Combining Incomparable Public Session Keys and Certificateless Public Key Cryptography for Securing the Communication Between Grid Participants

Elvis Papalilo and Bernd Freisleben

Department of Mathematics and Computer Science, University of Marburg,
Hans-Meerwein-Str., D-35032 Marburg, Germany
{elvis,freisleb}@informatik.uni-marburg.de

Abstract. Securing the communication between participants in Grid computing environments is an important task, because the participants do not know if the exchanged information has been modified, intercepted or coming/going from/to the right target. In this paper, a hybrid approach based on a combination of incomparable public session keys and certificateless public key cryptography for dealing with different threats to the information flow is presented. The properties of the proposed approach in the presence of various threats are discussed.

1 Introduction

Security is a key problem that needs to be addressed in Grid computing environments. Grid security can be broken down into five main areas: authentic-cation, authorization/access control, confidentiality, integrity and management of security/control mechanisms [1].

Grids are designed to provide access and control over enormous remote computational resources, storage devices and scientific instruments. The information exchanged, saved or processed can be quite valuable and thus, a Grid is an attractive target for attacks to extract this information. Each Grid site is independently administered and has its own local security solutions, which are mainly based on the application of X.509 certificates for distributing digital identities to human Grid participants and a Public Key Infrastructure (PKI) for securing the communication between them. The primarily used techniques for assuring message level security are:

- public/private key cryptography – participants use the public keys of their counterparts (as defined in their certificates) for encrypting messages. In general, only the participant in possession of the corresponding private key is able to decrypt the received messages.
- shared key cryptography – participants agree on a common key for encrypting the communication between them. The key agreement protocol is based on using the target partner's certified public key.

R. Meersman and Z. Tari et al. (Eds.): OTM 2007, Part II, LNCS 4804, pp. 1264–1279, 2007.

These solutions are built on top of different operating systems. When all participants are brought together to collaborate in this heterogeneous environment, many security problems arise.

In general, Grid systems are vulnerable to all typical network and computer security threats and attacks [2], [3], [4], [5], [6]. Furthermore, the use of web service technology in the Grid [7] will bring a new wave of threats, in particular those inherited from XML Web Services. Thus, the application of the security solutions mentioned above offers no guarantees that the information exchanged between Grid participants is not going to be compromised or abused by a malicious third party that listens to the communication.

Furthermore, they all escape the idea *why a participant in the Grid environment was chosen among the others for completing a specified task* and for *how long a collaboration partner is going to be considered.* Thus, the behaviour of the participants also needs to be considered in order to limit the possibility of malicious participants to actively take part in a collaboration.

An alternative solution to the problem is the establishment of a secured communication channel between collaborating participants (using a virtual private network - VPN). Thus, the transport mechanism itself is secured. Although in this case an inherently secure communication channel is opened between parties, the method itself is impractical to be used in Grid environments [8] due to:

- administration overhead – new tunnels need to be configured each time a new virtual organization joins or leaves the environment.
- incompatibility between different formats used for private IP spaces in small and large networks – 16-bit private IP space is preferred for small networks, while in large networks the 24-bit private IP space is preferred. There exists the possibility that (multiple) private networks use the same private IP subnet.

In this paper, we propose a hybrid message level encryption scheme for securing the communication between Grid participants. It is based on a combination of two asymmetric cryptographic techniques, a variant of Public Key Infrastructure (PKI) and Certificateless Public Key Cryptography (CL-PKC). Additionally, we first sort the collaboration partners according to their (past) behavior by considering the notion of trust in Grid environments, and in a second step, we assign to them the corresponding keys for encrypting the communication. Such a key is valid until no more tasks are left to be sent to this target partner, and as long as this partner is a trusted partner (according to the expressed trust requirements).

We mainly concentrate on the confidentiality of the communication between Grid participants, but issues related to authorization, integrity, management and non-repudiation will also be treated.

The paper is organised as follows. In section 2, related work is discussed. In section 3, an analysis of the threats to the communication between participants in Grid environments is presented. In section 4, our approach for securing the communication between Grid participants is proposed. Section 5 concludes the paper and outlines areas of future research.

2 Related Work

There are several approaches for establishing secure communication between Grid participants. For example, the Globus Toolkit [9] uses the Grid Security Infrastructure (GSI) for enabling secure communication (and authentication) over an open network. GSI is based on public key encryption, X.509 certificates and the Secure Sockets Layer (SSL) communication protocol. Some extensions have been added for single sign-on and delegation. A Grid participant is identified by a certificate, which contains information for authenticating the participant. A third party, the Certificate Authority (CA), is used to certify the connection between the public key and the person in the certificate. To trust the certificate and its contents, the CA itself has to be trusted. Furthermore, the participants themselves can generate certificates for temporary sessions (proxy certificates). By default, GSI does not establish confidential (encrypted) communication between parties. It is up to the GSI administrator to ensure that the access control entries do not violate any site security policies.

Other approaches try to improve the security of the communication between Grid participants by making use of different encryption methods. Lim and Robshaw [10] propose an approach where Grid participants use identity-based cryptography [11] for encrypting the information they exchange. However, in traditional identity-based encryption systems, the party in charge of the private keys (private key generator - PKG) knows all the private keys of its participants, which principally is a single point of attack for malicious participants. Furthermore, the approach requires that a secure channel exists between a participant and its PGK, which in turn is not very practical in Grid environments. In a later publication [12], the authors try to solve these problems by getting rid of a separate PKG and by enabling the participants to play the role of the PKG for themselves. Additionally, a third party is introduced with the purpose of giving assurances on the authenticity of the collaborating parties. Collaborating participants, based on publicly available information and using their PKG capabilities, generate session keys "on the fly", which are used between collaborating participants to exchange the initial information (job request, credentials from the third trusted party, etc.). During a collaboration, a symmetric key, on which parties have previously agreed, is used for encrypting/decrypting the information flow. This could also be a single point of attack (the attack is directed only towards a single participant) for a malicious participant willing to obtain it.

Saxena and Soh [13] propose some applications of pairing-based cryptography, using methods for trust delegation and key agreement in large distributed groups. All Grid participants that collaborate at a certain moment form a group. A subset of group members generates the public key, and the rest of the group generates the private key. A distributed trusted third party with a universal key escrow capability must always be present for the computation of the keys. These keys (public/private) are going to be used within the group for encrypting/decrypting the communication between group members.

A similar approach is followed by Shen et al. [14] where some strategies for implementing group key management in Grid environments are proposed. The main difference to the work by Saxena and Soh [13] is the re-calculation of the group key every time a participant re-joins the group.

The vulnerability of both approaches lies in the fact that all group members are aware of the public/private key. A malicious participant, already part of the group, could decrypt all messages that group members exchange between them. Even if a malicious participant is not part of the group, a single point of attack (gaining access or stealing key information from only a single group participant) could be sufficient to decrypt all the information the group participants exchange between them.

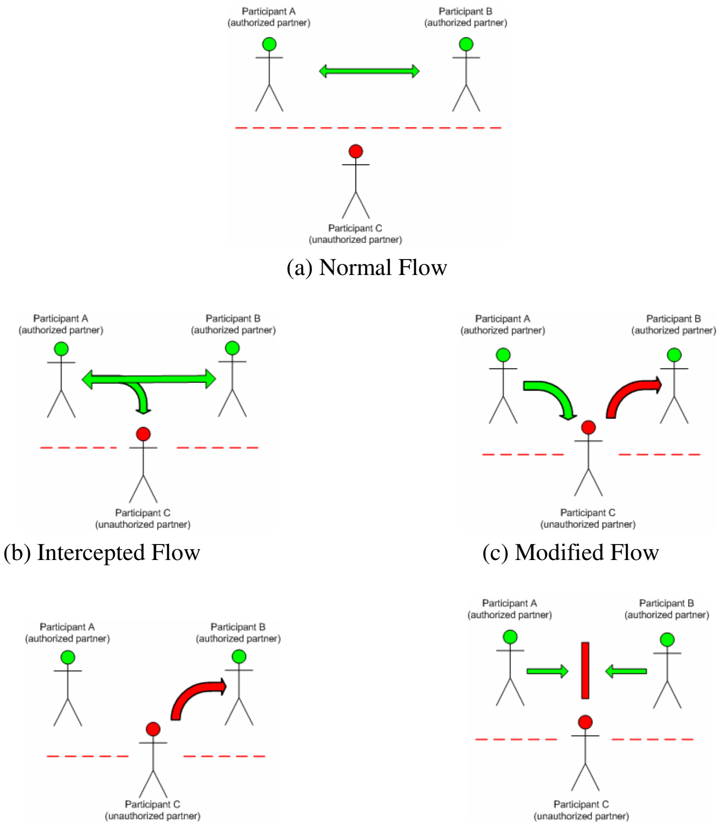
Crampton et al. [15] present a password-enabled and certificate-free Grid security infrastructure. Initially, a user authenticates itself to an authentication server through a username and password. After a successful verification, the user obtains through a secure channel the (proxy) credentials (public and private keys) that will be used during the next collaboration with a resource. The resource in turn verifies if the user is authorized to take advantage of its services and creates its proxy credentials and a job service in order to complete the tasks assigned by the user. A single trusted authority accredits the authentication parameters for the users, resources and authentication servers.

There are several problems with this approach. First, the complexity of the environment is artificially increased. While the authentication of the resources is done directly by the trusted authority, the authentication of the users is done by a third party, the authentication server. Adding more components to the authentication chain increases the points of attack. Second, the resource has to believe that the user is authenticated through a "trusted" authentication server and not by a malicious one. Third, the resource has to believe that the user is not impersonating someone else in the environment. Finally, a single participant (the trusted authority) is in charge of the authentication parameters of all other participants in the environment. It must be trusted by the participants, and at the same time it has access to private information of the participants. Thus, the participants' private information is not protected either in the scenario where this "trusted" third party turns out to be malicious or in the scenario where another malicious participant gains access to the private information of different participants through attacking this "trusted" third party (as a single point of attack).

Additionally, some web services security standards (applied also to Grid services) are also emerging. XML Signature [16] signs messages with X509 certificates. This standard assures the integrity of messages, but it does not offer any support for threat prevention. WS-SecureConversation [17] is a relatively new protocol for establishing and using secure contexts with SOAP messages. Partners establish at the beginning a secure context between them, but all the following messages are signed using the XML-Signature standard. XML Encryption [18] is also a standard for keeping all or part of a SOAP message secret. A participant in the communication is able to encrypt different sections of an XML document with different keys making possible for its collaboration partners to access only certain parts of the document, according to the assigned keys. However, in the case when many partners want access to the same part of the document or to the entire document at the same time, they come in the possession of the same key.

3 Communication Threats

A collaboration in Grid environments takes place between interacting participants. A participant is either a service provider (i.e. a node to host and provide a service, or a service instance running on the provider node) or a service consumer (i.e. a node that requests a service from a provider (including the request to deploy and perform a service at the provider), or a service instance running on the consumer node). In general, there exists a flow of information from a source participant to a target participant, as shown in Fig. 1.a.


Participant A (authorized partner)
Participant B (authorized partner)
Participant C (unauthorized partner)
Participant A (authorized partner)
Participant B (authorized partner)
Participant C (unauthorized partner)
Participant A (authorized partner)
Participant B (authorized partner)
Participant C (unauthorized partner)
Participant A (authorized partner)
Participant B (authorized partner)
Participant C (unauthorized partner)
Participant A (authorized partner)
Participant B (authorized partner)
Participant C (unauthorized partner)

(a) Normal Flow

(b) Intercepted Flow

(c) Modified Flow

(d) Forged Flow

(e) Interrupted Flow

Fig. 1. Communication Threat Scenarios between Grid Participants

This information flow can be the target of different threats. The same threats, as depicted by Stallings [19], can also be encountered in Grid environments: passive threats and active threats.

The aim of passive threats is to simply intercept the communication and obtain the information being transmitted, as shown in Fig 1.b. They affect the *confidentiality* of the exchanged information, and are difficult to detect due to the lack of direct intervention possibilities on the information the parties are exchanging.

The situation changes completely when active threats are considered. Here, intervention on the information flow is always possible. The information flow can be:

- modified: the *integrity* of the exchanged information is placed at risk as a result of the modification of the data being exchanged, through the intervention of an unauthorized third party (Fig 1.c).
- forged: the *authenticity* of the exchanged information is placed at risk as a result of the forged stream an unauthorized participant tries to exchange with the target participant, impersonating another authorized participant in the environment (Fig 1.d). This is also a *non-repudiation* problem.
- interrupted: the normal communication between partners is interrupted as a result of any intervention from an unauthorized participant in the environment (Fig 1.e). This is a threat to *availability*.

Prevention is the key to fighting passive threats. For active threats, fast detection and recovery are crucial.

In this paper, we will concentrate on issues related to confidentiality and integrity of the messages exchanged between participants. Furthermore, authorization and management issues will be sketched.

4 Approaches to Securing the Communication Between Grid Participants

4.1 Basic Key Management Model and Encryption Scheme

Grid systems typically make use of public key cryptography for securing a communication session between collaborating participants [1]. Two parties use a randomly generated *shared key* for encrypting/decrypting the communication between them. To ensure that the data is read only by the two parties (sender and receiver), the key has to be distributed securely between them. Throughout each session, the key is transmitted along with each message and is encrypted with the recipient's public key.

A second possibility is to use *asymmetric session keys*. Each of the parties randomly generates a pair of session keys (a public and a private one). Their application is similar to symmetric session keys with the difference that in this case different keys are used for encrypting and decrypting messages.

In this paper, we allow each Grid participant to generate its own keys such that each participant simultaneously possesses multiple public keys while all these keys correspond to a single private key. This method was first proposed by Waters et al. [20] and was later further developed by Zeng and Fujita [21].

According to their scheme, each time two participants A and B communicate with each other, the sender (participant A) decides to use either a public key from its pool of existing public keys or to generate a new one. This key is going to be sent to the receiver (participant *B*). Whenever *B* sends a message to *A*, the message is encrypted using *A*'s previously sent public key. Upon receipt, *A* decrypts it using its private key. The entire process is described in Fig. 2:

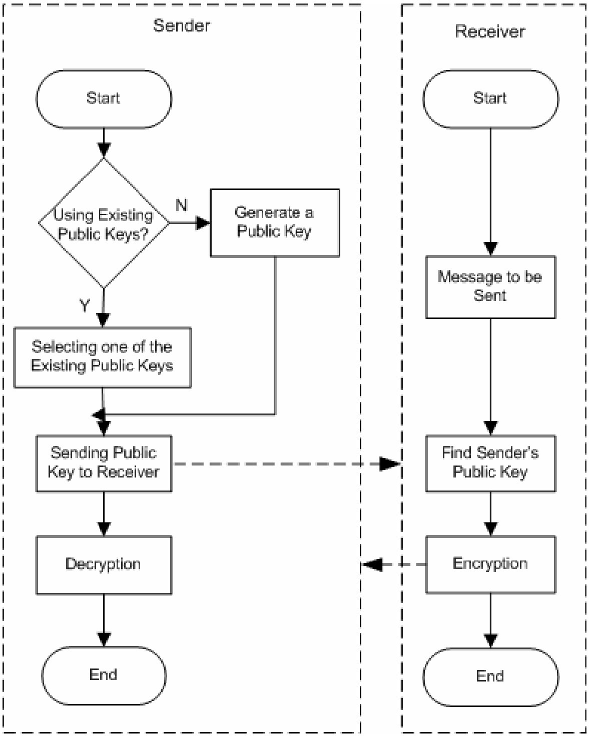


Fig. 2. Encrypting/Decrypting Scheme Used in [20]

The generation of the public keys is done according to the following algorithm:

```
1. Select a cyclic group $G$ of order $n$;
2. Select a subgroup of $G$ of order $m$, where $m <= n$;
3. Select and fix the private key $x$, where $1 < |x| < m$;
4. Select a generator $g$ of $G$;
5. Select indicator $r$, where $0 < |r| < m$;
6. Compute $y_1 = g^r$ and $y_2 = y_1^x$;
7. Release public key $(y_1, y_2)$.
```

Fig. 3. Generating Multiple Public Keys

In Fig. 3, the terms *group* and *subgroup* used by Zeng and Fujita [21] were originally defined by Menezes et al. [22].

To apply the above key management model to Grid environments, we propose the following:

- First, a collaboration in Grid environments has to take place between trusted participants. In [23], we presented a model that manages trust among Grid participants. In terms of a trust-based communication model, the collaboration takes place between the *trustors* (subjects that trust a target participant) and *trustees* (participants that are trusted). Two Grid participants involved in an interaction play both the role of a trustor and a trustee to each other.

  According to our model:

  - a participant interacts with the target participant(s) and learns their behavior over a number of interactions. In this case, the participant reasons about the outcome of the direct interactions with others. When starting an interaction with a new participant, i.e. no information about previous behavior exists, it can use its beliefs about different characteristics of these interaction partners and reason about these beliefs in order to decide how much trust should be put in each of them.
  - the participant could ask others in the environment about their experiences with the target participant(s). If sufficient information is obtained and if this information can be trusted, the participant can reliably choose its interaction partners.
- Second, *the number of public keys has to equal the number of the trusted partners (trustees) each Grid participant (trustor) selects*. In general, a normal collaboration between a trustor and its trustees, according to [23], takes place as described in the following scenario. The trustor specifies the trust requirements regarding its future partners. Then, the participants which comply with the current application requirements (Grid-enabled application) are selected. The decision which one of the chosen participants should be considered further as a collaboration partner is made by comparing the trustor's trust requirements with the obtained trust information about these specific participants (personal experience, third parties' experience). Once the trustor has taken a decision regarding the "trustworthiness" of its counterparts, it generates a single private key and exactly as many public keys corresponding to this single private key as the number of its trusted partners. These keys will be used for securing the communication between the trustor and its trustees during the collaboration that is going to take place.
- Third, directly after the generation of these public keys, *the trustor has to assign a key to each of its trustees*. Thus, every trustee uses a separate public key for encrypting the messages/information it exchanges with the trustor. The trustor itself uses a single private key for decrypting the communication flow.
- Fourth, *the generated keys should be valid only during the lifetime of the upcoming collaboration*. Since the trust values that participants establish to each other change according to the personal performance (and intentions), a trusted participant in the current collaboration is not necessarily a trusted one in future collaborations.

The entire approach is summarized in the algorithm shown in Fig. 4.

1. According to its needs and to the trust information gathered from different sources, the trustor establishes all the target participants (trustees) that are going to be considered in the very next collaboration (the number of trusted partners is referred with n).
2. A private key (P_B) is determined and the algorithm presented in Fig. 3 is repeated n times ($K_B(n)$- n public keys are generated).
3. The generated public keys are sent to the trustees; every trustee receives only one key ($K_B(i)$).
4. Each trustee, once it wants to send a message/information to the trustor, encrypts the information flow using the respective $K_B(i)$.
5. As soon as the trustor receives the encrypted message/information, it uses P_B to decrypt it.

Fig. 4. Multiple Public Keys Management Scheme

The advantages of the proposed approach are:

- public keys are created by the trustor itself and are distributed directly and only to trusted participants. Not every participant in the environment is aware of them. Thus, the proposed approach mitigates also the non-repudiation problem,
- the lifetime of the private key (P_B) and the incomparable public keys ($K_B(i)$) does not span over the lifetime of the collaboration itself.

However, since the public keys are going to be distributed through a “public” and “non-secure” communication channel, the key distribution scheme is vulnerable to a “man-in-the-middle” attack. Thus, a third “unauthorized” participant could either obtain the key(s) by intercepting the information flow as shown in Fig 1.b or by impersonating some other trusted participant in the environment [24].

For this reason, we extend our approach by applying a double encryption scheme. A second pair of keys, generated via a certificateless key generation scheme, and information tightly related to the participant itself, is used, as described in the following.

4.2 A Double Encryption Scheme

4.2.1 Certificateless Public Key Cryptography in Grid Computing

Certificateless public key cryptography (CL-PKC) was first proposed by Al-Riyami and Paterson in [25]. It combines elements of identity-based public key cryptography and traditional public key cryptography.

The generation of the keys is done in two stages. In the first stage, a participant in the environment receives from a key generation center (KGC), over a confidential and authentic channel, a partial private key. This partial key is computed using an *identifier* of the participant.

In the second stage, the participant produces its private key by combining the partial private key with some secret known only to the participant. Thus, no one else, other than the participant itself, knows the generated private key. A public key, which matches the private key, is then published.

A distinct feature of the model is that it completely eliminates the need to obtain a certificate from the trusted authority in order to establish the authenticity of a public key.

According to [26], the Grid is aimed at enabling virtual communities to share geographically distributed resources as they pursue common goals, assuming the absence of central location, central control, omniscience, and existing trust relationships. Thus, having a central KGC is quite impossible. In order to overcome this problem, we propose to use a hierarchical model for KGCs. The idea is presented in Fig. 5.

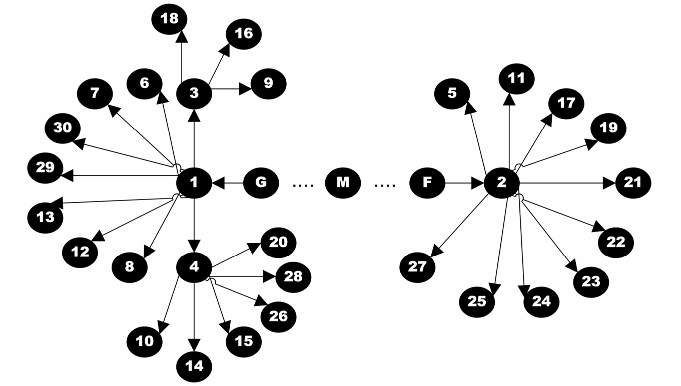


Fig. 5. Establishing a Hierarchical Model for KGCs

Every Grid participant, other than being a possible partner for the other Grid participants in the environment, could also be a KGC for another participant or even for more than one of them (i.e. in Fig. 5, participants G and F are KGC for participants 1 and 2 respectively, participant 1 is KGC for participants 3, 4, 6, 7, 8, 12, 13, 29 and 30; participant 3 is KGC for participants 9, 16 and 18; participant 4 is KGC for participants 10, 14, 15, 20, 26 and 28; participant 2 is KGC for participants 5, 11, 17, 19, 21, 22, 23, 24, 25, and 27). Considering the graph in Fig. 5, dedicated KGCs, like G, M and F (in charge only of partial key distribution; e.g. international or national centers, universities, etc.), have a relationship of first order, i.e. a supposed direct relationship between them exists and they have the same importance. Within such a scheme, all participants are connected through chains to each other. Participants that do not have such a connection (i.e. do not possess a KGC or serve as a KGC for themselves), have an *infinite relationship* with other participants (not present in Fig. 5).

The following information could be used from KGCs for computing the partial private keys:

- ***What a participant is*** – refers to personal *attributes* of every single participant. Examples of these traits include hardware and software peculiarities of the participant (i.e. operating system, hardware in use, network physical address, IP address, etc). Part of these attributes or a combination of them is difficult to duplicate and very specific to a single participant.
- ***What a participant does*** – refers to unique patterns of *behavior* that this participant manifests during the collaboration with others in the environment. Trust values can be gathered from different (ex) partners [23] of the target participant, whose partial private key a KGC is currently computing, and be used during the computation process.

Having received this partial private key, the Grid participant could generate the full private key.

4.2.2 A Protocol for Encrypting/Decrypting the Information Flow Between Grid Participants

The proposed protocol is a combination of the approaches described above and works in the following manner (assuming that each KGC has a master key $\boldsymbol{M_{KGC}}$ and a public key $\boldsymbol{K_{KGC}}$):

- Every participant $\boldsymbol{i}$ contacts its KGC (***KGC(i)***) for receiving the partial private key;
- The ***KGC(i)*** computes the partial private key $\boldsymbol{P_{PK}(i)}$ using its master key $\boldsymbol{M_{KGC}(i)}$, its public key $\boldsymbol{K_{KGC}(i)}$ and an identifier ***ID(i)*** (personal attributes or specific patterns of behavior) of the participant;
- The participant, in an intermediary step, computes a secret value ***S(i)*** making use of $\boldsymbol{K_{KGC}(i)}$ and ***ID(i)***. This secret value ***S(i)*** is then combined with the partial private key obtained $\boldsymbol{P_{PK}(i)}$ and the KGC's public key $\boldsymbol{K_{KGC}(i)}$ for generating the actual private key $\boldsymbol{P_{CL}(i)}$; Similarly, the public key $\boldsymbol{K_{CL}(i)}$ is generated from the combination of the user's secret value ***S(i)*** with the public key $\boldsymbol{K_{KGC}(i)}$ of its KGC. This public key ($\boldsymbol{K_{CL}(i)}$) is made available to the others through placing it in a public directory;
- The participant, according to the application requirements and to the trust information gathered from considered trust sources, establishes all the partners (trustees) that are going to be considered during the very next collaboration (the number of trusted partners is referred to with $\boldsymbol{n}$); two partners that decide to collaborate with each other are both trustor and trustee to each other; a participant in the environment with an *infinite relationship* to the trustor is not considered at all as a trustee;
- The participant $\boldsymbol{i}$ (in this case the trustor), determines a private session key $\boldsymbol{P_B(i)}$ and $\boldsymbol{n}$ different public session keys $\boldsymbol{K_B(n)}$ (a different public key for each of the $\boldsymbol{n}$ established trustees);

- Before sending each public session key $\boldsymbol{K_B(j)}$ to the target trustee $\boldsymbol{j}$, the trustor encrypts it with the corresponding $\boldsymbol{K_{CL}(j)}$;
- The trustee $\boldsymbol{j}$, once receiving the encrypted message, decrypts it using its $\boldsymbol{P_{CL}(j)}$, obtaining the $\boldsymbol{K_B(j)}$ that it is going to be used to encrypt the information flow with its partner.
- Once a collaboration has to take place, the trustor first encrypts the information using the public session key $\boldsymbol{K_B(i)}$ assigned by its partner and then re-encrypts the already encrypted information using the $\boldsymbol{K_{CL}(j)}$ key made public by its partner;
- The double encrypted information is initially decrypted by the trustee using its $\boldsymbol{P_{CL}(j)}$ key. The obtained information is further decrypted using the personal private session key $\boldsymbol{P_B(j)}$.

4.3 Discussion

In this section, we discuss how our approach deals with the different threat scenarios presented in section 3 of the paper:

Intercepted Flow. Here, the following scenarios can be distinguished:

- An unauthorized participant does not have any clue about the existence of the encryption of the information flow or does not possess any of the decryption keys. This is an ideal scenario, because the encryption itself brings the advantage that the unauthorized participant cannot gather any information. A brute force attack will result in significant costs and time to break the encryption.
- An unauthorized participant is aware of the encrypted flow and is able to forge or obtain the $\boldsymbol{P_B}$ and $\boldsymbol{P_{CL}}$ keys. Forging both keys of a Grid participant is an extremely difficult task, because $\boldsymbol{P_B}$ is valid only during the ongoing session, and $\boldsymbol{P_{CL}}$ is generated using specific information of this participant and is in possession of only the participant itself. Even the KGC has no complete knowledge of $\boldsymbol{P_{CL}}$. The only possibility for an unauthorized participant is to take control of the authorized participant for obtaining the original keys. However, in order to have a fully decrypted information flow, the unauthorized participant needs to obtain all the private keys of all the authorized participants involved in the current collaboration.

Modified Flow. Following the same reasoning as above, modifying multiple encrypted information flows is a very difficult task for an unauthorized participant. Enormous efforts, monetary means and time are needed in order to succeed.

Forged Flow. In our approach, two participants establish a collaboration between them only if they are considered as trusted partners for each other. The only possibility for an unauthorized participant to forge an information flow is to impersonate another participant in the environment. However,

- impersonating a participant *C* in the environment does not mean that it is a trusted partner for participant *A*, although *C* might have been considered as trusted for participant *B* (non-transitivity of trust). An additional attack to the trust information of participant *A* is needed. Even though, since trust changes with time (increases, decreases), a trusted partner for participant *A* during a current collaboration is not necessarily a trusted one in future collaborations.

- impersonation is not enough. An unauthorized participant also needs the information owned by the authorized participant it is impersonating (i.e. the public key(s) delivered from its trusted partners).

Interrupted Flow. This attack prevents or inhibits the normal collaboration between trusted participants; our approach does not offer any direct possibility to prevent such attacks. However, let us consider the scenario presented in Fig. 6.

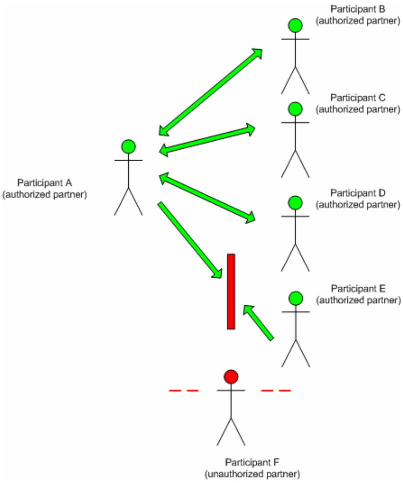


Fig. 6. Interrupted Flow between Trusted Grid Partners (Scenario No. 1)

In Grid environments, the trustor generally collaborates with more than one trustee. As already presented in [23], the entire process is monitored and trust information is collected with respect to every single trustee the trustor collaborated with. The components to be monitored could be derived from the parameters of QoS like: reliability (correct functioning of a service over a period of time), availability (readiness for use), accessibility (capability of responding to a request), cost (charges for services offered), security (security level offered), performance (high throughput and lower latency), etc. In terms of Fig. 6, the attacked QoS element is the availability of one of the trustees. The indirect solution offered by our approach is that after some unsuccessful efforts to contact the attacked partner, the flow is directed towards the other available and trusted partners and the rest of the collaboration is going to take place only with them.

However, for the attack scenario presented in Fig. 7, our approach does not offer any prevention possibilities.

In this case, (distributed) denial-of-service prevention mechanisms need to be considered.

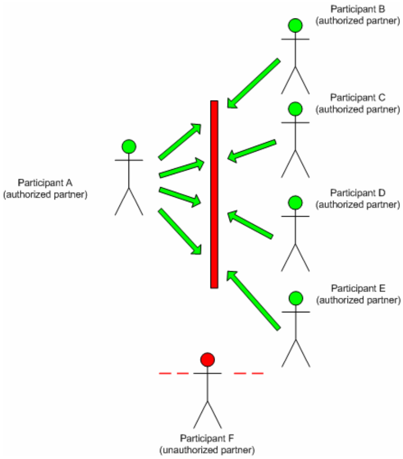


Fig. 7. Interrupted Flow between Trusted Grid Partners (Scenario No. 2)

5 Conclusions

Securing the communication between Grid participants is an important task, and there are many threats to the information Grid participants exchange between them. The approach presented in this paper for securing this information was based on the following ideas. First, a collaboration has to take place only between trusted participants. To establish and manage trust among Grid participants, our previous work [23] can be used. Second, our approach makes use of a double encryption scheme in which the transmitted information is initially encrypted using incomparable public session keys (a technique where a participant generates itself several public keys corresponding to a single private key; the number of public keys equal the number of trusted partners a participant identifies). In a second stage, this already encrypted information is encrypted again using keys generated through a technique based on certificateless public key cryptography. Finally, we have discussed how the proposed approach deals with different analyzed threat scenarios.

Future work will concentrate on implementing the proposed encryption mechanism in real Grid environments. The principal interest is to receive an experimental confirmation of its properties in the face of different threats. Furthermore, we will proceed with the evaluation of the consequences our approach has for Grid participants in terms of collaboration costs and speed of processing.

References

1. Lock, R., Sommerville, I.: Grid Security and its Use of X.509 Certificates, http://www.comp.lancs.ac.uk/computing/research/cseg/projects/dirc/papers/gridpaper.pdf
2. Negm, W.: Bringing Balance to Web Services (2004), http://www.forumsystems.com/papers/04_Bringing_Balances_Security.pdf
3. Negm, W.: Anatomy of a Web Services Attack (2004), Available: http://www.forumsystems.com/papers/ Anatomy_of_Attack_wp.pdf

4. Lindstrom, P.: Attacking and Defending Web Services (2004), Available: http://www.forumsystems.com/papers/Attacking_and_Defending_WS.pdf
5. Bloomberg, J., Schmelzer, R.: A Guide to Securing XML and Web Services (2004), Available: http://www.reactivity.connectthe.com/xml
6. De Roure, D., Jennings, N., Shadbolt, N.: Research Agenda for the Semantic Grid: A Future E-Science Infrastructure (2001), Available: http://www.semanticgrid.org/v1.9/semgrid.pdf
7. Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist, F., Subramaniam, R., Treadwell, J., Von Reich, J.: The Open Grid Services Architecture, http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf
8. Tsugawa, M., Fortes, J.A.B.: A Virtual Network (ViNe) Architecture for Grid Computing. In: IPDPS. Proceedings of 20th International Parallel and Distributed Processing Symposium, Rhodes Island, Greece, vol. 10 (2006)
9. http://www.globus.org
10. Lim, H.W., Robshaw, M.J.B.: On Identity-Based Cryptography and Grid Computing. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 474–477. Springer, Heidelberg (2004)
11. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
12. Lim, H.W., Robshaw, M.J.B.: A Dynamic Key Infrastructure for Grid. In: Sloot, P.M.A., Hoekstra, A.G., Priol, T., Reinefeld, A., Bubak, M. (eds.) EGC 2005. LNCS, vol. 3470, pp. 255–264. Springer, Heidelberg (2005)
13. Saxena, A., Soh, B.: Pairing-Based Cryptography for Distributed and Grid Computing. In: ICC. Proceeding of the IEEE International Conference on Communications, Istanbul, Turkey, pp. 2335–2339 (2006)
14. Shen, Zh.D., Wu, X.P., Wang, Y.H., Peng, W.L., Zhang, H.G.: Group Key Management in Grid Environment. In: IMSCCS. Proceedings of the 1st International Multi-Symposium on Computer and Computational Sciences, Hangzhou, Zhejiang, China, pp. 626–631 (2006)
15. Crampton, J., Lim, H.W., Paterson, K.G., Price, G.: A Certificate-Free Grid Security Infrastructure Supporting Password-Based User Authentication. In: Proceedings of the 6th Annual PKI R&D Workshop, Gaithersburg, Maryland, USA (2007)
16. XML-Signature Syntax and Processing. W3C (February 2002), http://www.w3.org/TR/xmldsig-core/
17. Web Services Secure Conversation Language (WS-SecureConversation) (February 2005), http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf
18. XML Encryption Syntax and Processing (December 2002), http://www.w3.org/TR/xmlenc-core/
19. Stallings, W.: Cryptography and Network Security, 4th edn. Prentice-Hall, Englewood Cliffs (2006)
20. Waters, B.R., Felten, E.W., Sahai, A.: Receiver Anonymity via Incomparable Public Keys. In: CCS, Washington, D.C., USA, pp. 112–121 (2003)
21. Zeng, K., Fujita, T.: Methods, Devices and Systems for Generating Anonymous Public Keys in a Secure Communication System. Patent No. 20060098819 (2006), http://www.freepatentsonline.com/20060098819.html
22. Menezes, A.J., van Oorschot, P.C., Yanstone, S.A.: Handbook of Applied Cryptography, 5th edn. CRC Press, Boca Raton (2001), http://www.cacr.math.uwaterloo.ca/hac/
23. Papalilo, E., Friese, T., Smith, M., Freisleben, B.: Trust Shaping: Adapting Trust Establishment and Management to Application Requirements in a Service-Oriented Grid Environment. In: Zhuge, H., Fox, G.C. (eds.) GCC 2005. LNCS, vol. 3795, pp. 47–58. Springer, Heidelberg (2005)

24. Lenstra, A.K., Yacobi, Y.: User Impersonation in Key Certification Schemes. Journal of Cryptology 6(4), 225–232 (1993)
25. Al-Riyami, S.S., Paterson, K.G.: Certificateless Public Key Cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
26. Foster, I., Kesselman, C., Nick, J.M., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In: Open Grid Service Infrastructure WG, Global Grid Forum (2002)