View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2000 Proceedings

European Conference on Information Systems (ECIS)

2000

Towards a Secure Web Based Health Care Application

Susanne Roehrig
Swiss Informationstechnik

Konstantin Knorr
University of Zurich

Follow this and additional works at: http://aisel.aisnet.org/ecis2000

Recommended Citation

Roehrig, Susanne and Knorr, Konstantin, "Towards a Secure Web Based Health Care Application" (2000). *ECIS 2000 Proceedings*. 136.
http://aisel.aisnet.org/ecis2000/136

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

search

Towards a Secure Web-Based Health Care Application

Susanne Röhrig
Swiss Informationstechnik AG
Diggelmannstrasse 22
CH – 8047 Zurich, Switzerland

Konstantin Knorr
University of Zurich, Department of Computer Science
Winterthurerstrasse 190
CH – 8057 Zurich, Switzerland

***Abstract*—Even though security requirements in health care are traditionally high, most computerized health care applications lack sophisticated security measures or focus only on single security objectives. This paper describes special security problems that arise when processing health care data using public networks such as the Internet. It proposes a structured approach using a context-dependent access control mechanism over the Internet as well as other security mechanisms to counter the threats against the major security objectives: confidentiality, integrity, availability, and accountability. The feasibility of the proposed security measures is shown through a prototype, which has been developed in a research project focussed on security in health care.**

***Keywords*—Health Care, Internet, Security, Workflow Management**

I. Introduction

In health care a lot of data are generated that in turn will have to be accessed from several departments of a hospital. The information kept within the information system of a hospital includes sensitive personal data that reveal the most intimate aspects of an individual's life. Therefore, it is extremely important to regard data protection laws, privacy regulations, and other security requirements. When designing information systems for health care purposes it is an imperative to implement appropriate access control mechanisms and other safeguards.

Furthermore, a tendency to use the Internet as a communications media can be observed. As the Internet is an insecure transmission media the security requirements that must be met by the overall system are high.

Reference [1] gives a general introduction of privacy aspects and security measures in health care. In [2], [3], and [4] distributed systems and networks (e.g. the Internet) in health care are presented regarding security aspects but focussing on confidentiality and privacy.

Unlike most approaches published today we take a general and structured outlook on the problem and take into account the four major security objectives: confidentiality, integrity, availability, and accountability. These objectives and their special meaning in health care systems are regarded, specifically for data transmission and storage.

During the project MobiMed[1] a prototype was developed to show the feasibility of the implementation of security mechanisms required in a web-based health care application.

This paper describes the specific security requirements in health care environments focussing on the additional security demands resulting from the use of the Internet as a communications media.

The remainder is organized as follows: After describing the security objectives and their meaning in health care in section II, the added security requirements when using the Internet are discussed in section III. General security measures are described in section IV. Subsequently, in sections V and VI, the application scenario and the implementation of the prototype are illustrated focussing on the security measures taken. Section VII summarizes the results achieved, while section VIII describes further research topics.

II. General Security Objectives

Generally, when talking about data security – in health care as well as in other areas – the three *objectives* confidentiality, integrity, and availability are identified. This division dates back to 1980 and was described in [5]. In this section we will describe these security objectives and illustrate their specific meaning in a health care environment.

Confidentiality – Confidentiality is defined as the state that exists when data is held in confidence and is protected from unauthorized disclosure. If the content of a communication is disclosed to an unauthorized person or even if the fact that a communication between two persons took place is made public, this is considered a loss of confidentiality.

In health care many sensitive data are processed, leading to high requirements on the confidentiality of data. Traditionally, the need for confidentiality stems from national data protection legislation and the professional secret of medical staff. Most data protection laws also state that data may only be used for the purpose they were collected for, e.g. data collected during a medical treatment may not be used for marketing purposes.

Integrity – Integrity is the state that exists when data has not been tampered with or has been computed correctly from source data and not been exposed to accidental or malicious alteration or destruction. Erroneous input and fictitious additions are also considered violations of data integrity. The demand for integrity is also included in some data protection laws (e.g. Switzerland).

The modification (whether with malicious intent or due to a program failure) of data stored within a health care application may lead to a mistreatment of the respective patient which could be hazardous to his health. In a case published in [6] an attack on a hospital information system in Liverpool was described. The attacker changed the data of prescriptions. A patient was saved only because a nurse re-checked the prescription and did not administer the prescribed medication realizing that it was lethally toxic.

[1] Privacy and Efficiency of Mobile Medical Systems. Further information about the project can be found at http://www.ifi.unizh.ch/ikm/MobiMed/

Availability – When all required services and data can be obtained within an acceptable period of time the system is called available.

Data concerning a patient have to be available at any time, to prevent loss of time in case of emergencies, since a treatment without knowing certain medical data could harm the patient's health. [7] describes what may happen if the availability of a health care system is not assured, citing the collapse of the London Ambulance Service in October and November 1992. Due to the overload and collapse of a new computerized dispatching system, London was left with partial or no ambulance cover for longer periods, which is believed to have led to a loss of about 20 lives.

Recently, more security objectives have been identified to better suit the needs of electronic commerce systems with all their legal aspects. The most important one is accountability.

Accountability – If the accountability of a system is guaranteed, every participant of a communication can be sure that his partner is exactly the one he or she pretends to be. This allows holding users responsible for their actions.

In health care accountability mechanisms like that are necessary because it is important to know who performed a certain service at a certain time. Today, manual signatures of the responsible person ensure accountability.

III. Additional Requirements for Distributed Applications

When transferring medical data over the Internet, additional security requirements arise This mainly touches the following two areas:

Secure data transfer: If data are collected at different sites the need for data transfer arises when the data are processed at a central location. The data transmitted might be subject to certain risks – especially if public networks (e.g. the Internet) are used as transportation media.

Secure data storage: If the sites of data collection and of data processing are different, it has to be ensured that the storage on the other side of the communication is authorized. This depends on data protection laws as well as the professional secret in health care, which is embodied in the penal legislation of many countries. Also the other security objectives have to be regarded.

Moreover, the computers taking part in data transfer have to meet the same security requirements as any other computer used in health care. However, these have been described thoroughly (e.g. [8] proposes nine principles of data security for clinical information systems) and are not topic of this paper.

In the following subsections the aspects "secure data transfer" and "secure data storage" are regarded more closely.

A. *Secure Data Transfer*

Additional security requirements when transferring medical data over the Internet – organized according to the security objectives identified in section II. – are as follows.

Confidentiality – When data is exchanged over the Internet it is generally accessible to everybody who has access to the network. The main threats to confidentiality for data on the Internet are presented in [9].

Integrity – Data transferred over the Internet not only are read easily but are also manipulated as easily. Even though the data are protected against transmission errors on the lower layers of the TCP/IP[2] protocol family, intentional damage can not be prevented as an attacker may easily recompute checksums to forge data [9].

Availability – To ensure a certain availability the Internet was build redundantly. However, the current implementations of TCP/IP allow attackers to disturb the operation of computers or parts of the network, e.g. through denial-of-service (DoS) attacks like SYN-flooding [9]. A well-known example of how the availability of a major part of the Internet may be compromised is the *Morris Worm Incident* of November 1988 where a self-replicating program used design flaws of BSD-derived versions of UNIX and disrupted normal Internet connectivity for days [10].

Accountability – In a distributed system especially, problems identifying the originator of a message might occur. On the Internet the techniques to produce a wrong IP address of the sender's computer are well known (so-called IP spoofing). Moreover, the address could be changed any time during the transmission.

B. *Secure Data Storage*

More security requirements arise, when collecting data at one site and storing them centrally at another.

Confidentiality – When medical data are collected at one site and stored at another – as presented in the scenario in section V – the regulations of the data protection legislation have to be followed.

In Switzerland the use of medical data for research purposes is allowed only in specific cases that are listed in [11]. In contrast, the security policy of the British NHS[3] states that a patient in treatment implicitly agrees to the use of his or her data for research purposes [8].

The use of anonymized data, however, is allowed under most laws; anonymized meaning that the patient cannot be re-identified from the data stored. We therefore conclude that – though they have different focuses – the national data protection legislation has to be followed, i.e. as few data as possible have to be collected about a patient and the access to them has to be as restricted as possible.

Integrity – The data stored at the receiver's side has to be kept in a state of integrity, and the user sending these data has to trust that they are not changed. An attacker might use a known system vulnerability to gain access to the system, where he could modify data.

Additional integrity may be achieved by the use of plausibility controls before storing.

Availability – The availability of data and services at the storing side has to be kept up at any time, as the users in the hospitals must have access in case any problems occur. Recent incidents where the availability of commercial Internet servers was disturbed are the distributed denial-of-service attacks against Internet merchants in February 2000. Among others the Internet bookstore amazon.com and the

[2] Transmission Control Protocol / Internet Protocol

[3] National Health Service

Internet auctioneer ebay.com could not be accessed during several hours [12].

Accountability – The server where data is stored has to prove to the users that it is the one it pretends to be. Otherwise an attacker might run a denial-of-service attack, take the regular server out of order, and pretend to be this server – consequently receiving data not intended for it.

IV. General Security Measures

To counter the risks described above, appropriate security measures have to be implemented.

Protection of confidentiality – Access control mechanisms, encryption techniques, or anonymization are generally used to protect confidentiality. Access control mechanisms grant or restrict access to data or applications on either application or operating system layer. Encryption techniques are used to prevent unauthorized persons from reading data not intended for them. Anonymization means that any reference to the concerned person is removed from the data.

Protection of integrity – The integrity of data (e.g. messages) can be guaranteed by the use of cryptographic check sums (so-called message authentication codes). A hash value of the message is calculated and appended. The receiver again calculates the hash value of the message. By comparison with the sent value, transmission errors can be detected. In connection with public-key encryption such hash values may be used as digital signatures.

Protection of availability – Within a distributed system the availability of the computers and the communications media must be guaranteed. Measures to protect the availability of a system are either of organizational and technical nature or concern the surrounding infrastructure. Organizational and technical measures include:

- appropriate configuration of any hard- and software used on the system,
- regular use of security software (such as scanners or intrusion detection systems),
- protection of the whole system from outside attack by using a well-configured firewall system,
- rules concerning passwords,
- back up frequencies and the secure preservation of back up media, and
- documented procedures what to do in case of security violations.

Measures concerning the infrastructure are e.g. the safe placement of server systems and cables as well as measures for fire protection.

In addition, one principle design concept of the Internet was redundancy, i.e. the data packets may find multiple ways from sender to recipient, so that one damaged routing element might not disturb the whole communication chain. Thus, redundancy increases the availability of the system.

Protection of accountability – Measures to guard the accountability within a system are log files – where all activity is recorded – and digital signatures together with appropriate public-key infrastructures and certificates.

Since the confidentiality is traditionally the most important security aspect of medical data, the main focus of the security measures described in more detail in section VI lies on the protection of the confidentiality of the data stored and transmitted.

The following paragraphs depict access rights, encryption, and certification, which are also the main security mechanisms implemented in the MobiMed prototype.

A. Access Control Mechanisms

A well-known system to grant or deny access rights is the use of role-based access control together with an access control matrix. Such a matrix is built in a way that the objects to be accessed (i.e. the data or the applications of the system) are listed on one axis and the roles on the other axis. The access rights are organized as the matrix entries. Each user is assigned one or more roles; he or she is granted access only if one of his or her roles has access to the requested data.

This approach can be enhanced if not only the user's role but also the state of the process of work is considered when granting or denying access rights. Hence, the access is only granted to persons who – at a given time – need this data to accomplish their tasks. The access control becomes dependent on the context. This concept dates back to [13]; [14] suggests granting access rights according to the context of a business process. A necessary prerequisite the analysis and modeling of the underlying processes. This modeling is done using a workflow management system[4]. In comparison to an access control system based purely on role-based access matrices, this scheme offers additional protection against insiders trying to misuse stored data.

A system using workflow states as the basis to grant access rights was described in [16] using the ActionWorkflow System [17].

B. Encryption

To protect data from being read by unauthorized persons, encryption techniques can be used. Two different classes of encryption are distinguished: symmetric and asymmetric algorithms.

When using a symmetric algorithm both parties of a communication have to agree beforehand on one common key, which is used for both encrypting and decrypting the data. Common symmetric encryption schemes are DES[5] and IDEA[6]. An advantage of symmetric encryption is that it is rather fast; a disadvantage, however, is the difficult key management as keys have to be agreed upon by each pair of potential communication partners and to be distributed in a secure manner.

An asymmetric algorithm demands that each user owns his or her own pair of keys: a private key, which is kept secret and known only by its user, and a public key, which is published and can be accessed by everyone who wants to communicate with the key's owner. Those schemes are called public-key algorithms. A well-known public-key algorithm is RSA[7]. When a message is encrypted with one of the keys, it can only be decrypted using the other key. This feature can be used to implement confidentiality as

[4] An Overview of Workflow Management Systems is given in [15].

[5] Data Encryption Standard

[6] International Data Encryption Algorithm

[7] named after its inventors: Ron Rivest, Adi Shamir, and Leonard Adleman

well as accountability: If a message has to be kept secret, it will be encrypted using the recipient's public key, so that only he can decrypt it, as no-one else knows his private key. If it has to be proven that a certain person sent a message, she will encrypt it using her private key; everybody who decrypts it will use her public key, thereby ensuring that it was she who wrote the message. A disadvantage of public-key algorithms is that they are rather slow to compute, whereas the key management easier compared to symmetric algorithms.

To combine the advantages of symmetric and asymmetric algorithms, so-called hybrid techniques are used. An asymmetric scheme is used to negotiate a symmetric session key, which is later used to encrypt the data being sent. This combines the simple key management of asymmetric techniques and the fast encryption of the data being sent with symmetric techniques.

A thorough description of cryptographic algorithms can be found in [18].

C. *Certification*

To establish trust when using public-key crypto-systems, it is necessary that the users are given guarantee that a public key really belongs to the person the key is issued to. Certificates issued by a trusted third party like a CA[8] offer a solution.

Such an authority signs the users' keys with its own private key, thereby certifying that the information about the user is correct and that the public-key really belongs to the respective user. Another user can now check the certificate by verifying the signature, thus ensuring that the key really belongs to its user. Of course, the CA must be a trustworthy entity that certifies keys only after the user has identified himself. The CA's own public key may in turn be signed by another CA, thereby creating either a hierarchy of CAs or a cross-certification structure. An infrastructure where trust is established and users can access and check other user's keys is called a PKI[9].

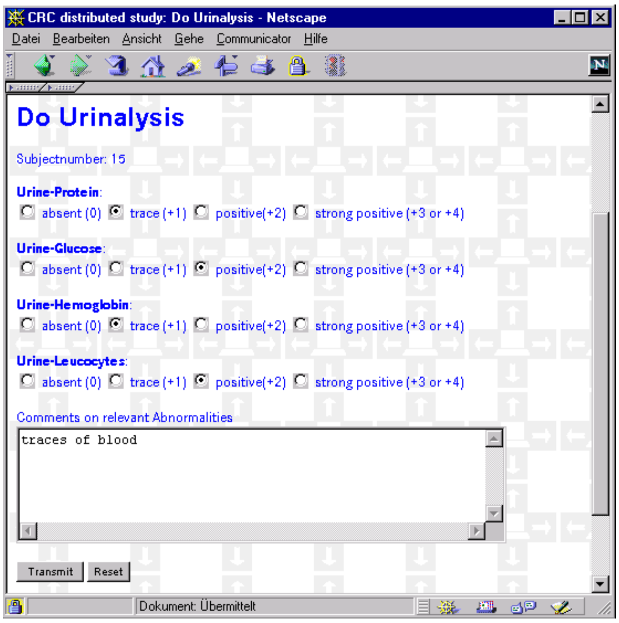
Another task of a CA is the management of cryptographic keys. More information about certification and public key infrastructures can be found in [19]. The principles of key management in general are described in [20].

The structure of certificates issued by CAs is standardized. One important standard is the X.509 authentication framework to support the X.500 directory services [19]. An X.509 certificate comprises among other data a certificate serial number, the identifier of the signature algorithm, the certificate's validity period, the user's name, and the user's public key information. All this is signed with the CA's private key.

V. The Application Scenario

A. *The Clinical Research Cycle*

The application developed during our project supports a clinical study – a so-called *Clinical Research Cycle* (CRC).


CRC distributed study: Do Urinalysis - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
Transmit
Reset
Dokument: Übermittelt

Do Urinalysis

Subjectnumber: 15

Urine-Protein:
○ absent (0) ● trace (+1) ○ positive(+2) ○ strong positive (+3 or +4)

Urine-Glucose:
○ absent (0) ○ trace (+1) ● positive(+2) ○ strong positive (+3 or +4)

Urine-Hemoglobin:
○ absent (0) ● trace (+1) ○ positive(+2) ○ strong positive (+3 or +4)

Urine-Leucocytes:
○ absent (0) ○ trace (+1) ● positive(+2) ○ strong positive (+3 or +4)

Comments on relevant Abnormalities

traces of blood

Fig. 1: Screenshot of the prototype

In the scope of this study a new medication is tested on volunteers within a hospital.

This test is carried out in four steps:

1. During the first part the test person is checked, whether he is eligible for the study. For this end, certain questions concerning the patient's anamnesis, inclusion and exclusion criteria are examined.
2. Consequently, a pre-examination is carried out recording a laboratory analysis of the patient's blood and other tests before the patient is treated with the medication to be tested.
3. Subsequently, the medication is administered and the same tests as in step 2 are carried out.
4. During a post-examination a few weeks after the medication, the same tests are carried out for a third time.

This practice allows to analyze the impact of the medication on the patient's health. The test results are then sent for analysis to the pharmaceutical company who commissioned the test.

Traditionally, the CRC is carried out with paper-based questionnaires. This procedure is error-prone and tedious, because large amounts of paper are used and the data inserted in the questionnaires has to be re-entered manually into an electronic system at the pharmaceutical company. In our prototype the questionnaires are electronic – to be more specific they have been implemented as HTML[10] forms with text areas, radio buttons, and select boxes as input options. A sample screenshot of an input mask is shown in fig. 1.

In our scenario the data are collected at several hospitals and are sent to a central WWW and database-server to be stored there. No data of the CRC are kept at the hospitals.

This process is a good example for distributed processing as the test may be carried out not only in one but in several hospitals, e.g. to find an appropriate number of test persons.

[8] Certification Authority

[9] Public Key Infrastructure

[10] Hypertext Markup Language

search

When processing data in a distributed manner, the data protection and security requirements arise, that are topic of this paper.

The questionnaires of the CRC contain highly sensitive data concerning the patient like his history of drug abuse and mental condition. Therefore it is very important to regard data protection regulations.

A patient volunteers to participate in the CRC. Before any examination he explicitly consents that his data may be stored. In this consent the patient acknowledges that his anonymized medical data can be used for the given purpose. Only data raised during the CRC are stored in the prototype's database.

B. *The User Interface of the Prototype*

The prototype developed during our research project supports the execution of a CRC described in the last subsection. This subsection briefly discusses the user interface and the operational framework of the prototype.

The user interface is completely based on HTML so that standard web browsers can be used. To use the prototype a user has to enter the URL[11] of the prototype's web site in his browser.

When a user logs onto the system (user name and password are required) a menu offers the following alternatives:

1. Browse the description of the underlying CRC. Both the process and the logical and temporal order of the questionnaires are explained.
2. Show the work list depending on the user's role (e.g. "doctor" or "nurse"). Following a hyperlink in the work list automatically opens the corresponding questionnaire. Figure 1 shows an example. Each entry in the work list represents a patient (identified through his initials and date of birth) and the possible steps of the examination.
3. Browse data that already exists (from other questionnaires in the study) if access rights are granted (cf. section VI.A).

VI. Implementation

In sections II, III, and IV security was examined in a very thorough but abstract way. Practical considerations will be the topic of this section. The implementation of the security measures is shown according to the prototype developed in our research project.

The architecture of the prototype is shown in fig. 2. It is a typical client-server application. The client may be any web browser capable of SSL[12] encryption and decryption. The server side consists of a WWW[13] server, several Perl[14] scripts [21], and a DBMS[15]. The database stores the medical data associated with the test of the new medication and the workflow data (more precisely the model of the underlying process) needed for the context-dependent access control (cf. section IV.A.). The Perl scripts act like a "glue layer" between the WWW and the database server. The following example illustrates the way the different components of the prototype work together. A typical client request is treated in six steps:

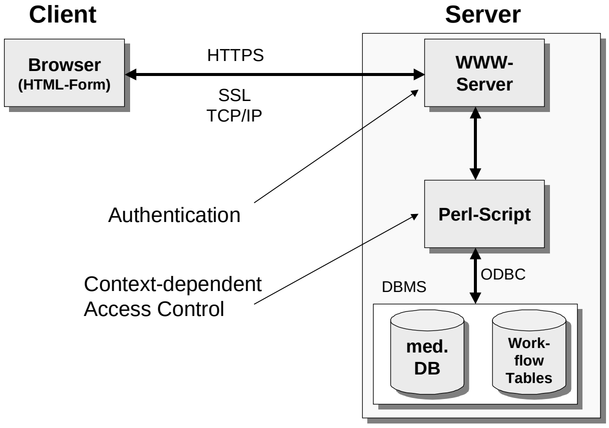


Fig. 2: The prototype's architecture

1. The client requests some data from the medical database. This request is done via a special HTML form. The data packets exchanged between browser and server are SSL-encrypted.
2. The WWW server receives the request and forwards the relevant entries in the HTML form to a Perl script. The whole communication between WWW server and the Perl script makes use of CGI[16]-technology.
3. The Perl script generates an SQL[17] statement out of the different entries in the HTML form, establishes an ODBC[18] connection to the database server, and transfers the SQL statement to the DBMS.
4. The database server executes the SQL statement and returns the result containing the requested data items to the Perl script.
5. The Perl script "translates" the result of the SQL statements to HTML and forwards the HTML output to the WWW server.
6. The WWW server hands the dynamically generated HTML page back to the browser. The data are SSL-encrypted, again.

The server side of the application is running on a single PC using the operating system Windows NT 4.0. The WWW server is the Internet Information Server 4.0 and the DBMS is SQL Server 6.5. Perl version 5 is used.

After this rather technical description of the prototype its main security mechanisms are considered in detail.

How these mechanisms correspond to the four security objectives – confidentiality, integrity, availability, and accountability – is shown in table 1. A bullet (•) indicates that the security mechanism implements the security objective.

A. *Workflow-based access rights*

In our prototype access rights are granted according to the state of the underlying CRC. For this a model of the process

[11] Uniform Ressource Locator
[12] Secure Sockets Layer
[13] World Wide Web
[14] Practical Extraction and Report Language
[15] Database Management System
[16] Common Gateway Interface
[17] Structured Query Language
[18] Open Database Connectivity

TABLE 1:
SECURITY OBJECTIVES IMPLEMENTED IN THE PROTOTYPE

		security objectives			
		confidentiality	integrity	availability	accountability
security mechanisms	workflow-based access rights	●			
	SSL	●	●		●
	authentication	●			●
	log files				●
	further security measures	●	●	●	

has to be created. Such a model is called a workflow specification. The main issues of such a specification are [22]:

- activities,
- data items,
- participants,
- control flow, and
- data flow.

Note that the workflow specification has to be defined before the workflow is executed. During the execution many instances of the workflow model are generated. In our prototype one workflow instance is created for each patient.

In the CRC the activities correspond to the questionnaires to be filled (fig. 1 shows the activity “Do Urinalysis”). The data items are the information contained in the questionnaires; the participants are the three user roles “doctor”, “nurse”, and “pharma monitor”. The control flow determines in which sequence the activities must be executed, whereas the data flow states which data items – already existing – are needed to carry out a new activity.

The state of a workflow describes which activities have been performed and which activities are ready for execution. For better handling, the states of the CRC have been numbered in our prototype.

The context-dependent access control is done using a database table which consists of the four *entries* user role, name of the questionnaire, state of the workflow, and the kind of access right. For instance, (nurses, 3, urinalysis, write) is a possible tuple in the table, i.e., a nurse has write permission on the database table which contains the laboratory data of the urine of a patient, if the workflow of the patient is in phase three of the CRC.

We use this strict means of access control to ensure the confidentiality in our system, thus not only keeping outsiders from viewing the data but also to restrict insiders to use the data only when necessary, i.e. the access to the data is as restricted as possible.

B. SSL

The CRC can be executed in different hospitals. The Internet – a public network – is used to communicate with the central database of the pharmaceutical company. As stated before, the Internet is a highly insecure network. Specific security considerations have to be made. To protect the integrity and authenticity of data packets on the Internet a special security protocol is used, called SSL.

SSL provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection [23] and is located above TCP/IP and below HTTP[19] on the protocol stack. A hybrid encryption scheme is used, typically the keys have a length of 512 bits (asymmetric) and 40 bits (symmetric). To prove the authenticity of the server (possibly of the client, too) a certificate from a trusted third party must be obtained. For more information about SSL see [23].

For our prototype a certificate by the Swiss CA *SwissKey* was obtained, which proves that the web-server is run by a trusted entity known to the CA.

We use SSL not only to keep the communication confidential but also to prove that the server is the same it claims to be – thus enforcing accountability within the system. The use of SSL also enhances the integrity of the data transmitted through message authentication codes.

C. Authentication

Only legitimate users are allowed to communicate with the WWW server of the prototype. To prove their legitimization users have to provide a valid login name and password every time they start a session with the server. Both username and password are transmitted in an encrypted way – using the SSL connection – so that no attacker can easily gain a username and a password by intercepting the communication. Authentication is a measure to increase confidentiality and accountability of our application.

Users should be encouraged to choose a password which cannot be guessed easily, because an attacker gaining knowledge of a valid user password can act on behalf of that user [24].

D. Log Files

All user activity dealing with sensitive medical data is logged by the application. Unsuccessful or attempted data access is logged, too. The log file contains the following entries:

- the name and role of the user,
- the time and date of access,
- the performed activity,
- the IP address and domain name (if available) of the client,
- the patient identification number,
- the phase to the patient’s workflow at the access, and
- the accessed database table.

The log file can be used to provide evidence what data was requested, accessed, or inserted by a user at a specific time. This is used to increase the system’s accountability. Furthermore log files can be used to do load management, if several servers are involved, to do recovery if an unexpected error occurred, and to fulfill certain legal documentation requirements.

[19] Hypertext Transfer Protocol

E. *Anonymization*

In a clinical trial – supported by our prototype and used in order to facilitate a test of a contrast medium in a Swiss hospital – no personal patient data at all is submitted to the web-server (and consecutively to the pharmaceutical company). The patient's personal information is kept at the hospital. On the server only a unique identifier is used. The medical data cannot be matched to the person's name and demographic data. Therefore, all data protection requirements are met.

F. *Further Security Measures*

The data to be protected in our prototype are stored in a central database. This database is located at a secure site distinct from the hospitals where the test of the new medication is done. There is a physical separation of the locations where data are raised and stored. Hence, it is very difficult for a user to tamper with data he has already transferred to the database.

The security of a server system as well as the applications on the server can be tested through special security scanners. One prominent example is the scanner distributed by *Internet Security Systems* [25]. To exclude as many known system vulnerabilities as possible, such a security scan has been performed on the computer hosting both WWW- and database-server – increasing the availability of the system.

Furthermore, when submitting data to the server, they are checked for completeness, thus enhancing the integrity of the system.

When using our prototype in practice, it is very important to implement a system to back up data to minimize the risk of data loss. Most database systems provide means to support data back up. If the WWW server has a high workload, back up or stand-by servers may be necessary, too.

The use of CGI-scripts may provide opportunities to attack the system. Special "script scanner software" – like Whisker – is available to check all scripts on a server for these vulnerabilities [26]. More ideas to enhance the prototype's security by implementing without the use of CGI-scripts are described in [27].

The **confidentiality** on the server is ensured by a sophisticated access control mechanism; encryption protects the confidentiality during the transfer of the data.

Data **integrity** is provided for by plausibility checks on the server side and by message authentication codes offered by SSL during the transmission.

Physical protection of the server and the use of special security software to scan the system for vulnerabilities enhance the **availability** of the system.

The use of SSL and certificates, login procedures, and the use of a log file ensure the system's **accountability**.

Table 1 gives an overview of the security measures implemented in the prototype and the security objectives aimed at.

To sum up, the prototype implements different security mechanisms to counter the four major security objectives in a multi-level security architecture.

VII. Conclusion

In this paper we have concentrated on security requirements of web-based health care applications. The main focus has been laid on a systematical presentation of the security objectives: confidentiality, integrity, availability, and accountability.

These objectives have been defined, and their implications on data transfer and data storage have been shown. The implementation of the security requirements was illustrated with our prototype, an Internet-based application supporting the data management associated with the testing of a new medication in a hospital. The prototype's main security features are SSL encryption of the communication between client (WWW browser) and server (WWW and database server), password authentication at the WWW server, the use of log-files, and the so-called context-dependent access control which yields additional protection for the sensitive medical data in the database.

VIII. Future Research

Future research will focus on the following topics:

1. The presented security requirements and the aspects of data transfer and data storage will be presented and extended in a structured matrix, yielding a systematic model to organize and measure the security of distributed health care applications.
2. The prototype could furthermore be extended with digital signatures to permit the so-called "four eyes principle", which is very important in health care scenarios. To introduce digital signatures in the prototype, a public-key infrastructure has to be established. Also, elaborate authentication mechanisms like smart cards or biometric devices for authentication purposes will be considered assuming that the necessary browser interfaces will be provided.
3. A third research topic will be the legal requirements of health care applications in different European countries. A typical example is the legal status of digital signatures, which varies from country to country.

IX. Acknowledgments

We would like to thank Prof. Bauknecht and the Information and Communications Management group of the University of Zurich for their help and support.

The work of the authors was funded by the Swiss National Science Foundation under grant no. 45359 as a part of the Swiss Priority Programme (SPP) Information and Communication Systems.

Bibliography

[1] Thomas C. Rindfleisch: Privacy, Information Technology, and Health Care, *Communications of the ACM*, 1997, No. 8, Vol. 40, p. 93-100

[2] Baker D. B., Masys D.R.: PCASSO: a design for secure communication of personal health information via the Internet., *International Journal of Medical Informatics*, May 1999; 54(2):97-104

[3] David M. Rind et al.: Maintaining the Confidentiality of Medical Records Shared over the Internet and the World Wide Web, *Annals of Internal Medicine*, 15 July 1997, 127:138-141

[4] B. Blobel: Security Requirements and Solutions in Distributed Electronic Health Records, in *Proceedings of the IFIP/Sec '97*, p. 377-390, 1997

search

[5] *Guidelines for Security of Computer Application*, Federal Information Processing Standards Publication 73, Department of Commerce, National Bureau of Standards, June 1980.

[6] *Der Spiegel*, Nr. 9, 1994, S. 243

[7] Ross J. Anderson: *Information technology in medical practice: safety and privacy lessons from the United Kingdom*, to appear in: Australian Medical Journal, http://www.cl.cam.ac.uk/users/rja14/austmedjour/austmedjour.html

[8] Ross J. Anderson: *Security in Clinical Information Systems, British Medical Association*, Tech. Report, London, 1996.

[9] Daniela Damm, Philip Kirsch, Thomas Schlienger, Stephanie Teufel, Harald Weidner, Urs Zurfluh: *Rapid Secure Development – Ein Verfahren zur Definition eines Internet-Sicherheitskonzeptes*. Projektbericht SINUS. Institut für Informatik, Universität Zürich, Technical Report 99.01, Feb. 1999.

[10] Eugene H. Spafford, *The Internet Worm Program: An Analysis*, CS-TR-833, Department of Computer Science, Purdue University, West Lafayette, 1988

[11] *Leitfaden für die Bearbeitung von Personendaten im medizinischen Bereich. Bearbeitung von Personendaten durch private Personen und Bundesorgane*. Der Eidgenössische Datenschutzbeauftragte informiert, Bern, 1997.

[12] Spiegel Online, *Attackiert – Das Web im Fadenkreuz*, 09.02.2000, http://www.spiegel.de/netzwelt/technologie/0,1518,63447,00.html

[13] Roshan K. Thomas, Ravi S. Sandhu: Towards a task-based paradigm for flexible and adaptable access control in distributed applications, *Proceedings of the 1992-1993 ACM SIGSAC New Security Paradigms Workshop*, p. 138-142, Little Compton, RI, 1993

[14] Ralph Holbein: *Secure Information Exchange in Organizations – An Approach for Solving the Information Misuse Problem*, PhD Thesis, University of Zurich, 1996

[15] Dimitrios Georgakopoulos, Mark Hornick, Amith Sheth: An overview of workflow management: from process modeling to workflow automation infrastructure, *Distributed and Parallel Databases*, 3:119-153, 1995

[16] Ulrich Nitsche, Ralph Holbein, Othmar Morger, Stephanie Teufel: Realization of a Context-Depended Access Control Mechanism on a Commercial Plattform, *Proceedings of the IFIPSec '96*, Chapman & Hall.

[17] Raùl Medina-Mora, Terry Winograd, Rodrigo Flores, Fernando Flores: The Action Workflow Approach to Workflow Management Technology, *Proceedings of the 92 CSCW Conference*, November, 1992.

[18] Bruce Schneier: *Applied cryptography*. New York, Wiley 1994

[19] Marc Branchaud: *A Survey of Public-Key Infrastructures*, Master Thesis, McGill University, Montreal, 1997

[20] Walter Fumy, Peter Landrock: Principles of Key Management, *IEEE Journal on Selected Areas in Communication*, Vol. 11, No. 5, June 1993

[21] Larry Wall: *Programming Perl*, Cambridge, Mass., O'Reilly, 1996

[22] Stephan Jablonski, Markus Böhm, Wolfgang Schulze, (Hrsg.): *Workflow-Management*. Heidelberg, dpunkt Verlag für digitale Technologie, 1997

[23] *The SSL Protocol, Version 3.0*, Internet Draft, 1996, http://home.netscape.com/eng/ssl3/draft302.txt

[24] R. Morris, K. L. Thompson: Password security: a case history, *Communications of the ACM*, 22, 11 (Nov. 1979), pp. 594-597

[25] http://www.iss.net

[26] http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=2

[27] Ulrich Ultes-Nitsche, Stefanie Teufel, *Secure Access to Medical Data over the Internet*, ECIS 2000