Practical Authentication Protocols for Protecting and Sharing Sensitive Information on Mobile Devices

Imed El Fray[1], Tomasz Hyla[1], Mirosław Kurkowski[2], Witold Maćków[1], and Jerzy Pejaś[1]

[1] West Pomeranian University of Technology, Faculty of Computer Science and Information Technology, Szczecin, Poland
{ielfray,thyla,wmackow,jpejas}@zut.edu.pl
[2] University of Luxembourg, Computer Science and Communication Group, 6, rue Richard Coudenhove-Kalergi, 1359 Luxembourg, Luxembourg
miroslaw.kurkowski@uni.lu

Abstract. Mobility of users and information is an important feature of IT systems that must be considered during design of sensitive information protection mechanisms. This paper describes an architecture of MobInfoSec system for sharing documents with sensitive information using fine-grained access rules described by general access structures. However, the proper usage of general access structures requires trusted components and strong authentication protocols. They allow to establish secure communication channels between different system components. In the paper we propose a conference protocol based on Boyd's ideas with key transport and key establishment mechanisms. We show that the protocol achieves three goals: (a) the key and participants' mutual authentication, (b) the common secure communication channel, and (c) the personal secure communication channels between the protocol initializer and other protocol participants.

Keywords: mobile device, sensitive information, authentication protocols, conference protocol, secure communication channel.

1 Introduction

As more and more information within organisations is created, stored and shared electronically, the issue of protecting, sharing and archiving sensitive information has become a major concern. Shared information is often stored in the network and downloaded on mobile devices when they are needed. The information should be stored in an encrypted form at a fine-grained level to reduce the risks and vulnerabilities associated with information security, i.e., anonymity, privacy, information retrieval, loss, theft and interception. Such a solution is based on cryptographic access control mechanisms and is typically implemented in two stages [1,2]. At the first stage the information is encrypted (according to

Z. Kotulski et al. (Eds.): CSS 2014, CCIS 448, pp. 153–165, 2014.

some pre-defined access control policy) and is made available on a public server. At the second stage the encrypted information can be collected by any entity. However, the information can be read only by an entity that meets the requirements specified in the access policy related to the encrypted information. Usually, the access policy requires a well-known group of participants who cooperatively try to decrypt a ciphertext. A group decryption process must be preceded by strong mutual authentication of all group members. The authentication process is initiated by an entity U_0 (called the chairman). The chairman is interested in deciphering the information downloaded from the network. If deciphering requires cooperation of $(n+1)$ entities $U_0, U_1, \ldots, U_n$ (participants of a group U), then authentication is not an easy task. In such a situation it is required to design authentication protocol that will be effective primarily in terms of running time. The natural solution to the problem is to use any two-party one-to-one authentication protocol (e.g., [3,4,5,6]). A chairman executes (sequentially or simultaneously) n times the one-to-one protocol with every other member from the group U. Successful completion of each protocol enables to authenticate every pair of users (U_0, U_i), $i = 1, \ldots, n$, and to establish n independent secure communication channels between them. This type of simple generalization of two-party protocols to the multi-party situation (especially for large group of participants) may be too expensive, in terms of both communications and computation, because each principal needs to receive and verify explicit authentication information from all other group members. Multi-party conference key agreement protocols are more advanced generalization of two-party protocols for establishing keys [1,6]. The protocols of this type are executed between entities belonging to a common group of entities (called the conference). However, the messages exchanged between the parties are authenticated only by the initiator of the protocol. On the one hand, this allows to reduce the time complexity of these protocols and enables to create a common secure communication channel, but on other hand, it does not ensure mutual authentication of an initiator with other group members.

1.1 Our Contributions

The first objective of this paper is to describe the general architecture of MobInfoSec system, which enables cryptographic protection of sensitive information in accordance with Originator Controlled (ORCON) access control rules [10,11]. The ORCON rules release a user from the obligation to monitor any information (especially against unauthorized copying). The information is removed when a user is no longer allowed to access it. The MobInfoSec needs the strong authentication between key components like secret protection modules. The definition of the MobInfoSec system, its properties and parameters allowed us to derive design goals for authentication protocol and achieve the second objective of this paper, i.e., the proposal of conference authentication protocol design for protecting and sharing sensitive information on mobile devices. The protocol summarizes the results from our previous work on authentication and key establishment protocols

for different multi-party authentication models. We review the existing solutions and classify their suitability for protection of sensitive information in mobile devices. The result of this review is the proposal of multi-party key agreement protocol based on Boyd protocol idea [4,5,6,7]. The protocol allows the establishment of a common secure channel and enables mutual authentication of each pair of protocol participants $(U_0, U_i),\ i = 1, \ldots, n$. Additionally, personal secure communication channels are also established.

1.2 Paper Organisation

The remainder of this paper is organized as follows. In the next section we shortly describe the architecture of MobInfoSec system, its properties, components and their mutual relations. The same Section presents the identified trusted domains and authentications problem in MobInfoSec system. In Section 2.3 we derive design goals for authentication and key establishment protocols in MobInfoSec system and present comparison of selected protocols. Section 3 contains description of a new Boyd's based conference protocol and short discussion concerning its security. The paper ends with conclusions, including directions for our future investigations.

2 Background

2.1 MobInfoSec System

MobInfoSec is a distributed, modular, and configurable cryptographic access control system to sensitive information [1]. The system allows building confidence to software and hardware components of popular mobile devices available at the market. One of the most complex components that needs to be implemented properly to enable access control according to ORCON rules is a strong mutual authentication scheme between secret protection (SP) modules. In the MobInfoSec system the strong mutual authentication is required before any group decryption operation. Its main purpose is to create secure (trusted) communication channels between mobile devices (i.e., between SP modules inside mobile devices). MobInfoSec architecture (see Fig. 1 - the arrows use UML notation for labels) consists of several subsystems which are divided into three categories: subsystems working on server-side of the system (at service provider site), subsystems used by mobile users and external subsystems performing services used by MobInfoSec. The system consists of six logical subsystems connected with three subsystems in an external environment.

Server-side components. Policies and Assertions Management Subsystem (PAMS) contains several components that provide key features and can be divided into three categories. The first one is related to management of targeted access policies and their templates (generation, storage and distribution). The second group of functions is related to management of users and mobile devices.

The third category contains functions related to assertions (attributes) management. The subsystem only distributes the data to Standard Trusted Services (STS) and is not available directly for mobile devices. The STS is the source of that information via the trusted components providing the information from PAMS. Dispatcher Subsystem (DS) is used to generate targeted access policies and to encrypt documents with sensitive information in accordance with those policies. Generated policies are published in the repository located in STS. An encrypted document linked with a target access policy is published in External Subsystem in an untrusted document registry. **Mobile components.** User Subsystem (US) and Mobile Device Protection Subsystem (MDPS) are two logical subsystems that are located in Mobile Device. US is responsible for authentication and authorization of users and mobile devices, for distribution of access policies to mobile devices and it enforces access policy in the case of decryption. Additionally, there is located an application that presents the data subjected to access policy. The integrity of trusted applications sets is supervised by MDPS. MDPS through SP module provides specific cryptographic keys to the trusted code. The SP module is a source of trust (at various levels, depending on SP type). SP protects directly trusted US components implementing ORCON rules. **External components.** External PKI Services Subsystem provides services related to a public key infrastructure (PKI). Furthermore, External Model PKI Services Subsystem provides PKI services which are not available in External PKI and Cryptographic Services Subsystem and are necessary for the functioning of new algorithms and protocols developed especially for the MobInfoSec system. That subsystem is not a part of MobInfoSec system and belongs to its environment. External Systems subsystem contains untrusted mobile device that can be vulnerable for attempts tampering its integrity. The mobile device is a platform for placement of dispatcher or user subsystems. Another part of external systems is an untrusted document registry. The untrusted document registry contains encrypted documents. It might be public http or ftp server or service intended to store files in a cloud.

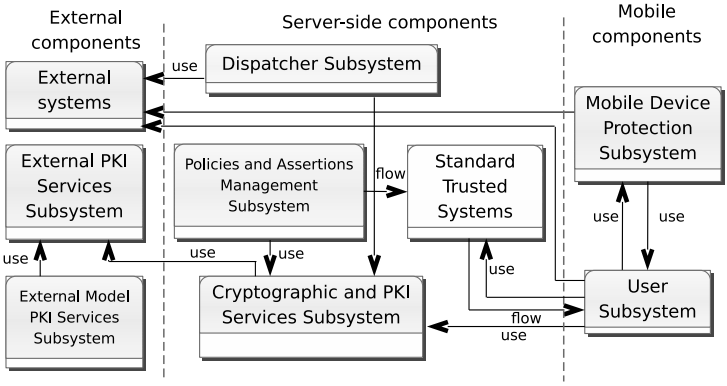

Fig. 1. MobInfoSec subsystems [1]

2.2 Trusted Domains and Authentications

MobInfoSec system can be treated as a set of distributed cooperating applications located in different network places (Fig. 2). Applications can be grouped according to the trust domain. A single domain is created around a trusted application or a group of trusted applications. Communication between applications in a single domain is secure, which may result, e.g., from the fact of deploying them in one location or the use of security technologies such as SSL. The problem that remains open is a communication between components located in different trust domains. The communication requires the creation of trusted paths and channels. The paths and channels created using strong cryptographic mechanisms allow applications from different trust domains to trust each other and mutually accept decisions.

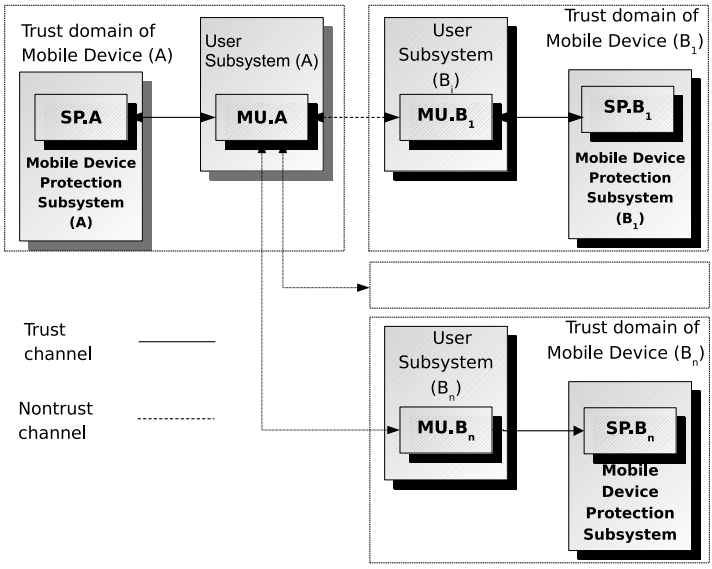

Fig. 2. Trust domains concept for different mobile devices

2.3 Basic Entity Authentication Protocols and a Key Establishment Protocols

Cryptographic authentication protocol depends primarily on the methods used to generate session keys and on the number of protocol participants (according to Boyd [7]). Generally MobInfoSec system requires secure communications between multiple entities. It is necessary to design protocols that establish keys for groups of principals to achieve such goal. In the MobInfoSec system the most important is a cryptographic authentication protocol implemented between SP components located in different trust domains. One of these domains (called a chairman) is the initiator of the protocol and should mutually authenticate with every other trust domain and establish secure communication channels.

Table 1 presents a few potentially useful protocols considered for MobInfoSec system and based on literature analysis [3,4], [6,7], [10]) compares their properties given in [4]. The most important are three groups of properties (in addition to confidentiality and integrity of keys): two or multi-party authentication, independent (personal) versus common communication channels and existence of a formal security proof. Protocols providing multi-party authentication are more effective (require fewer runs). However, they allow creating a common channel, which is not always beneficial from the specific application point of view. This is the case of MobInfoSec system, where communication channel are created for sending shadows to the initiator. The initiator is the only recipient of those messages and no other group member should have access to them. However, this problem can be solved, and it is possible to provide access to confidential information only to the originator of the protocol (at the expense of additional cryptographic operations, see Section 3).

Table 1. Authentication protocol comparison

		Protocol (variant)				
No.	Property	Transport RSA (EN 14890) [3]	Key transport ISO/IEC 11770-3 Mechanism 5 [10]	Lim-Lee key agreement Protocol 5 [6]	Boyd's conference protocol A [7]	Boyd's conference protocol B [4]
1.	Mutual authentication	+	+	+	−	−
2.	Multi-party authentication (one to many and many to one)	−	−	−	+[5)]	+[5)]
3.	Key integrity	+	+	+	+	+
4.	Key authentication	+	+	+	+	+
5.	Personal (independent) communication channels	+[1)]	+[1)]	+[1)]	−	−
6.	Common communication channel	−	−	−	+	+
7.	Forward secrecy	−	−	−	−	−
8.	Backward secrecy	N/A[2)]	N/A	N/A	−	−
9.	Liveness	+[3)]	+[3)]	+	+	+
10.	Key control	+	+[4)]	+	+	+
11.	Key freshness	+[3)]	+[3)]	+[3)]	+[4)]	+[4)]
12.	Key confirmation	+	+	+	+[5)]	+[5)]
13.	Formal security proof	+	−[6)]	−[6)]	+	+

Legend:

+ means that the protocol has indicated property, perhaps after meeting additional requirements presented in footnote
[1)] It also applies to the case when the protocol is used to authenticate the initiator of the protocol with other members of the participants group
[2)] N/A - not applicable
[3)] Applies to all participants of the protocol
[4)] Applies only to the initiator of the protocol
[5)] If session key is used by all members of the group
[6)] Lack of information about the existence of a formal security proof.

Considering the above facts and the existence of formal security proofs, for further work on authentication protocols in the MobInfoSec three protocols were adopted: RSA-based key transport protocol (according to EN 14890) and Boyd's conference protocols A and B. RSA-based key transport protocol is normally run by the two parties. Mutual authentication of protocol initiator with each of the n group members requires to initiate and perform n independent authentication protocols. Their successful completion allow to authenticate the protocols initiators with all the other participants and to establish n independent communication channels. Boyd's conference protocols in two variants (see Tab. 1) allow a chairman to establish a common communication channel with a certified key that can be established during running only one instance of the authentication protocol. Although it does not provide mutual authentication of the initiator with other participants in the protocol, this can be achieved after modifications introduced into the protocol presented in Section 3. Moreover, these modifications allow to achieve the independent communication channels, while there is still the common communication channel.

3 3 Conference Authentication Protocol Design

In this section the conference protocol `SPs_Conference_Key_Agreement` is presented, one of two designed especially for the MobInfoSec system. The main protocol purpose is mutual authentication of mobile devices, which are under control of a user A and users B_i, $i = 1, \ldots, n$, and establishment of secure communication channels. We assume that the user A is an initiator. In a typical use scenario of `SPs_Conference_Key_Agreement` a user A, (the owner of mobile device $UM.A$), needs to retrieve shadows of keys from devices $UM.B_i$, being under control of users B_i. Consequently, the restored key can be used to decrypt the document.

3.1 Notation and Assumptions

For the protocol description the following short names of components and notations are used (see also Fig. 2):

`DS[key](msg)`	A digital signature of a message `<msg>` created using a key `<key>`
`E[key](msg)`	Encryption of a message `<msg>` using a key `<key>`
`h(msg)`	A digest calculated for a message `<msg>` using hash function `h`
`MAC[key](msg)`	Message Authentication Code of `<msg>` built with `<key>`
`MU.X`	Authentication module under control of a user `X` installed in a mobile device
`PrK.SP.X.AUT`	A private authentication key installed in the `SP` belonging to an entity `X`
`PuK.SP.X.AUT`	A public authentication key installed in the `SP` belonging to an entity `X`
`Q||Z`	A concatenation of information `Q` and `Z`
`SP.X`	Secret Protection module installed in a user `X` mobile device
`X`	A protocol participant

We assume that for a given set of entities $P = \{A, B_1, \ldots, B_n\}$, where n ≥ 1, the entity A is the preferred entity responsible for initiating the protocol. The aim of the protocol is the mutual authentication with each entity $B_i, i = 1, \ldots, n$, and generation of a key material necessary to ensure the confidentiality and authenticity of information exchanged between the parties.

3.2 `SPs_Conference_Key_Agreement` Protocol Description

The `SPs_Conference_Key_Agreement`$(A, B_1, \ldots, B_n)$ protocol is based on the Boyd's conference protocol idea [4], [7]. Successful completion of the protocol authenticates directly only an entity A. Other entities authenticate themselves indirectly in the moment of usage of key generated based on an agreed key material. The key material is common for all entities that take part in the protocol. The `SPs_Conference_Key_Agreement`$(A, B_1, \ldots, B_n)$ protocol consists of five phases.

Phase I. Protocol participants exchange certificates between each other and activate necessary keys. After successful completion of that phase, `SP.A` has certificates `C.SP.B`$_i$`.AUT` and public keys `PuK.SP.B`$_i$`.AUT` of modules `SP.B`$_i(i = 1, \ldots, n)$, and each module `SP.B`$_i(i = 1, \ldots, n)$ contains the certificate `C.SP.A.AUT` and the public key `PuK.SP.A.AUT` of the module `SP.A`. Modules `SP.A` and `SP.B`$_i$ $(i = 1, \ldots, n)$ have activated their keys needed during execution of cryptographic operations. Subsequent phases are performed as follows.

Phase II. Activation of the module `SP.A` and generation of the first component of common key material:

1. MU.A requests activation by SP.A of private key PrK.SP.A.AUT:

   MU.A → SP.A: activate security key (PrK.SP.A.AUT)

2. SP.A activates key PrK.SP.A.AUT and sends confirmation to MU.A:

   SP.A → MU.A: conf.OK

3. MU.A requests from SP.A to generate a random number and store it under its control:

   MU.A → SP.A: get rand

4. SP.A generate random number RND.SP.A and stores it in the memory:

   SP.A → MU.A: conf.OK

Phase III. (Generation of key material' components) For each pair of entities $(A, B_i), i = 1, \ldots, n$:

5. MU.A requests from SP.B_i via MU.B_i to send it back together with ID.:

   MU.A → MU.B_i → SP.B_i: get challenge

6. SP.B_i generates challenge RND.SP.B_i and together with its ID, SN.SP.B_i, sends it back to MU.A:

   SP.B_i → MU.B_i → MU.A: RND.SP.B_i || SN.SP.B_i

Phase IV. Signing and Decryption

7. MU.A sends authentication request to SP.A:

   MU.A → SP.A: authenticate(RND.SP.B_1 || SN.SP.B_1 || ... || RND.SP.B_n || SN.SP.B_n)

8. SP.A generates random padding PRND.SP.A, prepares preToken.SP.A and signs it:

   DS[PrK.SP.A.AUT](preToken.SP.A)

   where:

   preToken.SP.A = textA.SP.A || PRND.SP.A

```
|| RND.SP.B_1 || SN.SP.B_1 || ...|| RND.SP.B_n
|| SN.SP.B_n || h(PRND.SP.A || RND.SP.A
|| RND.SP.B_1 || SN.SP.B_1 || ... || RND.SP.B_n
|| SN.SP.B_n) || textB.SP.A
```

and then for every $i = 1, \ldots, n$ generates additional random number `exRND.SP.B`$_i$ and calculates a ciphertext in the form:

```
E[PuK.SP.B_i.AUT](RND.SP.A || exRND.SP.B_i);
```

next `SP.A` sends it together with its signature to each `SP.B`$_i$ via `MU.B`$_i$:

```
SP.A → MU.A → MU.B_i → SP.B_i:
    E[PuK.SP.B_i.AUT] (RND.SP.A || exRND.SP.B_i)
    || DS[PrK.SP.A.AUT](preToken.SP.A)
```

9. Each `SP.B`$_i$ (for $i = 1, \ldots, n$) decrypts `E[PuK.SP.B`$_i$`.AUT] (RND.SP.A || exRND.SP.B`$_i$`) || DS[PrK.SP.A.AUT] (preToken.SP.A)` and after that `SP.B`$_i$ verifies `SP.A` signature (after the confirmation of compliance with a random challenge `RND.SP.B`$_i$ sent previously) and returns back confirmation to `MU.A`:

```
SP.B_i → MU.B_i → MU.A  SP.A: conf.OK
```

Remark 1. When the protocol is completed - the module `SP.A` and each module `SP.B`$_i$ (for $i = 1, \ldots, n$) have confidential key materials `RND.SP.A` and `exRND.SP.B`$_i$$(i = 1, ..., n)$. On this basis each party calculates:

(a) common key material:

```
K.SP.A/SP.B_1..n = KDF (RND.SP.A || RND.SP.B_1 || ... || RND.SP.B_n),
```

where `KDF` denotes a key derivation function (it is used to create session keys ensuring confidentiality and message integrity);

(b) personalised key material known only to a pair $(A, B_i), i = 1, \ldots, n$:

```
inK.SP.A/SP.B_i = KDF (K.SP.A/SP.B_1..n||exRND.SP.B_i).
```

Phase V. Key material authentication and establishment of independent trusted channels. For each pair of entities $(A, B_i), i = 1, \ldots, n$:

10. `MU.A` sends authentication request to `SP.B`$_i$ via `MU.B`$_i$:

```
MU.A → MU.B_i → SP.B_i:  MACauthenticate(RND.SP.B_i || SN.SP.B_i)
```

11. `SP.B`$_i$ calculates message authentication code and sends it back `SP.A` via `MU.B`$_i$:

```
SP.B_i → MU.B_i → MU.A → SP.A:
    MAC[inK.SP.A/SP.B_i](RND.SP.A || exRND.SP.B_i || RND.SP.B_i)
```

12. SP.A verifies MAC [inK.SP.A/SP.B$_i$] (RND.SP.A || exRND.SP.B$_i$ || RND.SP.B$_i$) and after successful verification of its compliance with received value sends it to SP.B$_i$:

$$\texttt{SP.A} \rightarrow \texttt{MU.A} \rightarrow \texttt{MU.B}_i \rightarrow \texttt{SP.B}_i\texttt{: conf.OK}$$

After the protocol completion module SP.A is authenticated mutually with every other modules SP.B$_i$($i = 1, \ldots, n$). It results from the step 8, in which module SP.A has used its private key to create a digital signature, which is then verified by each of modules SP.B$_i$ (step 9). Also, in step 9 each of entities SP.B$_i$ had to use its private key to decrypt a ciphertext received from SP.A. Thus each SP.B$_i$ might recover random numbers (RND.SP.A || exRND.SP.B$_i$) and calculate key material (K.SP.A/SP.B$_{1..n}$,inK.SP.A/SP.B$_i$). This material is used in step 11 by entity SP.B$_i$ to calculate message authentication code and then to successful verification by entity SP.A in the step 12; this ends authentication of entity SP.B$_i$ by SP.A. It is easy to notice, that key material K.SP.A/SP.B$_{1..n}$ allows to build common secure communication channel. Whereas material inK.SP.A/SP.B$_i$ is known only to a pair of entities (SP.A, SP.B$_i$) - it enables to create individual communication channels. This last property is particularly useful in the MobInfoSec system. In the system each of the entities SP.B$_i$ must send to SP.A a shadow or partially decrypted document confidentially.

3.3 Verification of Protocol's Security

In MobInfoSec system authentication protocol should work even under worst-case assumptions, namely messages may be eavesdropped or tampered by an attacker or dishonest or careless principals. The attacks can be conducted without attacking and breaking cryptography, but rather by attacking communication itself. These attacks exploit weaknesses in the protocol's design whereby protocols can be defeated by cleverly manipulating and replaying messages in the manner not anticipated by the designer [11]. Many formal methods for analysing cryptographic protocols and increasing the assurance that the protocol satisfies its security requirements exist. Some example of such methods and tools are CSP and FDR [12], OFMC [13] and the AVISPA tool [14], CryptoVerif [15], the crypto-module of the VerICS tool [16] and the PathFinder tool [17,18]. Three of these tools, i.e., AVISPA, VerICS and PathFinder, were used to investigate the main part (Steps 5-9) of proposed protocol (see Section 3.1). To model these steps we use HLPSL and ProToc languages. Next, we have examined correctness of the protocol using authentication and security properties. For all defined properties the proposed protocol is correct and secure. Computations were carried out on a computer equipped with the quad core processor Intel Pentium D (3000 MHz), 2 GB main memory, and the operating system Linux, and for each case took no more than 20 ms. More detailed description of the experiments and achieved results can be found in [19].

4 Conclusions

In this paper we have introduced a new multi-party conference authentication protocol based on ideas presented by C. Boyd [4], [7]. This protocol is a fundamental element of MobInfoSec system that enables access control according to ORCON rules to sensitive information stored and shared in encrypted form [2], [9]. The main objectives of this protocol is a mutual authentication of each pair of the protocol participants and building both common and personal secure communication channels between them. We model core security properties of proposed protocol in HSPL and ProToc languages and use the different tools, i.e., AVISPA, VerICS and PathFinder, to automate our security analysis. The protocol security analysis was conducted under the assumptions of perfect cryptography and that the protocol messages are exchanged over a network that is under the control of the Dolev-Yao intruder [20]. The extensive investigation has shown that our protocol does not contain flaws and is resistant against attacker following the Dolev-Yao model. The proposed protocol is a little more complex than Boyd's A or B protocol and: (a) it can be completed with $3n$ (n – number of protocol participants except an initiator) broadcast messages (without counting request messages), i.e., with n messages more than for Boyd's protocol, (b) the computation required for U_0 (the same as in Boyd's protocol) is one signature, n public key encryptions and (c) in opposite to Boyd's protocol, the n additional MAC calculations are required (one calculation per each entity $U_i, i = 1, \ldots, n$). However, main drawback of our protocol (like Boyd's protocol) is a lack of a forward secrecy, because the compromise of any principal's decryption key results in compromise of key materials (compare Step 9). Therefore, the future work will concentrate on extending our conference protocol to cover this drawback and to provide a forward secrecy.

Acknowledgments. This scientific research work is supported by NCBiR of Poland (grant No PBS1/B3/11/2012) in 2012-2015.

References

1. Hyla, T., Pejaś, J., El Fray, I., Maćków, W., Chocianowicz, W., Szulga, M.: Sensitive Information Protection on Mobile Devices Using General Access Structures. In: The Ninth International Conference on Systems, ICONS 2014, pp. 192–196. IARIA (2014)
2. Hyla, T., Pejaś, J.: A practical certificate and identity based encryption scheme and related security architecture. In: Saeed, K., Chaki, R., Cortesi, A., Wierzchoń, S. (eds.) CISIM 2013. LNCS, vol. 8104, pp. 190–205. Springer, Heidelberg (2013)
3. CEN, prEN 14890-1: Application Interface for smart cards used as Secure Signature Creation Devices - Part 1: Basic services (2012)
4. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer, Heidelberg (2003)
5. Dong, L., Chen, K.: Cryptographic Protocol Security Analysis Based on Trusted Freshness. Springer, Heidelberg (2012)

6. Lim, C.H., Lee, P.J.: Several practical protocols for authentication and key exchange. Information Processing Letters 53, 91–96 (1995)
7. Boyd, C., González Nieto, J.M.: Round-Optimal Contributory Conference Key Agreement. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 161–174. Springer, Heidelberg (2002)
8. Chen, Y.-Y., Lee, R.B.: Hardware-Assisted Application-Level Access Control. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 363–378. Springer, Heidelberg (2009)
9. Hyla, T., Pejaś, J.: Certificate-Based Encryption Scheme with General Access Structure. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 41–55. Springer, Heidelberg (2012)
10. ISO/IEC 11770-3:2008 Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques (2008)
11. Matsuo, S., Miyazaki, K., Otsuka, A., Basin, D.: How to Evaluate the Security of Real-Life Cryptographic Protocols? In: Sion, R., Curtmola, R., Dietrich, S., Kiayias, A., Miret, J.M., Sako, K., Sebé, F. (eds.) FC 2010 Workshops. LNCS, vol. 6054, pp. 182–194. Springer, Heidelberg (2010)
12. Ryan, P.Y.A., Schneider, S.A., Goldsmith, M.H., Lowe, G., Roscoe, A.W.: The Modelling and Analysis of Security Protocols: the CSP Approach. Addison-Wesley (2001)
13. Basin, D.M., Mödersheim, S., Viganò, L.: OFMC: A symbolic model checker for security protocols. International Journal of Information Security 4(3), 181–208 (2005)
14. Armando, A., et al.: The AVISPA tool for the automated validation of internet security protocols and applications. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 281–285. Springer, Heidelberg (2005)
15. Blanchet, B.: A computationally sound mechanized prover for security protocols. In: IEEE Symposium on Security and Privacy, Oakland, California, pp. 140–154 (2006)
16. Kurkowski, M., Penczek, W.: Verifying Security Protocols Modeled by Networks of Automata. Fundamenta Informaticae 79(3-4), 453–471 (2007)
17. Kurkowski, M., Siedlecka-Lamch, O., Szymoniak, S., Piech, H.: Parallel Bounded Model Checking of Security Protocols. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J. (eds.) PPAM 2013, Part I. LNCS, vol. 8384, pp. 224–234. Springer, Heidelberg (2013)
18. Siedlecka-Lamch, O., et al.: A New Effective Approach for Modelling and Verification of Security Protocols. In: Proc. of CS&P 2012, pp. 191–202. Humboldt University Press, Berlin (2012)
19. Kurkowski, M.: Mobile device to protect classified information (MobInfoSec). Task 3: Protocols for authentication and information security. Part 2: Formal analysis of cryptographic authentication protocols. Technical Report, TR/ZUT WI KIO ZOI 0003.02/2014, West Pomeranian University of Technology in Szczecin, Poland (2014) (in Polish)
20. Dolev, D., Yao, A.: On the security of public-key protocols. IEEE Transactions on Information Theory 29, 198–208 (1983)