applies to remote applications, that may use VPN tunnels to send its log events to the audit server, either directly or through an application server acting as a proxy.

For especially sensitive operations (i.e. authorizing the wiretap of a suspect) and/or due to regulatory requirements, just logging those operations may be not enough. The details of such sensitive operations must be cryptographically signed by the officer requesting/approving it for proper authorization and to ensure non-repudiation.

3 New cryptographic algorithms

This section presents new cryptographic solutions that have been developed by the EU INDECT project [7]. Currently, two novel symmetric ciphers (block and stream) as well as a hash function are ready to be used by end-users. Additionally, new high-level security methods for quantum cryptography have been proposed. These solutions are the significant part of LEA security architecture.

3.1 INDECT block cipher (IBC)

Encryption of confidential data is the most important task of cryptography. It relies on transforming plain data into an encrypted form, unreadable to anyone except to those possessing the cryptographic key, by using an appropriate algorithm called cipher.

Nowadays, there are many different cipher algorithms. A well-known example is the advanced encryption standard (AES) that encrypts data using simple functions: substitutions based on a single S-box, permutations, and mixing the key with the data. The INDECT project has further developed these ideas: employing more substitution boxes (S-boxes), using a cipher with dynamic structure, etc. It allows increasing confidentiality level of encrypted data. Also, it is important to devise new ciphers with a completely different structure in order to replace currently used algorithms in case of new cryptanalysis methods appear, or due to older ciphers having too short key lengths (e.g. DES).

In general, the new cipher, called INDECT block cipher (IBC), consists of nonlinear transformations, which are dependent on the key [12,13]. This feature ensures a high level of security. Additionally, the large number of secure S-boxes makes each encryption highly unique. The construction of this cipher is based on substitution and permutation functions that are used in each round. This structure ensures a good performance and a fast data encryption.

The IBC algorithm is a block cipher. Each 256-bit block of data is divided into 64 sub-blocks. Each sub-block is transformed by the appropriate substitution box and output values are concatenated into one 256-bit block. At the end of the round, the permutation function based on S-box, further modifies the 256-bit block of data. These steps are repeated for a number of iterations (e.g. a minimum of 8 times).

One of the most novel ideas of the IBC cipher is its unique usage of the key. An IBC key is still a pseudo-random sequence. However it is used to create new S-boxes. These substitution boxes are based on the AES S-box, and ensure the same level of security. In this way, we can create about $5.35\cdot 10^{18}$ new S-boxes from a single AES S-box. All new S-boxes represent a unique non-linear transformation: substitution or permutation. Because of S-box size, the cryptographic keys of IBC cipher must be a multiple of 64 bits. Four key lengths have been chosen for practical use:

- **128 bits** where two S-boxes are used: one for substitution and another for permutation. For 128-bit keys, eight rounds of cipher are proposed.
- **192 bits** where three S-boxes are used: two for substitution and one for permutation. For 192-bit keys, ten iterations of the cipher are proposed.
- **320 bits** where five S-boxes are used: four for substitution and one for permutation. For 320-bit keys, twelve rounds of the cipher are proposed.
- **576 bit**s where nine S-boxes are used: eight for substitution and one for permutation. For 576-bit keys, fourteen rounds of the cipher are proposed.

The robustness of the IBC cipher was tested by means of several basic security metrics:

- The cipher and his internal S-boxes are balanced: the number of ones is half of all possible output bits.
- IBC meet strict avalanche criterion (SAC): a change of a single input bit changes each of the output bits with a probability of one half.
- The cipher and every S-box have 100 % completeness: for each selected entry, we can choose the value of other inputs in such way that it will bring change to the entry of any change in output.
- Diffusion order equals one: even if the value of the output bits change is large, the number of changes to entry is relatively low.
- S-boxes have low XOR table: every S-box have almost perfect XOR table (zeros and twos), it prevent against differential cryptanalysis.
- Nonlinearity of S-boxes is equal to 112 (maximum in this case is 128): it shows how much a function differs from the closest affine function.

All proposed S-boxes have good security properties: balancing, SAC, completeness, diffusion order, low XOR table, and nonlinearity. In comparison, the AES cipher uses only one such S-box. The parameters indicate that IBC cipher is resistant to differential and linear attacks (cryptanalyses). The authors performed multiple tests that verified the robustness of IBC cipher and that it ensures a high level of confidentiality. One of these tests is presented below. It compares a single encryption cycle of IBC and AES ciphers with similar parameters:

```
IBC cipher:
  SAC equals: 50 %
  Completeness is ensured for 100 % possible inputs
  minimum distance to affine function is:
  for 1 bit 105 (most significant)
  for 2 bit 92
  for 3 bit 101
  for 4 bit 105
  for 5 bit 99
  for 6 bit 104
  for 7 bit 106
  for 8 bit 106
AES cipher:
  SAC equals: 49 %
  Completeness is ensured for 100 % possible inputs
  minimum distance to affine function is:
  for 1 bit 100 (most significant)
```

```
    for 2 bit 105
    for 3 bit 101
    for 4 bit 103
    for 5 bit 103
    for 6 bit 96
    for 7 bit 101
    for 8 bit 106
```

Another important step towards using IBC in practice has been the implementation of error detection methods to increase the reliability of encryption software and to ensure a high level of data security. These error detection methods can be applied before the permutation step of the IBC algorithm since they do not change the number of '1 s' or '0 s' in the output, just their order. Therefore it is possible to verify if the output of the encryptor has any error and even how many bits have been affected.

Three different error detection algorithms were implemented in the software IBC encryptor: Parity checks, Berger code and cyclic redundancy check (CRC). Parity defines whether a number is even or odd – in this implementation refers to the evenness or oddness of a particular set of bits. The IBC 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 s' (or '1 s') 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.

Each implemented error detection method is able to find a different amount of errors during the encryption process. The average number of detected faults is presented in Fig. 2. It was proved that CRC algorithm successfully detects any occurred fault. However it does not provide information about the amount of occurred errors. 1-bit parity check has a worse error detection success rate, because any even number of errors cannot be detected. Berger code is a much better method than 1 or 2-bit parity: however for a small amount of errors, the 4-bits parity check is able to detect them with better precision. Finally, although the 8-bits parity check involves a significant overhead, this method accurately detects almost half of errors in the encrypted data.

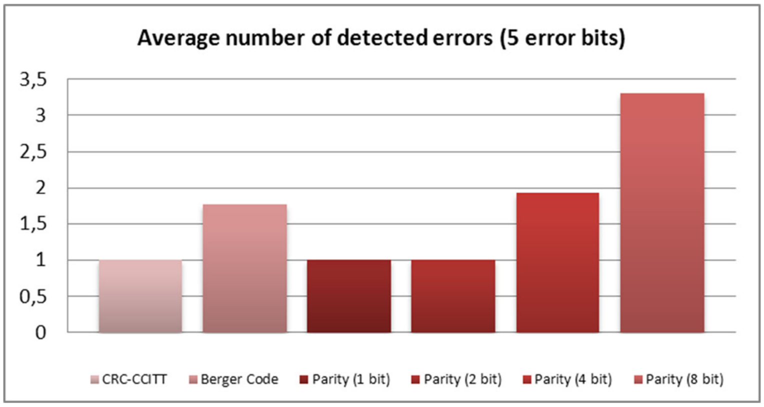

Fig. 2 Average number of detected errors for different methods (5 error bits were inserted during the encryption process)

3.2 INDECT stream cipher (ISC)

Stream ciphers are able to encrypt a single bit (or byte) of data using a generated key. They are a class of symmetric ciphers that are based on the one-time pad principle. The main difference is that in one-time pad ciphers the key length must be equal to the message length, while stream ciphers employ a key with a much smaller length.

The INDECT stream cipher (ISC) provides two encryption modes: a keystream generation based on linear feedback shift registers (LFSRs), and a symmetric stream cipher based on the IBC cipher.

The keystream generator based on LFSRs uses 16 binary registers and one additional operating in the integer domain. All registers are initialized at the beginning of the encryption process with the provided key. Since the size of registers varies, the provided key is truncated each time before initialization in order to fit in a given register. The key size required for this encryption mode must be 256 bits long.

The stream cipher based on IBC is the solution where a block of data is encrypted by the IBC algorithm using two operation modes: output feedback (OFB) and cipher feedback (CFB), which may operate over bit or byte streams.

3.3 INDECT hash function (IHF)

Hash functions are a group of transformations where variable length data are transformed into a small, fixed-size digest. This transition must be one-way only and part of the original information is lost. Although these functions have a small probability of collision (i.e. finding two different input data produce the same output), it is almost impossible in practice.

A new hash function, called INDECT hash function (IHF), has been created by the INDECT project. The hash function has a structure of substitution-permutation network with different key lengths and different number of rounds. Therefore, this hash function provides different security levels. The design of IHF is based on the IBC algorithm using the CBC-MAC chaining mode. Thus, the key sizes and number of rounds proposed for the INDECT Hash Function are the same as in IBC.

3.4 Quantum cryptography methods

Quantum cryptography (QC) is a new way of solving the key distribution problem of symmetric ciphers. It provides a secure key distribution service by means of the laws of quantum mechanics:

- Any measurement modifies the state of the transmitted *qubit* (quantum bit) and this modification can be discovered by end-users.
- It is not possible to clone an unknown qubit (it is not possible to measure the quantum state and simultaneously send a cloned qubit to the real receiver).

Some new high-level quantum cryptography methods have been proposed [14] by the INDECT project, including the verification of the described solutions. This study is based on both theoretical analyses and empirical tests in a custom-developed simulator (the QKD protocol simulator) [15].

The new QC methods are based on the idea of measuring the security level during the QBER estimation and privacy amplification processes. Two new functions were proposed: a measure of security and entropy of security [15]. These functions define the average security of the key when we uncover and compare a part of the exchanged key. Using these functions it is possible to specify the security levels of a QC system. Two security levels are proposed: basic and advanced security. Thanks to these security levels the security for specific end-users and services can be personalized. Also, the QC methods have been also validated by means of simulations.

4 LEA public key infrastructure (PKI)

A public key infrastructure (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 big 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 the proposed security architecture 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 validity and integrity of the stated information [1].

4.1 Digital certificates

A digital certificate is a representation of the link between the identity of a person or device and its corresponding digital information. This digital cryptographic information is essentially the public keys of the subject. The digital certificate also contains other information related to people or devices, and this information is independently signed by the so-called certification authority (CA).

The basic elements of the LEA PKI infrastructure are:

- **Root CA server** – is based on a self-signed (root) certificate, although it is always offline because it is only used to issue the certificates of its Sub-CAs.
- **Users CA** (Subordinate certificate authority for users) – manages all certificates related to users. These certificates are stored on smart-cards to enable two-factor authentication.
- **Devices CA** (Subordinate certificate authority for devices) – manages all system certificates issued for devices. In the LEA security architecture devices could be: servers, CCTV cameras, users' desktops, laptops, tablets, smart phones, other communication devices, etc.
- **Users RA** (Registration authority for users) - generates certificates from PKCS#10 requests, generates PKCS#12 for the end user, performs key recovery of the users'key (if requested using PKCS#12), edits users' data, revokes certificates, renews the certificates of existing users, generates a key storage for existing users, etc. The users RA of the INDECT project is operated by the EJBCA software package [4].
- **Devices RA** (Registration authority for devices) - generates certificates for devices, edits devices profiles, revokes certificates, renews certificates for existing devices. The INDECT devices RA is also operated using the EJBCA software.
- **PKI backup/log server** – for disaster-recovery procedures and for auditing the processes of certificate management. PKI logs should also be copied into the global LEA audit server.

The architecture of the proposed LEA PKI infrastructure is shown in Fig. 3.

The process for requesting and issuing certificates through a registration authority (RA) are as follows:

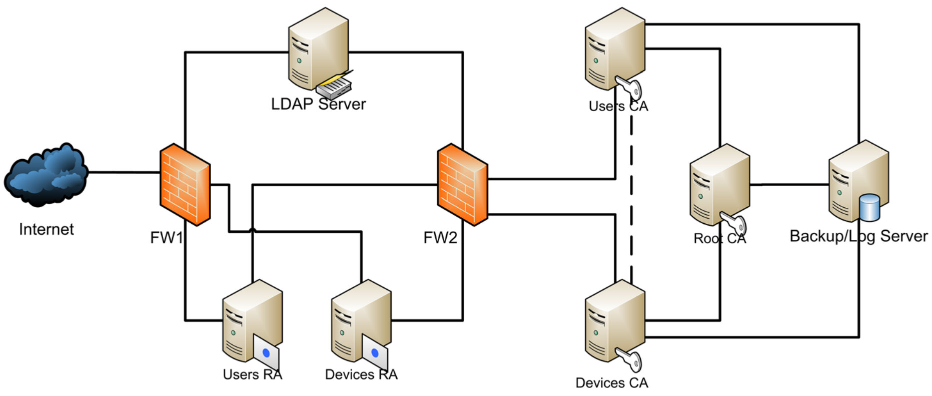

Fig. 3 LEA public key infrastructure (PKI)

1. A certificate request is sent to the RA by a user.
2. The certificate request is checked by the RA and stored locally.
3. The CA is waiting for certificate requests and periodically checks the RA database. It processes the request by issuing a certificate and stores it back in the RA's DB.
4. The RA periodically looks for new certificates issued by the CA.
5. The RA sends the new certificate to the user after processing it.

In order to issue a user certificate, the following data are required:

- Username and password – used by RA to access user's certificate data. The RA can edit certificate data until it sends a request to CA to issue the user's certificate.
- User's e-mail address – mandatory information for issuing a certificate.
- Common name (CN) – this value is used for short description of the certificate; usually the rank and full name of the user can be used.
- Country – in one national LEA this field will be same for all users, but in international LEA organizations, like INTERPOL or EUROPOL, this field would contain the country of origin of the user.
- Organization – this field should have the name of the LEA organization name, or it can specify the LEA's office for example (e.g. LEA-EU or LEA-BG etc.).
- Given name – the user's first name.
- Surname – the user's last name.
- UID – this string field is an ×.509 v3 certificate extension that is used as a unique identifier of the LEA user (i.e. LEA officer number).
- Lvl – this field specifies the maximum clearance level of the user. That is, the highest level of secret information which the user can access to. This field is also an extension to the standard certificate structure. Possible values range from 0 (unclassified) to 4 (top secret).

4.2 Certificate revocation list (CRL)

Often some certificates must be revoked before certificates' validity period expire, for instance if its private key is somehow compromised. In this case the CA must create a list of revoked certificates, called certificate revocation list (CRL). This list includes the serial number of the revoked certificate and the reason for its revocation. Up to date information about revoked certificates is critical for a healthy PKI system. Therefore, the proposed update time for the CRLs of the LEA PKI is 5 min. Four settings should be also configured on EJBCA [4] (the CA software used for managing certificates in INDECT) to define how CRL generation is done:

- CRL expire period: This is the validity period of the generated CRL. It is set to 24 h.
- CRL issue interval: This is the time when the new CRL will be issued. For LEA PKI it is set to 0, meaning that the new CRL will be issued after the old CRL is expired (24 h).
- CRL overlap time: This setting defines the time when the new CRL should be issued before the old CRL expires. For LEA PKI, the CRL overlap time is set to 10 min.
- Delta CRL period: This setting defines the amount of time a delta CRL (i.e. that only includes the differences with the previous CRL) is valid after being issued.

4.3 Certificate extensions for LEA users

Certificate extensions are optional by definition. This functionality was introduced in X.509 version 3. Based on these properties (extensions) it is possible to create a template and use it for issuing certificates for different purposes [1].

Each Police officer has a unique identifier that is used for authentication and identification purposes and stored in their certificates. The credentials of all users will be stored in LDAP repositories. For uniformity, the UID (user ID) attribute employed for user identification in LEA systems is also stored in certificates for offline access.

Additional information for rights management, such as the clearance level of users, is also stored in certificates. We consider the common security access levels: unclassified, restricted, confidential, secret and top secret. Therefore LEA certificates have an additional extension field that stores the maximum access level of the user as follows:

- Unclassified access level: 0
- Restricted access level: 1
- Confidential access level: 2
- Secret access level: 3
- Top Secret access level: 4

5 LEA communications security

Given the distributed nature of modern LEA ICT systems, one of the main components of the secure communication infrastructure is a virtual private network (VPN) framework that will enable the secure communication among multiple remote nodes and servers interconnected over public networks [8]. Nowadays VPNs are mostly based on two different technologies – SSL (secure socket layer) and IPsec (internet protocol security). The following subsections will overview the VPN software packages being evaluated by the INDECT project, which can be safely used in the proposed LEA security architecture. Finally, the performance of HTTPS with mutual authentication is evaluated in order to validate the proposal of performing user authentication using X.509 client certificates.

5.1 SSL VPNs

The best open-source SSL VPN solution is the OpenVPN software package [18]. OpenVPN can be installed in computers with almost any operating system. OpenVPN software is very flexible and scalable. The advantages of OpenVPN SSL VPNs are as follows [5]:

- OpenVPN can be installed on various platforms – computers with Linux, Windows, or Mac OS X operating systems; smart phones with Android, iOS or Windows Phone.
- OpenVPN offers two basic modes that run either as a layer 2 or layer 3 VPN. OpenVPN layer 2 tunnels are able to transport Ethernet frames. Because of this ability, OpenVPN behaves more as an IPsec VPN, than as a typical SSL one, which is mainly used for a secure web communication.
- Once OpenVPN has established a tunnel, the central firewall in the Police headquarters can protect the client device, even though it is not a local one.
- OpenVPN can use either TCP or UDP transport protocols and can work as a server or client. To improve the security level, a server can accept only connections initiated by clients within the specific virtual private network.
- Since OpenVPN version 2.0, a special server mode allows multiple incoming connections on the same TCP or UDP port, while still using different configuration for every single connection. Thus, only one port in the firewall has to be opened.
- OpenVPN has no problems with network address translation (NAT) and hence can be employed in networks with private IP addresses.
- OpenVPN offers many possibilities to start individual scripts during connection setup. These scripts can be used for a large variety of purposes like authentication or failover.
- Both tunnel endpoints can have dynamic IP addresses.

Within the LEA system, users will employ mainly OpenVPN to securely communicate between their remote terminals (e.g. desktop, laptop, tablet, smart phone, etc.) and servers located in the police headquarters (see Fig. 4). The LEA devices CA will be employed to authenticate the individual terminals.

To further improve the LEA communications security, so-called two-factor authentication mechanism can be employed. In that case, the user certificate is stored in a smart card. Such a solution provides a strong user authentication, because the user needs to possess the

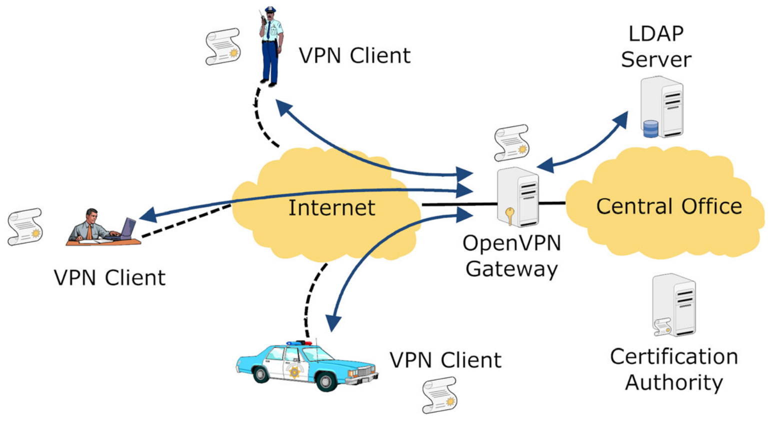

Fig. 4 Example of an OpenVPN network

authenticator token (i.e. smart card) and, concurrently, needs to know the password which protects the stored certificate.

To use these advanced authentication mechanisms, two open-source software packages exist – OpenSC and OpenCT. OpenSC [17] provides a set of libraries and utilities to work with smart cards and USB tokens. Its main focus is on authenticators that support cryptographic operations, and facilitate their use in security applications such as authentication, mail encryption and digital signatures. OpenSC implements the PKCS#11 API, so applications supporting this API (such as Mozilla Firefox and Thunderbird) can use it. On the card side, OpenSC implements the PKCS#15 standard and aims to be compatible with every software or card that does so too. OpenCT [16] implements drivers for several smart card readers and USB tokens. OpenCT also has a primitive mechanism to export smart card readers to remote devices via TCP/IP that may be useful in certain scenarios.

5.2 IPsec VPNs

The StrongSwan software package [21] provides an open-source IPsec VPN solution. StrongSwan is intended primarily for Linux devices, but it is fully compatible with other standard IPsec VPN implementations, and thus can be used in networks with mixed equipment (see Fig. 5).

The main benefits of StrongSwan IPsec VPNs are as follows:

- StrongSwan supports various popular platforms – computers with Linux, Mac OS X, or FreeBSD operating systems; smartphones with Android.
- StrongSwan implements both IKEv1 and IKEv2 (internet key exchange) protocols, and fully supports IPv6.
- StrongSwan enables dynamic IP addresses and interface updates with IKEv2 mobility and multi-homing protocol, and IKEv2 multiple authentication exchanges.
- It allows the automatic insertion and deletion of IPsec policy-based firewall rules.

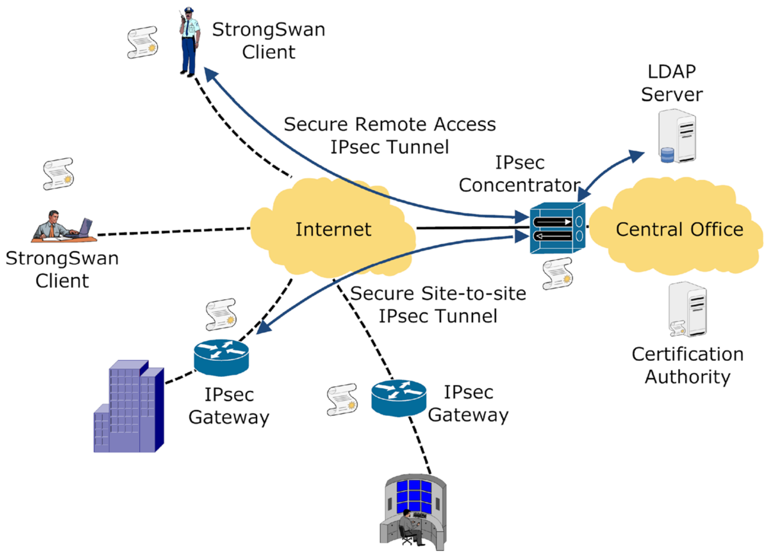

Fig. 5 IPsec VPN with StrongSwan clients

- StrongSwan supports NAT-Traversal via UDP encapsulation and port floating.
- The XAUTH functionality is based on IKEv1 main mode authentication.
- The device authentication is based on X.509 certificates or pre-shared keys.
- StrongSwan enables secure IKEv2 EAP (extensible authentication protocol) user authentication.
- RSA private keys and certificates can be stored on smart cards or USB tokens supporting the PKCS #11 standard.

5.3 Evaluation of OpenVPN with INDECT block cipher

The INDECT block cipher (IBC), which was described in Section 3.1, has been implemented into the OpenSSL 0.9.8v software package. OpenSSL is a collaborative project to develop a robust, commercial-grade, full-featured and open source toolkit implementing secure sockets layer (SSL v2/v3) and transport layer security (TLS) protocols as well as a powerful, general purpose cryptography library. The core module of OpenSSL implements the basic cryptographic and utility functions.

OpenVPN software package, which was described in Section 5.1, uses OpenSSL to perform all cryptographic operations; therefore it is also possible to protect LEA 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.

To analyse the performance of OpenVPN cryptographic operations, a benchmark using the *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 1. 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 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. For instance for video streaming from remote CCTV cameras to LEA headquarters that would be explained in Section 7. 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.

Nevertheless, improving the performance of the IBC cipher is one of the next goals of our research in this area. Furthermore, an overall security evaluation of OpenVPN tunnels with IBC should also be performed (e.g., using the tool for penetration tests described in [19]).

5.4 HTTPS mutual authentication performance evaluation

A key protocol of the proposed security architecture is HTTPS, since we assume most applications, including the LEA web portal, provide a web interface and/or REST or SOAP

Table 1 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

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. [2] 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. [3] 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 [10] 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 proposed 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 authenticate 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 test bed 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 i5-3320 M CPU @2.60GHz, 8GB RAM, Linux 3.8.13-64 bits) interconnected with a Fast Ethernet LAN. Table 2 shows the results of the performance tests, where a small static HTML page (177 bytes) was downloaded 10,000 times consecutively by 16 parallel

Table 2 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

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 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 [2,3,10], 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 proposed LEA security architecture, as well as to employ client certificates to implement a secure user authentication mechanism.

6 Secure mobile ad hoc networks for LEAs

In emergency scenarios, where the communications infrastructure may be severely damaged or just overloaded, mobile ad hoc network (MANET) technologies may become an essential tool for security forces or other first responders. Moreover, MANETs can be also useful for other LEA operations in remote areas out of cellular coverage.

Although there is a large number of routing protocols for MANETs, few have considered the stringent security requirements of law enforcement agencies or other kind of security forces. Furthermore, most of them [6] only consider logical attacks, (like “black/grey holes”), neglecting the biggest difference with traditional routing protocols, that is, the use of wireless links that are also vulnerable to physical attacks, such as jamming or sniffing.

Therefore, a new routing protocol for MANET, called RSAOMDV (ranked secure ad hoc on-demand multipath vector) has been designed in order to provide a secure ad hoc network for Law Enforcement Agencies and other security forces. In particular, RSAOMDV is a secure routing protocol based on AODV, the most popular reactive ad hoc routing protocol, which is already standardized by the IETF. In order to deal with physical attacks, RSAOMDV is able to discover several node-disjoint paths between each source and destination pair. Therefore, even if an attacker is able to disrupt one of the paths by means of jamming, the communication will be still possible by using the remaining paths.

Essentially, when the source node A wants to communicate with the destination node B, it sends a multipath route request (MRREQ) that floods the MANET. These MRREQ messages will arrive to B though all possible paths. Then B sends a route reply (RREP) message back to the source using all the different disjoint paths that have been found. In order to know which paths are fully disjoint it is only necessary to compare the first hop of each path (i.e. A's neighbour), since intermediate nodes only forward the first received MMREQ to avoid loops. Furthermore, intermediate nodes use these MRREQ and RREP messages to learn, respectively, where are the source and destination nodes.

This simple multipath discovery mechanism is based on the AOMDV [11] multipath routing protocol, although RSAOMDV adds several security mechanisms to avoid logical attacks. First of all, RSAOMDV performs access control by means of digital certificates issued to all nodes in the MANET by a trusted certification authority (i.e. LEA PKI). This way, a RSAOMDV node will only accept as its neighbour another node that has a valid certificate. Moreover, the access control protocol uses random nonces and timestamps to avoid replay attacks and to be sure that there is a bidirectional connection in order to thwart certain kinds of "rushing" attacks. Also, all RSAOMDV messages are signed by the source node, as well as by the previous hop in order to avoid impersonation and injection attacks. Thus each node can verify the integrity and the authenticity of communications from trusted nodes.

Finally, RSAOMDV data packets are encrypted end-to-end with a symmetric session key, securely exchanged during the path setup. However even encrypted information may suffer from statistical analysis attacks. Therefore, for extremely sensible and confidential information, RSAODVM may choose to use only paths composed by high rank officers. This information is also obtained during the path setup phase because certificates have a field with information about the rank or access level of the user. This rank will be used to make decisions about the discovered paths, such as which ones are composed by high-rank officers and thus are more secure. Currently, MRREQ messages use this rank and a mechanism based on hash-chains [6] to protect mutable fields and to prevent malicious nodes to hijack the routing and advertise bogus paths as better than valid ones.

Moreover, RSAOMDV features a novel certificate distribution and revocation protocol for ad hoc networks that does not require a permanent connection to the LEA PKI to check its certificate revocation list (CRL) or to any other fixed infrastructure. Certificates can be distributed in RSAOMDV either by including them in MRREQ and RREP messages (e.g. to check the signature of the source node) or by explicitly asking the previous node (e.g. to check its signature) using certificate request/response (CREQ/CREP) messages, which also enable neighbour access control. All obtained certificates are stored in a cache, so they do not have to be requested again (i.e. to fixed neighbours).

On the other hand, the main idea of RAOMDV certificate revocation is to allow high-rank users to revoke certificates of lower-rank ones, since we assume that low-rank users (e.g. a SWAT agent) are easier to compromise than high rank ones (e.g. the officer at charge in the operations centre). Thus, when a node receives a revocation message, which is periodically flooded, it first checks that it is signed from an officer with a higher rank than the revoked certificate. In that case, the revocation message is stored in the certificate cache (so new certificates are checked against it) and all data from the revoked user is rejected (e.g. routes from/to or going through it). Nevertheless, in order to prevent misuse (e.g. attackers compromising a high rank certificate and then try to revoke all lower rank certificates), the revocation mechanism can be also overridden by manually revoking the certificate of the compromised officer, and thus ignoring all his previous revocation messages.

In order to test RSAOMDV in a variety of scenarios with diverse scales, the complete protocol has been implemented in the Java-based JiST/SWANS [9] simulator. The results reported in this document were obtained from a simulated environment with 46 nodes in a 2×2 km^2 area and featuring IEEE 802.11 radios with a transmission range around 200 m.

One of the main characteristics of RSAOMDV is its ability to find multiple node-disjoint paths between source and destination nodes. This way, even if some authenticated nodes are attackers (i.e. "insiders"), and thus are able to disrupt any communication traversing them, RSAOMDV is still able to find valid paths between source and origin. This property can be seen in Fig. 6 that shows, for an increasing number of insider attackers, the average number of valid paths that can be found between random source destination pairs from a set of 46 nodes. As it can be seen, even a small number of insiders may disrupt single-path routing protocols, since 7 attackers (a 15 %) could be able to disrupt on average a 25 % of all paths. However, a node-disjoint multi-path protocol is able to find at least one valid path unless a large majority of the nodes are attackers (35 out of 46 nodes, a 76 %). Therefore it is clear that, from a security point of view, multi-path ad hoc routing protocols like RSAOMDV are preferable to single-path ones.

7 An example scenario

Let us illustrate the operation of the proposed LEA security architecture using the use case depicted in Fig. 7. In particular, let us follow the actions of a new officer who has recently joined the LEA implementing the proposed security architecture.

First of all, the new officer must obtain a valid certificate from the LEA PKI in order to be registered as a LEA user and access any LEA system. After checking the identity of the new user, the registration authority (RA) of the LEA PKI issues a certificate request to the users' certification authority (CA). Then, the RA operator chooses a template from which the user's certificate will be issued. The LEA PKI should support different certificate templates for users, devices and servers. The key length for users should be at least 2048 bits or 1024 bits for legacy devices. The LEA root CA itself should employ a key of 8192 bits. After finishing the registration process, the CA looks for a certificate request. Issuing a certificate consists of entering the username/password and choosing the requested certificate. The next step is to store the certificate in a holder, such as a smart cart. It is also possible to issue and store certificates in the LEA LDAP directory or the user's browser.

After issuing and storing the certificate, the user receives a smart card which contains her certificate; she is now ready to use LEA services according to her configured rights and privileges. The smart card also serves as her identification badge. This ensures that LEA users

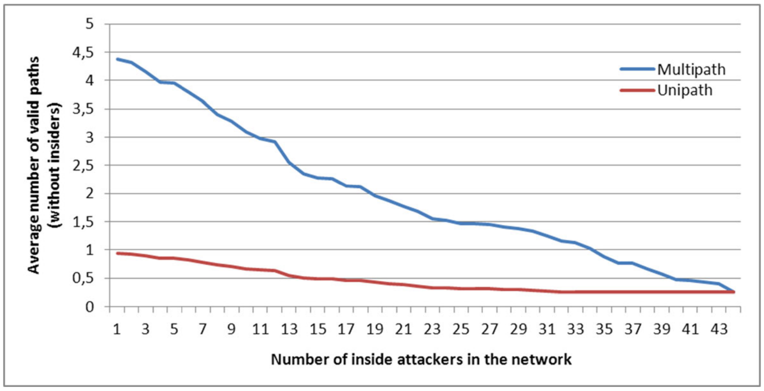

Fig. 6 Tolerance to insiders attacks of single-path and multi-path ad hoc protocols

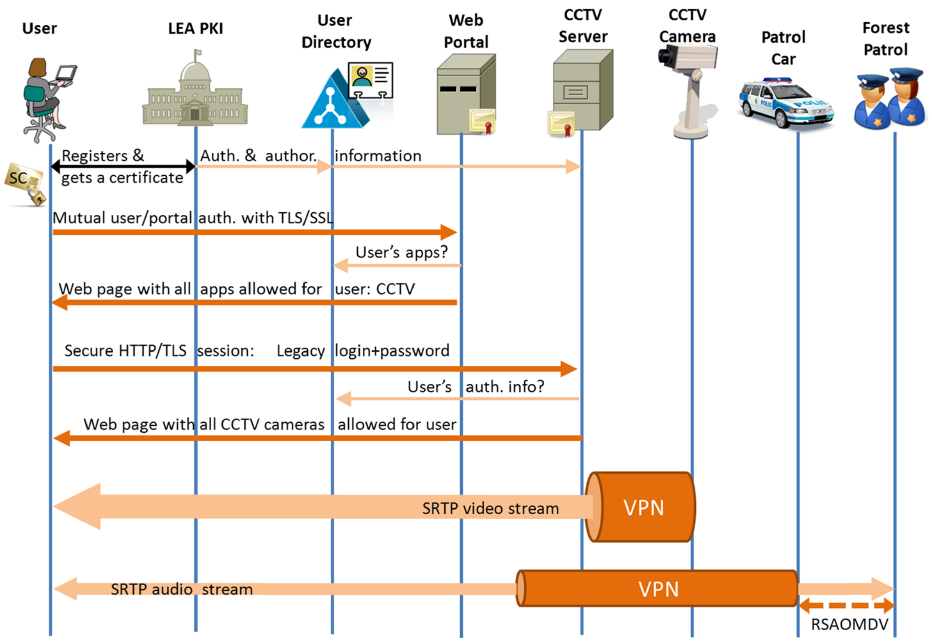
User
LEA PKI
User Directory
Web Portal
CCTV Server
CCTV Camera
Patrol Car
Forest Patrol
SC
Registers & gets a certificate
Auth. & author. information
Mutual user/portal auth. with TLS/SSL
User's apps?
Web page with all apps allowed for user: CCTV
Secure HTTP/TLS session: Legacy login+password
User's auth. info?
Web page with all CCTV cameras allowed for user
SRTP video stream
VPN
SRTP audio stream
VPN
RSAOMDV

Fig. 7 An example use case

always carry their certificates and private keys with them, and promptly notify relevant authorities if the card gets lost. Moreover, since the private key is securely stored on the smart card and protected by a PIN, the associated certificate of a lost card can be revoked before any attackers are able to extract the key. Additionally, the system administrator stores the common authorization rights of the new user in the LEA user directory, as well as in the applications she can access if they have specific user privileges. For instance, if the new officer is in charge of the CCTV cameras of a remote high-security facility, the CCTV system administrator should enable her viewing rights to that set of cameras; however, she may not be permitted to control the pan-tilt-zoom (PTZ) cameras until she completes the required training.

Once the user privileges are specified, the user can access the LEA web portal using a simple web browser. However, since the LEA portal only allows secure HTTPS sessions with mutual authentication, the browser, after checking the web server certificate (i.e. web browsers of all LEA computers trust LEA PKI root certificate), asks her to provide a certificate from the same LEA PKI. To do so, the user inserts her smart card into the SC reader of the computer and types the PIN to unlock it. Then the browser uses that certificate to authenticate against the LEA portal server. Note that this process is a two-factor authentication, because the user has to employ her smart card (i.e. "something you have") and enter a PIN (i.e. "something you know"), or even a second password to access sensitive applications. Following successful mutual authentication, which is also logged in the centralized LEA audit server, all web traffic is encrypted by the underlying TLS/SSL session using the INDECT block cipher (IBC). After a key has been established successfully, confidentiality is ensured by an IBC cipher with a 128-bit key, and data integrity is protected by the INDECT hash function (IHF). Generally, for security reasons (e.g. successful brute-force attacks using faster computers), modern ciphers should only be used for a limited period of time. Therefore, IBCs with longer keys (192, 320 or 576-bits) are likely to be used in the future. Additionally, significantly more secure solutions, such as quantum key distribution (QKD) algorithms, could be used in the far future, when this kind of secure communications will be available in real networks.

The LEA portal shows to the user the list of all applications she can access, by querying the user LDAP directory where common user rights are stored. The user then selects the CCTV application, and the LEA portal redirects the HTTP session to the CCTV application server, which is also web-based and served by HTTPS. Again, this HTTPS session may feature mutual authentication. However, legacy applications that do not support HTTPS mutual authentication can still authenticate their users by means of the login/password stored in the LEA user directory. However, this option should only be used as a last resource in legacy LEA applications, while they implement more advanced authentication mechanisms. Once authenticated and logged, the user can now access the CCTV cameras of the remote location she is monitoring (and only those ones).

Those video streams are conveyed to the LEA headquarters (HQ) where the CCTV application server is located via a VPN that connects the remote facility to the LEA HQ. All end devices in LEA VPNs use device certificates for their mutual authentication. Beside commercial solutions, these VPNs can be based on open-source solutions such as OpenVPN or StrongSwan. In particular, the OpenVPN software has been improved to implement the IBC cipher. The results of performance tests of such a solution were shown in Section 5.3.

Imagine a later scenario where the CCTV operator notices a strange movement in the forest beyond the perimeter of the remote high-security facility. She immediately contacts the remote security officer, who sends a patrol to check that part of the perimeter. The patrol car stops by the first trees, and two agents enter the forest to double check the alert. Since the trees block the transmissions of their cellular uplinks, the agents use a RSAOMDV-based secure ad hoc network to communicate with each other and with the security officer and the LEA HQ, using the patrol car as a gateway. Therefore, the agents (who also have valid certificates from the LEA PKI) may communicate hop-by-hop in a secure and reliable way via the multiple disjoint paths established by RSAOMDV, which are also encrypted end-to-end to prevent eavesdropping. Finally, the two agents return from the forest reporting that the flock of deer they have found does not seem an immediate threat to the facility.

8 Summary

This paper presents an integrated security architecture allowing law enforcement agencies (LEAs) to employ novel tools and applications in a secure and reliable way. By providing common security services such as public key infrastructure (PKI) or a centralized audit server, it will be much easier to integrate new investigation tools into existing police ICT systems, since it is only necessary to integrate the appropriate security service, instead of adapting each separate application. It is also worth noting that the proposed LEA security architecture also considers legacy applications, which makes it possible to integrate existing police ICT systems into the proposed security architecture.

In particular, this paper presents in detail the different security infrastructures, mechanisms and protocols that provide the main security services of such architecture. For instance, it provides an overview of the novel cryptographic algorithms developed by the INDECT project – INDECT block cipher (IBC), INDECT stream cipher (ISC) and INDECT hash function (IHF) – as well as including an analysis of the security properties of IBC and the security level provided by quantum key distribution protocols.

One of the key characteristics of this LEA Security Architecture is the widespread usage of digital certificates to authenticate users and devices and to establish secure communications among them. Therefore the LEA public key infrastructure (PKI) is one of the main components of this solution. The LEA PKI has a hierarchical structure with cross-certification, which enables all LEA users and devices to be securely associated with an asymmetric key pair by means of digital certificates. Moreover, multi-factor authentication is supported by storing users' certificates in smart cards.

Secure communications are implemented via standard security protocols such as virtual private networks (VPNs), supporting both SSL- and IPSec-based VPNs, and transport layer security (TLS), which also uses the digital certificates issued by the LEA PKI for mutual authentication. The performance of such an IBC-enabled VPN has been evaluated using OpenVPN and OpenSSL. The results show that, although further implementation optimizations are needed, the IBC-enabled VPN performance is enough to transport CCTV video streams. Also, although enabling HTTPS has an associated performance hit to secure web servers, TLS connection reuse greatly reduces it, even with mutual authentication, and thus enabling the proposed secure certificate-based authentication in LEA web servers.

In emergency scenarios or other environments with limited communication capabilities, mobile ad hoc network (MANET) technologies may be employed by LEA officers and other first responders. In these cases, the rank-based secure ad hoc on-demand multi-path distance vector (RSAOMDV) routing protocol could be employed to provide secure and reliable communications that are not vulnerable to logical or physical attacks. In particular, by finding all possible node-disjoint paths, RSAOMDV is able to withstand insider attacks with a high number of attackers that would disrupt other secure single-path routing protocols in the literature.

Finally, a complete use case example has been presented in order to illustrate the different components and common security operations of the proposed LEA security architecture.

Acknowledgments This research has been partially 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. Adams C, Lloyd S (2002) "Understanding PKI: Concepts, Standards, and Deployment Considerations". 2nd edn, Addison Wesley, ISBN 0-672-32391-5
2. Apostolopoulos G, Peris V, Saha D (1999) "Transport layer security: how much does it really cost?". Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99) vol 2:717–725, New York, USA
3. Coarfa C, Druschel P, Wallach DS (2006) Performance analysis of TLS web servers". ACM Trans Comput Syst 24(1):39–69
4. EJBCA PKI website. http://ejbca.org. Accessed 1 February 2013
5. Failner M, Graf N (2009) "Beginning openvpn 2.0.9". Packt publishing, Birmingham
6. Guerrero-Zapata M (2006) "Secure Ad hoc On-demand Distance Vector (SAOMDV) Routing". IETF Draft draft-guerrero-manet-saodv-06.txt
7. INDECT Project website. http://www.indect-project.eu. Accessed 1 February 2013
8. INDECT Consortium (2013) "D8.7: Definition of mechanisms and procedures for the security and privacy of the exchanged information".

9. Java in Simulation Time / scalable Wireless Ad Hoc Network simulator (JiST/SWANS). http://jist.ece.cornell.edu. Accessed 19 March 2014
10. Kleppe H (2011) Performance impact of deploying HTTPS. Technical Report, Universiteit van Amsterdam
11. Marina MK, Das SR (2001) "On-demand multipath distance vector routing in ad hoc networks". 9th International Conference on Network Protocols, pp. 11–14
12. Niemiec M, Dudek J, Romański Ł, Święty M (2012) "Towards Hardware Implementation of INDECT Block Cipher". MCSS 2012, CCIS 287, Springer-Verlag Berlin, pp. 252–261, ISSN 1865–0929
13. Niemiec M, Machowski L (2012) "A new symmetric block cipher based on key-dependent S-boxes". International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2012), Saint Petersburg
14. Niemiec M, Pach AR (2012) "The measure of security in quantum cryptography". IEEE Global Telecommunications Conference (GLOBECOM 2012), Anaheim, USA
15. Niemiec M, Romański Ł, Święty M (2011) "Quantum cryptography protocol simulator". IEEE Multimedia Communications, Services and Security (MCSS 2011), Krakow, Poland
16. OpenCT website. https://github.com/OpenSC/openct/wiki. Accessed 11 December 2013
17. OpenSC website. https://github.com/OpenSC/OpenSC/wiki. Accessed 11 December 2013
18. OpenVPN Community Software. http://openvpn.net/index.php/open-source.html. Accessed 11 December 2013
19. Rezac F, Voznak M, Tomala K, Rozhon J, Vychodil J (2011) "Security Analysis System to Detect Threats on a SIP VoIP Infrastructure Elements". Advances in Electrical and Electronic Engineering vol 9, num. 5: 225–232. ISSN 1336–1376
20. Stoianov N, Urueña M, Niemiec M, Machník P, Maestro G (2012) "Security infrastructures: towards the INDECT system security". IEEE Multimedia Communications, Services and Security (MCSS 2012), Krakow
21. StrongSwan website. http://www.strongswan.org. Accessed 11 December 2013
22. Urueña M, Machník P, Martínez MJ, Niemiec M, Stoianov N (2010) "INDECT advanced security requirements". IEEE Multimedia Communications, Services and Security (MCSS 2010), Krakow

Manuel Urueña received his M.Sc. degree in Computer Science from Universidad Politécnica de Madrid in 2001, and his Ph.D. degree in Telecommunication Technologies from Universidad Carlos III de Madrid in 2005. At present, he is an associate professor at the Telematics Engineering department of Universidad Carlos III de Madrid. His research activities range from P2P systems, through load balancing and service discovery protocols, to optical networks. He has been involved in several national and international research projects related with these topics, including the EU IST GCAP and the EU SEC INDECT projects.

Petr Machník received his M.Sc. and Ph.D. degrees in Telecommunications from VSB – Technical University of Ostrava, Ostrava, Czech Republic in 2004 and 2008, respectively. Presently, he is an assistant professor at the Department of Telecommunications, VSB – Technical University of Ostrava. His research activities are focused on computer, telecommunication and wireless networks and communication security. He has participated in several national and international research projects related to these topics, including INDECT project of the EU 7th Framework Program.

Marcin Niemiec received the M.S. and Ph.D. degrees in Telecommunications from AGH University of Science and Technology, Krakow, Poland in 2005 and 2011, respectively. Also, he studied in Universidad Carlos III de Madrid. He is an assistant professor at the Department of Telecommunications, AGH University of Science and Technology. His research interests focus on security and data protection, especially: security services, symmetric ciphers, cryptanalysis, malware, intrusion detection, and quantum cryptography. Co-organizer of many international meetings, workshops, and conferences. He has actively participated in 6th and 7th FP European programs (ePhoton/ONE+, BONE, SmoothIT, INDECT), Eureka-Celtic (DESYME) and several national projects. He co-authored over 50 publications and reports (papers, deliverables, book reviews, IETF draft, and book). He is a member of the IEEE.

Nikolai Stoianov received his M.Sc. degree in Computer Science from Shoumen Military School (University), Shoumen, Bulgaria in 1997, M.Sc degree in International and National Security and Defence in 2008 and M.Sc. degree in Organization and Management of Communication and Information Systems from "G.S. Rakovski" National Defence Academy (University), Sofia Bulgaria. He received his PhD degree in Security in information systems in 2004 from Defence Advanced Research Institute, Sofia, Bulgaria. At present, he is an associate professor in Cyber Security and Cryptology at Bulgarian Defence Institute (Research Institute), Sofia, Bulgaria and a part-time associate professor in Information Security and Applied Cryptology at University of Library Studies and Information Technologies. His research activities focus on information security, cryptography, cryptanalysis, key management systems, Public Key Infrastructures, symmetric ciphers, hash functions, cyber security and information assurance. He had participated in several national and international research project related to these topics, including EU SEC INDECT project. He co-authored 2 books and approx. 80 journal papers, publications and reports. He is member of "Bulgarian Union of Scientists", "AFCEA" Sofia Chapter and MENSA.