

Secure Communication and Cooperation via Shared Workspaces

Wolfgang Appelt, Sanjin Pajo, Wolfgang Prinz
Fraunhofer Institute for Applied Information Technology
Schloß Birlinghoven, 53757 St. Augustin, Germany
appelt@fit.fraunhofer.de pajo@fit.fraunhofer.de prinz@fit.fraunhofer.de

Abstract—In this paper, we discuss security issues of Web based CSCW systems and describe a conceptual model as well a recent extension to the BSCW Shared Workspace System [2] that satisfies high security requirements for communication and cooperation in an inter-organizational setting by using encryption technology.

I. INTRODUCTION

Security issues for CSCW systems have been discussed since many years. (A comprehensive survey of related literature can be found in [1]). Already at the very first CSCW Conference Greif and Sarin [4] argued that the security requirements of CSCW systems are specific and not sufficiently supported by the security features provided on operation system or database level.

In practice, however, security issues are often ignored in CSCW systems. Ahmed and Tripathi argue that this is due to the fact that CSCW systems “emphasize the motivations of cooperation and shared objectives” [1] although it is well known that users of a CSCW system may have conflicting goals (see, e.g., [6] or [7]).

We believe that with the emergence of Web-based CSCW systems and new ways of work organization security issues become of high importance for the future acceptance and usage of CSCW system. For example, we increasingly observe that locally distributed, inter-organizational teams are formed to work on a common goal, e.g., to create joint documents or other artifacts. Members of such teams often do not know each other in person but only meet “virtually” (see, e.g. [9] and [10] for case studies of such types of cooperation). In order to establish a sufficient level of trust within such teams for a successful cooperation, appropriate security measures are necessary [8].

At least the following requirements should be satisfied:

- R1: Users should be able to verify each others identity, i.e., if a user claims to be person A , other users should have means to verify this statement.
- R2: Verification of identity should also relate to artifacts used in the cooperation process, i.e., users can verify that an artifact was created by a particular person A and this person cannot deny his/her authorship.
- R3: Only authorized persons have access to artifacts used in the cooperation process. In particular, if a user makes such an artifact available to others, he/she can be sure that no unauthorized person can access the artifact or if an unauthorized person would get access to it, he/she would not be able to make any sensible use of it.
- R4: Users are able to specify access constraints on artifacts, e.g., a user may specify that a document can only be read, but may not be modified or deleted by other users.

In this paper we first discuss a number of security issues that should be addressed by Web-based CSCW system where we address primarily systems based on the shared workspace metaphor. We then describe several extensions to the BSCW system that were required to satisfy these requirements. A technological approach alone, however, is not sufficient as first evaluations have shown. Implementing a security policy requires also the consideration of several organizational issues.

II. SECURITY ISSUES OF WEB BASED GROUPWARE SYSTEMS

The usual work mode in a Web-based CSCW system based on the shared workspace metaphor is, in general, as follows (see Fig. 1):

A set of users forms a team and creates a joint workspace, usually a folder hierarchy, where they create/modify/read/exchange documents or other artifacts required for achieving a joint work goal.

In general, access to these workspaces is limited to the members of the team, i.e., the users have to authenticate

The work described in this paper was partially supported by the European Union's IST project ECOSPACE (contract number 035208).

themselves, usually based on a username/password authentication mechanism.

All interactions between the users and the respective CSCW system are Web-based, i.e., on the server side there is a Web server extended with communication and cooperation features and on the users'/clients' side there is a Web browser or some other applications using the standard Web protocols, in particular, HTTP.

In particular, all data exchange between the users/clients and the server is carried out via standard Web protocols.

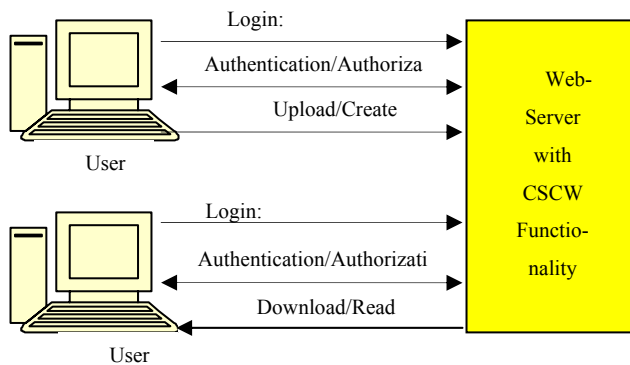


Figure 1. Web-based CSCW System.

In the following sub-section we shall discuss the security issues related to this form of communication and cooperation process.

A. User Authentication

The by large most common method of user identification in current Web-based systems is the user name/password scheme. In general, this authentication method cannot be considered secure as the many privacy violations known for this authentication method show.

With respect to the security requirements described above, two issues need to be distinguished:

Firstly, an unauthorized person who gets unintended knowledge of a user name/password combination gets access to a workspace and the information therein, i.e., a violation of R3.

Secondly, an unauthorized person who gets unintended knowledge of a user name/password combination gets access to a workspace and creates or modifies artifacts therein. Other members of the workspace may then associate these artifacts with the wrong person, i.e., this represents a problem for R1.

B. Data Transmission

The data from the local client to a Web server is usually transmitted via a considerable number of computers/routers in between of which a normal user is not aware. This is often considered a significant security risk, however, in practice this is, in principle, no problem. When client and server use an SSL communication channel, there exist hardly any security risks, e.g., that unauthorized persons get access to the

exchanged data. This applies both to the upload and download processes.

C. Trust in Data Security on the Server

From the users' perspective the data storage on the server side provides a major problem. Users may not even know the physical location of the server which whom they communicate nor the security measures that apply ("How is the system secured against hackers' attacks?" "Can the system administrator read my files?" etc.).

Even if the security measures on the server's side are very elaborate and provide a very high security level, it may be very difficult to create a "sufficient" level of trust on the users' side to deposit information they consider of high privacy on the server. This leads to the effect that a number of organizations have already established security policies that forbid its employees the storage of documents on external groupware systems. Obviously, such a policy represents a high barrier for the use of shared workspaces in cross-organizational collaboration processes.

D. Encryption and Digital Signatures

To address the issues discussed above and to satisfy the security requirements R1 – R4 we propose the following solution:

All artifacts in shared workspaces are encrypted.

All artifacts in shared workspace have a digital signature of its author.

Encryption and digital signatures are based on public/private key pairs.

In principle, these measures are sufficient for solving all security issues mentioned above except R4 (although in practice there exist some hurdles as will be discussed later):

R1 and R2: A user's public key is bound by a Certificate Authority to a particular person. Other users can access the Certificate Authority for verifying a user's identity. If all artifacts have a digital signature, the authors can be identified and they cannot deny their authorship.

R3 and user authentication: Users do not need to trust that the name/password authentication scheme is secure since encryption in combination with digital signatures prevent that intruders get access to the actual information of artifacts or deceive the authorized users about the origin of artifacts.

Data transmission and data storage: Similarly, users do neither have to trust the data transmission process nor the data storage on the server since encryption prevents any authorized access and any modification of the data during transmission or at the data store would invalidate the digital signature.

III. CONCEPTUAL MODEL FOR THE INTEGRATION OF KEY MANAGEMENT IN SHARED WORKSPACES

The conceptual model for the integration of security mechanisms in shared workspace systems – in particular the key management – shall be illustrated by a specific example shown in Fig. 2.

The ellipses denote the hierarchy of different shared workspaces. F1 is the root-folder with three subfolder F11-F13. F11 itself has three subfolders F111 and F112. The folders are shared among a group of users. These groups are illustrated by the hexagons (G1, G11, G12).

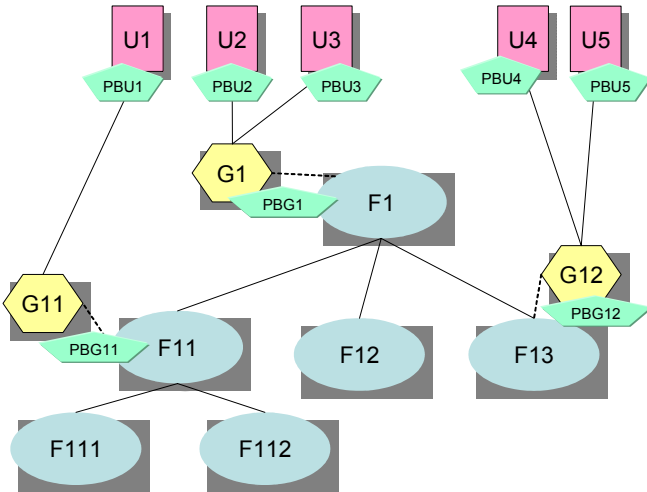


Figure 2. Shared Workspaces with User and Folder Objects

A particular feature of BSCW is the possibility to extend the access to a subfolder of a shared workspace by inviting additional members into the group. In our example G1 consisting of users U2 and U3 has been extended by user U1 for the subfolder F11. I.e. F11 as well as its subfolder F111 and F112 is now shared among users U1, U2, and U3. Similar F13 is shared between users U2, U3, U4 and U5. It should be noted that the users U1, as well as U4 and U5 do not have access to the superior folders F1 and F12. They do not even know about the existence of these folders.

The pentagons indicate the association of the public keys with the groups/folders as well as the users. The public keys are stored as additional attributes with the user and group/folder objects and can be accessed by all users who have access to these objects. In case of the users objects these are all users having access to the groupware system while the folder objects are only accessible to the users who belong to the group associated with the folder.

In the following we describe the procedure for the management and distribution of the public and private keys of users and folders.

For each user who is registered with the groupware system a public/private key pair exists. The public key is then stored with the user object in the groupware platform. The private key is managed locally by the user.

When a user creates a new shared folder, he/she also creates a new public/private key pair for this folder. Similar to the user's key, the public key is stored with the folder object. In our example these are the keys PBG1, PBG11, PBG12.

The private key of a folder must then be made available to the users of this folder. To ensure a secure distribution of the private folder key, the public keys of the users are used, i.e., the private folder key is encrypted using the public key of the users which are retrieved from the user objects. Then this encrypted key can be securely distributed either by email or by uploading the key to the shared folder.

A user who uploads a new contribution, e.g., a document, to the shared workspace must encrypt this document using the public key of the respective folder.

A user who downloads a document from a folder must decrypt the document before further use. For the decryption the private key of the folder is used. For this purpose he must be supported by a so-called key-ring management application that stores the private key of the user as well as the private keys of workspaces the user is a member of locally.

In summary, the various keys are applied in the following manner:

- The public and private keys of a folder are generated when a new folder is created and shared among a group of users.
- The public keys of a user and a folder/group are always stored in the shared workspace system to make them available for the encryption of information that is shared among the group.
- The public key of a folder is used to encrypt contributions to a shared folder. This key is stored in the shared workspace systems as an attribute of the folder.
- The private key of a folder is used to decrypt documents retrieved from the folder. The private key is managed locally using a key-ring application.
- The public key of a user is used to securely submit information to the user. In our case this mechanism is used to distribute the private keys of a folder to the users
- The user's private key is used by the user to decrypt the received folder key.
- A user's private key is also used when he/she adds a digital signature to an artifact before its upload to a shared folder.
- A key-ring application is required for the local management of private keys by the users of the systems.

Although several security management functionalities can be realised centralised as part of the shared workspace system, additional support is also required locally to provide

an easy to use solution. In the following section we describe how this concept has been successfully realised and seamlessly integrated with the BSCW shared workspace system.

IV. SECURITY FEATURES OF THE BSCW SYSTEM

The BSCW system [2] is a CSCW system based on the shared workspaces metaphor that is under development by FIT and FIT's commercial spin-off OrbiTeam since more than ten years and is today used by about one million users world-wide. Considering the security requirements specified in Chapter 1, the BSCW system until recently satisfied only R4: BSCW contains a role-based access rights model that allows users to specify very detailed access constraints on artifacts contained in BSCW workspaces.

Of course, users had always been able to encrypt and digitally sign an artifact before uploading it to a BSCW server and it could therefore be argued that all security aspects discussed above are already solved. However, we believe that support for security should become an intrinsic part of the BSCW system (and of other CSCW systems) to make the usage of security features as easy as possible for the users. This is particularly important for cooperation in locally distributed, cross-organizational teams since there may be no or incompatible security technology installed at different organizations. The burden of installing interoperable security technology at all users' sites is likely to prevent the users to address security issues at all which might result in lower level of cooperation because of the lower mutual trust of the users.

We have therefore recently added essentially two extensions to the system:

Firstly, we provided features for storing and managing public keys on a BSCW server.

Secondly, we added features for encryption and digital signatures when uploading artifacts to a BSCW server.

For the implementation we used to a large extent public domain software of the PGP (Pretty Good Privacy) security system that was originally developed by Phillip R. Zimmermann in the early 1990ies [11]. In the next sections we describe shortly those features of the PGP software we used and than in more detail the integration and extensions to the BSCW system.

A. The PGP Software

PGP is probably the most widely used system for email and document encryption and authentication. Both commercial and public domain versions of PGP implementations are available (see <http://www.pgpi.org> for comprehensive information on PGP). For the BSCW extension we used subsets of a public domain version called Cryptix [3] but a commercial version could also be used quite easily.

PGP's encryption method is a hybrid method based on both symmetric and asymmetric keys. Messages are

encrypted by symmetric keys, i.e., the same key is used for encryption and decryption since symmetric encryption/decryption can be carried out much faster (about 1,000 times) than asymmetric encryption/ decryption where a message is encrypted with the public key of the intended recipient and decrypted by the recipient with his/her private key. Only the symmetric key (created by PGP "on the fly" when encrypting a message) is encrypted with the public key of recipient. (In this section we use the term *message* for digital information that shall be encrypted and digitally signed. This term is often found in PGP literature since PGP is widely used for secure exchange of email messages.) The symmetrically encrypted message and the associated asymmetrically encrypted symmetric key are then sent to the recipient. The recipient decrypts the symmetric key with his/her private key and can then decrypt the message with the symmetric key.

It should be noted that PGP users are usually not aware – and need not be aware – of this two level encryption scheme. In their mental model of how PGP works, the message as such is encrypted with a public key and decrypted with private key.

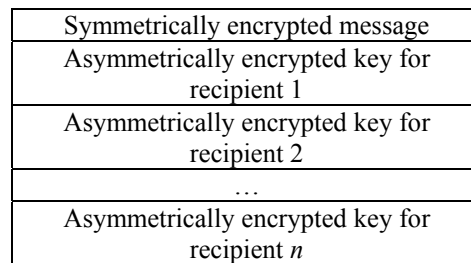


Figure 3. PGP encrypted message

PGP allows also encryption for multiple recipients by adding multiple encrypted symmetric keys, one for each intended recipient. A message encrypted by PGP therefore looks as shown in Fig. 3.

PGP provides also means for adding digital signatures to messages. For this purpose it applies a one-way *hash function* to the message, i.e., a function that takes a message of any length and produces a fixed-length (e.g., 256 Bits) *message digest* as output. Hash functions have the property that if the message is changed in any way, the same hash function would create an entirely different output. The message digest is then encrypted with the private key of the creator of the message and represents the signature which is added to the message. A recipient uses PGP to decipher the encrypted digest with the public key of the creator and then tries to recompute the message digest, thus verifying both the signature and the integrity of the message.

The PGP software provides also features for the creation of public/private key pairs. In fact, the first step of many users after installing the software is the creation of their public/private key pair.

When encrypting or decrypting data with someone's public key, one must be totally sure that the public key really

belongs to the person with whom communication is intended. For example, if person *A* pretends to be person *B* and gives another person a public key (of a public/private key pair), pretending that it belongs to person *B*, and the recipient then uses this key for encrypting a message, *A* – the wrong person – can decrypt the message.

This problem can be solved by so-called *digital certificates*. Certificates are essentially a collection of information that identify a person (e.g., name and address), the public key belonging to the person, and a digital signature of a trusted third party proving the correctness and authenticity of the certificate. The digital certificate of a person is made available to others so they can bind public keys to individual persons. In case of doubt they may access the trusted third party asking for a verification of the certificate.

PGP provides also support for two versions of digital signatures, the PGP certificate format and the certificate format according to the ITU Standard X.509 [5].

PGP stores keys as well as PGP certificate data in so-called key-rings. Data in key-rings, particularly key data, is secured by pass phrases, i.e., whenever a user wants to access a key, he/she has to enter the correct pass phrase.

It should be noted that PGP does not provide a full public key infrastructure (PKI). In particular, PGP does not solve the problem of *trust* in certificates completely although it provides a number of features for establishing so-called *Networks of Trust*. For a thorough solution of the trust problem a Certificate Authority (CA) would be required that guarantees the authenticity of certificates. The legal basis for these CAs differs from country to country and different organizations or companies may have different security policies related to the interaction with CAs.

We therefore also do not address the issue of trust in certificates within the BSCW system: Users of the security extension to BSCW have to solve these issues, in particular the issue of trust in other persons' public keys by external means.

B. BSCW server extensions: public keys as user and folder attributes

To support PGP encryption for users of the BSCW system we added PGP key data as user attributes, i.e., users can upload the public key of a public/private key pair and other information from a PGP certificate to a BSCW server. Users can also upload more than one PGP key. Once a public key has been uploaded, BSCW will create a container within the user's home folder called *Public Keys* where all public keys of the user will be stored. Other users may then view and download such a public key and use it for encrypting a document intended for the respective user.

Since PGP allows encryption for multiple recipients by adding multiple encrypted keys to a message (see previous section), this feature could be used for encrypting messages

for a group of people who collaborate by encrypting a message individually for each member of the group. However, this approach has a significant disadvantage: During collaboration processes the initial group is often extended by additional members. When messages are encrypted by using the public keys of the members who are part of a group at a particular point in time, members joining a group later could not read the messages that had been created until then.

Since in the BSCW system the creation of a group is always associated with the creation of a folder – their joint workspace – we added public keys as attributes to folders. Fig. 4 shows an example how a folder's PGP key data is displayed in the BSCW user interface.

Messages intended for the respective group having access to such a folder are only encrypted with the public key of that folder. If later on a person becomes a new member of the group, he/she has only to be provided with the private key of the folder. This could be done, e.g., by the workspace administrator who sends the private key to the new member encrypted with the member's public key.



Figure 4: Presentation of PGP key data

C. Security Support for BSCW Clients

The extensions described in the previous section are necessary requirements for supporting the indented security features at BSCW server level. However, we also wanted to provide support related to the encryption and decryption of documents as well as to the key and certificate management at the BSCW client side to make the user interface as easy as possible. Furthermore, we wanted to provide a self-contained, easy to install software environment that delivered to the users all necessary security technology .

For this purpose we enhanced an existing file uploader application called *JUploader* (*J* because it is written in Java) that had been under development for many years. The background of this component is the following: Although file upload was a HTTP feature from the very beginning of the respective Web standard, the first Web browsers did not support this feature. Therefore, for the first releases of the BSCW system we had to provide an external application that the users had to install locally on their computers which allowed the upload of files to BSCW servers.

Although in principle this application became obsolete after the Web browsers were enhanced with upload facilities – a browser is today fully sufficient at the BSCW client/user’s site – we maintained and enhanced this application continuously. For example, we added facilities for compression and multiple file upload that provided for bulk uploads often a much better performance than the file upload capabilities available from browsers.

Using the *JUploader* as the implementation basis, we added at the BSCW client site essentially three additional features:

Support for encrypting and digitally signing files before upload. This was achieved by extending the *JUploader* with the respective functionality: When the user starts the *JUploader* for uploading a file to a BSCW server, he/she first selects the file, is then prompted whether encryption shall be applied and/or a digital signature shall be added, and finally selects the respective public key that shall be used for encryption and the respective private key that shall be used for signing from his/her key-ring.

Support for decrypting files and verifying digital signatures after download files. This is provided by a Java

application called *JDecrypt*. This application starts automatically – after a respective configuration of the Web browser – whenever a user downloads a PGP encrypted file, prompts the user for the specification of a private key for decryption and/or public key for signature verification from the user’s key-ring, shows respective verification information and then stores the decrypted file on the user’s local file system.

Support for key and certificate data management. This is provided by a Java application called *JKeyManager*. This application manages the user’s key-rings, e.g., it inserts a public key of another user into the key ring or it computes a public/private key pair for the user.

All extensions related to the security technology were based on the Cryptix software package [3]. This software is also written in Java and could therefore easily be combined with the *JUploader*.

The interaction of the components at the client site with each and with a BSCW server is shown in Figure 5.

For example, when a user initiates encryption with the *JUploader*, the *JUploader* accesses the PGP key rings of the user through the *JKeyManager* component, encrypts the document with the respective components of the Cryptix software, and then uploads the encrypted file via SSL to a BSCW server.

It should be noted that the users do not have to install these different components individually. The user downloads only one installation file. During the installation process all software components (*JUploader*, *JKeyManager*, *JDecrypt* and all other components required from the Cryptix software suite) are installed automatically.

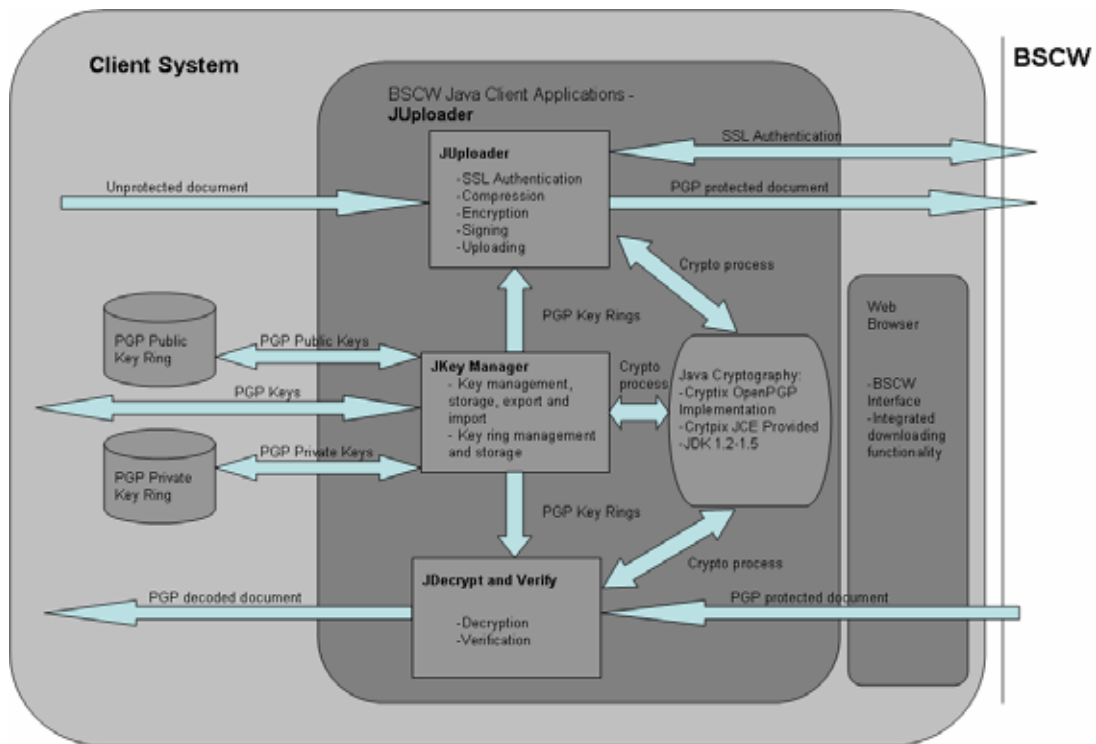


Figure 5: Interaction of all security components at client site

D. Future Extensions

The BSCW security extension described above currently support only one type of artifact appearing in BSCW workspaces: those artifacts which are considered *documents* in BSCW terminology. Note that a BSCW document is anything that is upload to or downloaded from a BSCW server, i.e., a BSCW document is not necessarily a text document but may also be a picture, digital music, video, or a piece of software.

However, users can create other types of artifacts in their workspaces such as discussion forums with notes, URL objects, calendars and entries therein, email messages, workflow objects, or polls. We plan to add encryption and digital signatures for at least some of these object types. For example, discussion forums are often used for cooperation processes in BSCW workspaces and encryption and digital signatures for notes in these discussion forums are required for appropriate security support.

Another planned extension is a security enforcement mechanism: It should be possible to specify that only encrypted artifacts can be uploaded to a BSCW workspace/folder. If a user tries to upload an unencrypted artifact (recognizable by its MIME type), the BSCW server would not store it in its data base.

V. EVALUATION RESULTS

We made first evaluations of the security extensions with a group of BSCW users. All of them can be considered advanced or even expert users with respect to BSCW but most of them had not used security technology before. The users received installation instructions (less than half a page) for the software they had to install locally and a user manual (about 35 pages) describing the features of the security extensions and their usage. Most participants used MS Windows systems, a few Linux systems.

We made the following observations:

In general, downloading and installing the software were no problems for the users. A few users had some difficulties because they had still an older Java runtime version which did not interoperate with the Java code of JUploader and the other components so they needed to upgrade their Java runtime system.

Several users had problems getting the encryption features working. For example, they did not realize that they had to upload their PGP key information to the BSCW server before they could use any of the security features.

The users were satisfied with the performance of the encoding/decoding software. The overhead of encryption before a file upload and of decryption after file download does not significantly slow down the interaction process with the BSCW server. (For example, for a document of about 10 megabytes encryption/decryption takes less than 3

seconds on a state-of-the-art PC; the response time with the BSCW server used during the evaluation was about 20 – 30 seconds.)

Some users created a rather large number of keys – although a requirement for the large number was not recognizable – and then became confused which key could be or should be used for a particular file. Similarly, other users became confused which public key of a user they should use when they observed that a user had several public keys stored under his/her user profile.

Most users did not use the full functionality provided (and required for certain security issues). For example, almost no user verified identity or certificate data. It should be noted, however, that all keys during the evaluation were created and “self-certified” by the users so a verification process would not have been really meaningful.

In general, when collecting feedback from the users on their experiences with the security extension we observed that the users were not fully aware which type of security problems were solved or could be solved with the security features provided by the BSCW system.

The user manual given to the users needs improvements. Although it fully described the functionality of the JUploader, JKeyManager and JDecrypt components – and, in general, the descriptions were well understood by the users – it should be extended by an introduction describing the security concepts on which the software is based. It should also contain recommendations on the usage of the software (e.g. “Keep the number of your public keys as low as possible, preferably use only one public key.”).

VI. ORGANIZATIONAL ISSUES OF SECURITY

The first evaluations showed that the pure provision of security technology is currently not sufficient for achieving the intended security goals and needs to be supported by appropriate organizational means.

At present, many people are not sufficiently familiar with security technology. Therefore, introducing security technology currently in an organization requires training of the users, in particular, in the concepts underlying the technology (e.g., “How does encryption/decryption with public/private key pairs work? What is a certificate? What is a digital signature? What level of security can be achieved with which means?”). Furthermore, it seems necessary to provide user guidelines or recommendations how they should use available security technology. At an organizational level it might even be necessary to develop security policies and respective enforcement procedures.

Of course, for locally distributed, cross-organizational teams training measures and security policies represent a particular challenge. However, such teams have usually a

rather limited number of members who should be able to agree on a common security policy rather easily. We believe that for a sensible usage of the security extensions for the BSCW system described above it should be sufficient if one member, e.g., the coordinator, has some basic knowledge of security concepts and issues some security guidelines to the group (e.g., “Encrypt and sign all document. Always verify signatures. Verify a person’s identity when first using his/her public key”).

Furthermore, we believe that the currently low familiarity with security technology will change in the near future since companies and organizations increasingly develop security policies, install the respective security infrastructure (keys and certificates, software for encryption/decryption and digital signatures), train their staff in the usage of the security infrastructure, and enforce the application of security policies.

VII. CONCLUSIONS

We have identified a number of security requirements that we believe should be satisfied by Web-based CSCW systems. Encrypting digital artifacts and attaching digital signatures to them can satisfy these requirements. We have described a respective extension of the BSCW system based on PGP. This requires also local installation of security software at the users’ sites; we have provided a respective software suite that is quite easy to install, self-contained and guarantees interoperability within a group of BSCW users. Further security related extensions of the BSCW system are necessary and envisaged for the future.

First evaluations of this extension by BSCW users show that the mere provision of encryption/decryption and digital signature technology is usually not sufficient but requires supporting organizational support such as training in security concepts and the development of security policies of an organization or team.

REFERENCES

- [1] T. Ahmed and R. Tripathi, “Security Policies in Distributed CSCW and Workflow Systems”, *Technical Report. University of Minnesota*, Minneapolis, MN, 2002. Available at: www.cs.umn.edu/Ajanta/papers/ahmed_tripathi_ProcIEEE.pdf
- [2] W. Appelt, “WWW Based Collaboration with the BSCW System”, *Proceedings of SOFSEM'99* (Milovy, Czech Republic, November 26 - December 4) Springer Lecture Notes in Computer Science 1725, Heidelberg, 1999, pp. 66-78.
- [3] Cryptix Foundation Limited. Cryptix Project. 1995-2005. Available at <http://www.cryptix.org>.
- [4] I. Greif and S. Sarin, “Learning from Notes: Organizational Issues in Groupware Implementation”, *Proceedings of the First Conference on Computer Supported Cooperative Work. (CSCW '86)* (Austin, USA, December 03-05, 1986), ACM Press, New York, NY, 1986, pp. 175-183.
- [5] ITU Standard X.509 - The Directory: Public-key and Attribute Certificate Frameworks. International Telecommunication Union, Geneva, Switzerland, 2000.

- [6] R. Kling, "Cooperation, Coordination and Control in Computer-Supported Work", *Communications of the ACM Trans. Program. Lang. Syst.*, 34, 12 (Dec. 1991), pp. 83-88.
- [7] W.J. Orlikowski, "Data Sharing in Group Work", *Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work (CSCW '92)* (Toronto, Canada, November 01-04, 1992), ACM Press, New York, NY, 1992, pp. 362-369.
- [8] W. Prinz, W. and S. Kolvenbach, "Support for Ministerial Workflows", *Proc. of Conference on Computer Supported Cooperative Work (CSCW'96)*, (Boston, USA, November 16-20, 1996), ACM Press, New York, NY, 1996, pp. 199-208.
- [9] G. Stevens and V. Wulf, "A New Dimension in Access Control: Studying Maintenance Engineering across Organizational Boundries", *Proceedings of the ACM 2002 Conference on Computer Supported Cooperative Work (CSCW '02)* (New Orleans, USA, November 16-20, 2003), ACM Press, New York, NY, 2002, pp. 196-205.
- [10] O. Stiemerling and F. Pacull, "Secure Component-Based Groupware", *Proceedings of the CSCW2000 Workshop on Component-Based Groupware*. (Philadelphia, USA, December 2-6, 2000, pp. 43-48.). Available at <http://doc.telin.nl/dscgi/ds.py/View/Collection-2254>
- [11] Zimmermann. P., *The Official PGP User's Guide*, MIT Press, Cambridge, MA, 1995.