Registration protocol security analysis of the electronic voting system based on blinded intermediaries using the Avispa tool

I.A. Pisarev <ilua.pisar@gmail.com>
L.K. Babenko <lkbabenko@sfedu.ru>
Department of Information Security, Southern Federal University,
Taganrog, Rostov region, 347928, Russia

Abstract. Electronic voting systems are a future alternative to traditional methods of voting. It is important to verify the main algorithms on which system security is based. This paper analyzes the security of the cryptographic protocol at the registration stage, which is used in the electronic voting system based on blind intermediaries created by the authors. The registration protocol is described, the messages transmitted between the parties are shown and their content is explained. The Dolev-Yao threat model is used during protocols modeling. The Avispa tool is used for analyzing the security of the selected protocol. The protocol is described in CAS+ and subsequently translated into the HLPSL (High-Level Protocol Specification Language) special language with which Avispa work. The description of the protocol includes roles, data, encryption keys, the order of transmitted messages between parties, parties' knowledge include attacker, the purpose of verification. The verification goals of the cryptographic protocol for resistance to attacks on authentication, secrecy and replay attacks are set. The data that a potential attacker may possess is detected. The security analysis of the registration protocol was made. The analysis showed that the objectives of the audit were put forward. A detailed diagram of the messages transmission and their contents is displayed in the presence of an attacker who performs a MITM-attack (Man in the middle). The effectiveness of protocol protection from the attacker actions is shown.

Keywords: e-voting; cryptographic protocols; cryptographic security; cryptographic protocols security verification

DOI: 10.15514/ISPRAS-2018-30(4)-10

For citation: Pisarev I.A., Babenko L.K. Registration protocol security analysis of the electronic voting system based on blinded intermediaries using the Avispa tool. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 4, 2018, pp. 155-168. DOI: 10.15514/ISPRAS-2018-30(4)-10

1. Introduction

The creation of e-voting systems is a serious problem. There are a number of ready-made systems [1,2] that are used in practice, but they are far from a sufficient level of reliability and the presence of necessary mechanisms, such as complete anonymity of the voter or vote checking opportunity after counting stage. There are also a lot of works, in which perspective methods of conducting electronic voting are considered, based on such principles as homomorphic encryption, including threshold schemes, mix-net, secret sharing schemes and others [3-16]. However, in most cases, the authors of such works show theoretical calculations, from which the basic structural unit of interaction between parties does not follow, namely, cryptographic protocol. Any method on which electronic voting is based, no matter how good it is, loses its security if there are any flaws in the structure of cryptographic protocol that lead to various attacks by the intruder. Thus, the goal of this paper is to test the cryptographic protocol in the important registration stage from various attacks, such as attack on parties' authentication, data privacy and replay-attacks using the Avispa tool [17].

2. Avispa tool

Avispa is a tool for automated security analysis of cryptographic protocols [17]. With the help of Avispa, in the context of the developed protocols, it is possible to verify the parties' authentication, the secrecy of data and protection against replay-attacks. It is impossible to perform integrity checks, in particular, used in protocol CMAC mode (Cipher-based message authentication code) using the Avispa tool. The protocol does not imply the use of timestamps in their classic implementation as a part of message. Instead, the developed system uses a temporary session control by server, in which long live sessions are broke down.

In the paper registration stage is analyzed. Three sides are modeled: user, server-intermediary and main server. The protocol will be analyzed after the phase of common session key distribution between the parties. The protocol will be described in CAS+ [18] language, then translated using the Avispa translator into HLPSL [19]. The check will be carried out using the On-the-Fly Model Checking (OFMC) module, where the verification goals are the transmitted data confidentiality and parties' authentication.

For verification, it is necessary to describe the protocol in one of the formal languages: CAS+ or HLPSL. The first language is simpler in syntax and allows you to quickly describe the protocol. An example of syntax is shown below:

```
protocol NeedhamSchroederPublicKey;
identifiers
A,B              : user;
Na,Nb            : number;
KPa,KPb  : public_key;

messages
1. A -> B        : {Na, A}KPb
2. B -> A        : {Na, Nb}KPa
3. A -> B        : {Nb}KPb

knowledge
A        : A,B,KPa,KPb;
B        : A,B,KPa,KPb;
```

```
session_instances
 [A:alice,B:bob,KPa:ka,KPb:kb];
```

The second language HLPSL is the language with which Avispa works directly. An example of syntax is shown below:

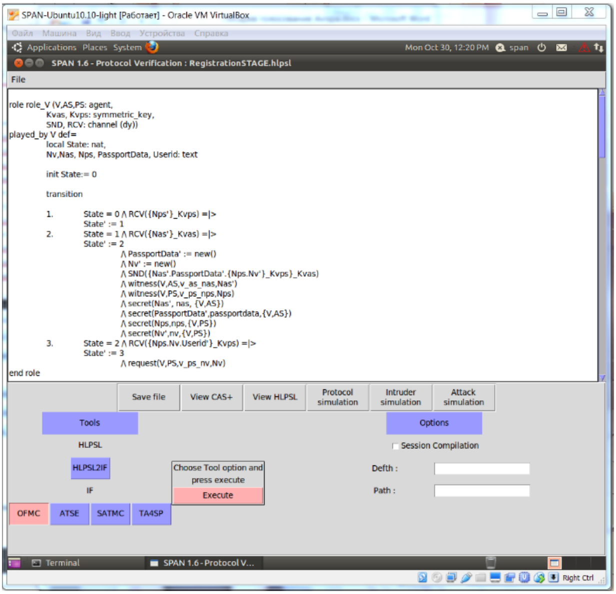
```
role Alice (A, B: agent,
             KPa, KPb: public_key,
             SND, RCV: channel (dy))
    played_by A def=
    transition

    0.  State  = 0 /\ RCV(start) =|>
        State':= 2 /\ Na' := new() /\ SND({Na'.A}_KPb)
                  ...

 role Bob(A, B: agent,
          KPa, KPb: public_key,
          SND, RCV: channel (dy))
    ...
```

The syntax of this language is more difficult and the best way to describe the protocol is to describe it in CAS+, and then use Avispa to convert it to HLPSL. It is worth to say that if the more complex and larger your protocol, then there is greater chance of errors occurring during translation, so after that you need manually to fix some fragments in HLPSL. It is also worth to say that you should not describe the goals of checking in CAS +, but rather add them directly in HLPSL.

During protocols describing, the following entities are used: roles, data, message order, sessions and verification purposes. After the description of the protocol, including the indication of verification objectives, it is possible to analyze protocol security against attacks. For analyzing, you can use different modes, but the most effective is the OFMC mode (see Fig. 1).

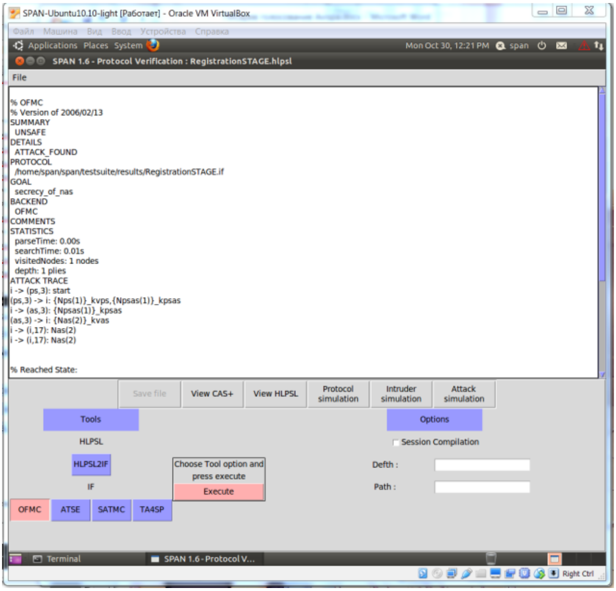
It requires an additional specification for all data involved in the verification, as well as the message area where verification is required for party authentication. As a result of verification, the corresponding result will be issued. In case of attacks detection, the type of attack and its progress will appear in the form of corresponding changes in messages by the intruder, as in Fig. 2.

If there are no attacks, then the program output will contain a corresponding message that protocol is safe (see Fig. 3). Using the «Protocol simulation» button, you can see the interaction scheme of the parties in your protocol. With the help of the button «Intruder simulation» such a scheme will appear, only with the participation of the intruders’ side, in which the data intercepted by him will appear. With the help of the button «Attack simulation» you can see the scheme of the attack with intruder, provided that there is an attack in your protocol.


SPAN-Ubuntu10.10-light [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Applications Places System
Mon Oct 30, 12:20 PM span
SPAN 1.6 - Protocol Verification : RegistrationSTAGE.hlpsl
File
Save file
View CAS+
View HLPSL
Protocol simulation
Intruder simulation
Attack simulation
Tools
Options
HLPSL
□ Session Compilation
HLPSL2IF
Choose Tool option and press execute
Execute
Defth :
Path :
IF
OFMC
ATSE
SATMC
TA4SP
Terminal
SPAN 1.6 - Protocol V...
Right Ctrl

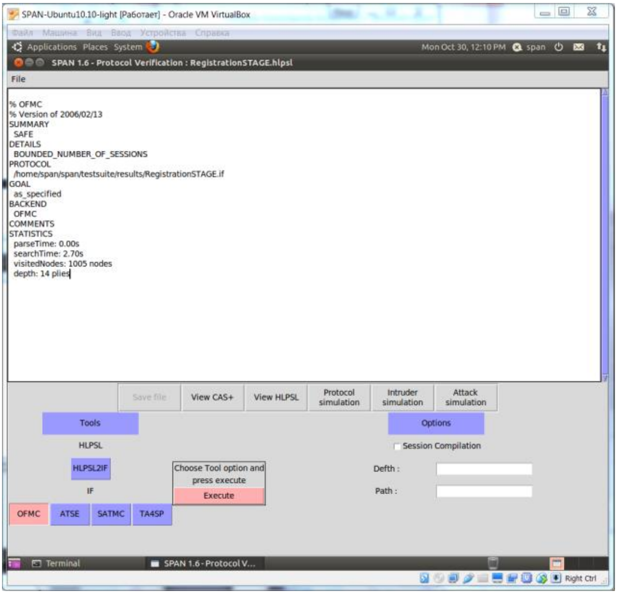
```
role role_V (V,AS,PS: agent,
          Kvas, Kvps: symmetric_key,
          SND, RCV: channel (dy))
played_by V def=
          local State: nat,
          Nv,Nas, Nps, PassportData, Userid: text

          init State:= 0

          transition

          1.        State = 0 /\ RCV({Nps'}_Kvps) =|>
                    State' := 1
          2.        State = 1 /\ RCV({Nas'}_Kvas) =|>
                    State' := 2
                              /\ PassportData' := new()
                              /\ Nv' := new()
                              /\ SND({Nas'.PassportData'.{Nps.Nv'}_Kvps}_Kvas)
                              /\ witness(V,AS,v_as_nas,Nas')
                              /\ witness(V,PS,v_ps_nps,Nps)
                              /\ secret(Nas', nas, {V,AS})
                              /\ secret(PassportData',passportdata,{V,AS})
                              /\ secret(Nps,nps,{V,PS})
                              /\ secret(Nv',nv,{V,PS})
          3.        State = 2 /\ RCV({Nps.Nv.Userid'}_Kvps) =|>
                    State' := 3
                              /\ request(V,PS,v_ps_nv,Nv)
end role
```

Fig. 1. UI


SPAN-Ubuntu10.10-light [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Applications Places System
Mon Oct 30, 12:21 PM span
SPAN 1.6 - Protocol Verification : RegistrationSTAGE.hlpsl
File
Save file
View CAS+
View HLPSL
Protocol simulation
Intruder simulation
Attack simulation
Tools
Options
HLPSL
□ Session Compilation
HLPSL2IF
Choose Tool option and press execute
Execute
Defth :
Path :
IF
OFMC
ATSE
SATMC
TA4SP
Terminal
SPAN 1.6 - Protocol V...
Right Ctrl

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/RegistrationSTAGE.if
GOAL
  secrecy_of_nas
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (ps,3): start
(ps,3) -> i: {Nps(1)}_kvps,{Npsas(1)}_kpsas
i -> (as,3): {Npsas(1)}_kpsas
(as,3) -> i: {Nas(2)}_kvas
i -> (i,17): Nas(2)
i -> (i,17): Nas(2)


% Reached State:
```

Fig. 2 – Founded attack trace


SPAN-Ubuntu10.10-light [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Applications Places System
Mon Oct 30, 12:10 PM span
SPAN 1.6 - Protocol Verification : RegistrationSTAGE.hlpsl
File
Save file
View CAS+
View HLPSL
Protocol simulation
Intruder simulation
Attack simulation
Tools
Options
HLPSL
Session Compilation
HLPSL2IF
Choose Tool option and press execute
Execute
Defth :
Path :
IF
OFMC
ATSE
SATMC
TA4SP
Terminal
SPAN 1.6 - Protocol V...
Right Ctrl

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/RegistrationSTAGE.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 2.70s
 visitedNodes: 1005 nodes
 depth: 14 plies
```

Fig. 3 – Result after verification of safe protocol

2. E-voting system description

2.1 System architecture

The system architecture is based on the use of the following components: client application for voter - V, 3 server applications that will be located on different physical machines: AS (authentication server), PS (processing server), VS (voting server), encryption application for the passport database and ballots DBE (database encryptor). The general scheme of the interaction of components is shown in Fig. 4. The basic principle on which the system protocols are based - blinded intermediaries (see Fig. 5).

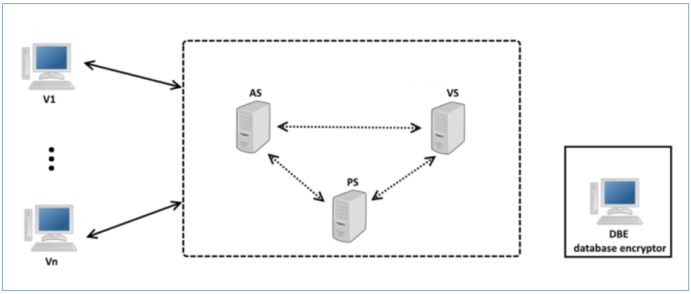


Fig. 4 – System architecture

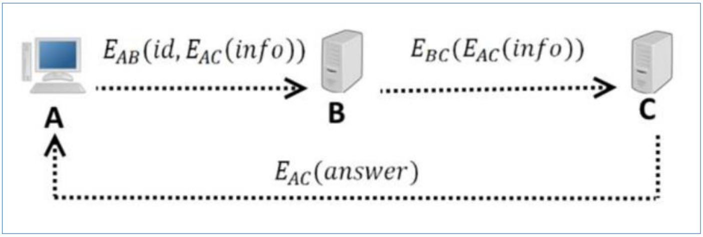


Fig. 5 – Blinded intermediaries principle

There are 3 interacting sides A, B, C. Using the protocol for generating a common secret key, the session key AB, BC, AC are generated. A encrypts some information *info* on the AC key, appends an *id* to it, encrypts it on the AB key and sends this message to B. B in this case is a blinded intermediary, because it can decrypt only the first part of the message with *id*, and the remainder with *info* can not. It accepts the message, decrypts and checks if *id* is in the database and, then redirects the remainder of the message encrypted again on the BC key to the C side. C receives the message, decrypts *info*, encrypts the answer response on the AC key and sends it to A. This principle ensures that: *info* will be accepted only if *id* is in the database and that it is impossible to correlate *id* with *info*.

2.2 Stages description

Stages of electronic voting in the context of the system:

- Preparation. At this stage, a database of voters and a ballot are created. This data is encrypted, and officials deliver this data to the appropriate server components of the system.
- Registration. At this stage, users log in to the system using their identification data, at the moment - using passport data, and they get their anonymous identifier. It should be noted that by using the previously described principle of blind intermediaries, it is impossible to correlate open passport data with an anonymous identifier, which ensures the requirement of anonymity.
- Voting. Users receive a ballot, make their choice and send filled ballot with their anonymous identifier to the server. If such an identifier is present, the vote is accepted, and the verification identifier is sent to user, with which he or she can check vote after counting stage. It is worth noting that it is very important that the user can check his vote after the counting.
- Counting results and votes checking. At the last stage, the votes are counted, the results are published in the public domain, and any voted user can check his or her vote with a verification identifier.

4. Registration stage

The electronic voting system based on blind intermediaries, includes a registration stage in which the voter is given anonymous identifier after presenting his passport data. A simplified scheme of the registration stage is shown in Fig. 6.

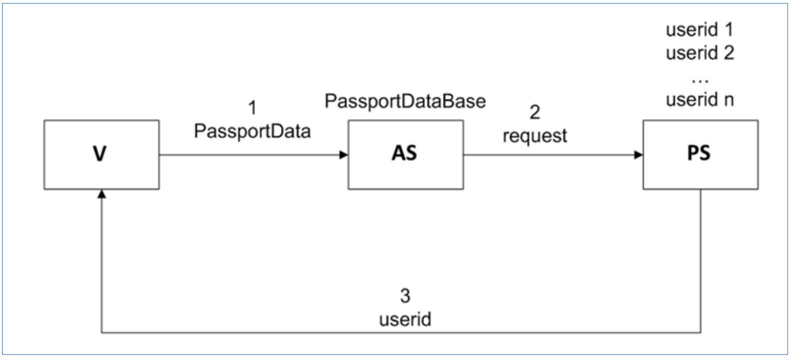


Fig. 6 – Simplified scheme of registration stage.

Secret keys V, VAS, VPS are generated using the protocol for generating a common session key. The server parties generate random numbers and send messages (1), (2), (3) to their recipients. They will be used for parties' authentication. V generates Nv. Next, it generates a message (4) with the passport data, which is a hash from a set of document fields, encrypted random numbers on the shared secret key VPS, calculates the CMAC, encrypts all this data on VAS key, calculates the CMAC and sends to AS. AS in this case is a blinded intermediary. It checks the message integrity by CMAC checking, searches *PassportData* in the database and, if successful, redirects another part of the message (5) to side PS. PS checks integrity, if successful, generates *userid*, adds it to database and sends to V as a message (6). The voter decrypts the message, checks integrity and values of random numbers, and remembers his anonymous unique identifier *userid*, with which the user can vote.

ECDHE (V, AS) - vas
ECDHE (V, PS) – vps
ECDHE (PS, AS) – psas
V: генерирует N_{as}
(1) AS -> V: $E_{vas}(N_{as})$
PS: generates N_{ps}
(2) PS -> V: $E_{vps}(N_{ps})$
PS: generates N_{psas}
(3) PS -> AS: $E_{psas}(N_{psas})$
V: generates N_v.

(4) V -> AS: E_{vas} (N_{as}, PassportData, E_{vps} (N_{ps}, N_{v},), CMAC1), CMAC2

AS -> V: "Success"

(5) AS -> PS: E_{psas} (N_{psas}, E_{vps}(N_{ps}, N_{v}), CMAC1), CMAC3

PS: generates userid

(6) PS -> V: E_{vps} (N_{ps}, N_{v}, userid), CMAC4

ECDHE is a Diffie-Hellman protocol on elliptical curves using ephemeral keys. In our case, we use a modified version of ECDHE-RSA, where authentication is done using a signature RSA and a server certificate which help to prevent MIMT (man in the middle) attacks. The protocol description is as follows.

ECDHE:

(1) V -> S: "Hello"

(2) S -> V: DHs,$Sign_{SKs}$(DHs),Certificate

(3) S: Проверяет Certificate и подпись $Sign_{SKs}$(DHs)

(4) V -> S: DHv

(5) Both sides generate a common session key K for further interaction with a symmetric cipher.

Here V is the client, S is the trusted server that has the certificate, DHs is the server secret part, DHv is the client secret part $Sign_{SKs}$(DHs) is the signature with the server's private key SKs, Certificate is the server certificate.

When servers generate common secret key, the same protocol is used, except that both parties exchange certificates and if they are valid, a common session key is generated. The security verification of the registration protocol will be carried out after this stage.

5. Security analysis of registration protocol using Avispa tool

Consider the description of the protocol in CAS + at the registration stage.

```
protocol EVotingRegistration;
identifiers
V,AS,PS                                    : user;
Nas,Nps,Npsas,Nv,PassportData,Userid       : number;
Kvas,Kvps,Kpsas                            : symmetric_key;

messages
1. PS -> V    : {Nps}Kvps
2. PS -> AS   : {Npsas}Kpsas
3. AS -> V    : {Nas}Kvas
4. V -> AS    : {Nas,PassportData,{Nps,Nv}Kvps}Kvas
5. AS -> PS   : {Npsas,{Nps,Nv}Kvps}Kpsas
6. PS -> V    : {Nps,Nv,Userid}Kvps

knowledge
```

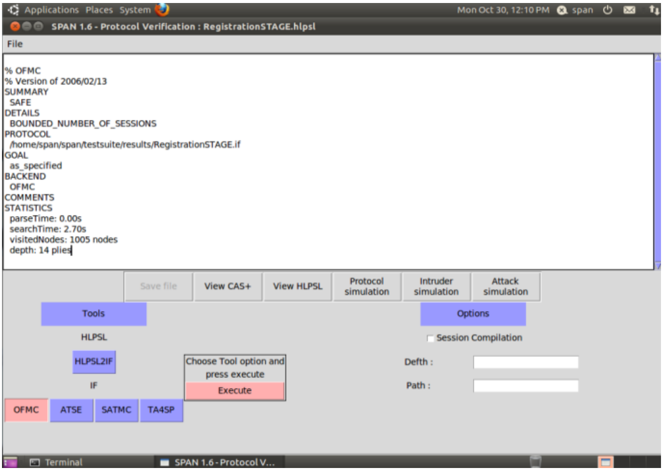
```
16   V      : V,AS,PS,Nas,Nps,Nv,PassportData,Userid,Kvas,Kvps
17   PS     : V,AS,PS,Nps,Npsas,PassportData,Kvas,Kpsas
18   VS     : V,AS,PS,Npsas,Nps,Nv,Userid,Kvps,Kpsas
19
20   session_instances
21    [V:v,AS:as,VS:ps,Kvas:kvas,Kvps:kvps,Kpsas:kpsas]
22    [V:v,AS:as,VS:ps,Kvas:kvas,Kvps:kvps,Kpsas:kpsas];
23
24   intruder_knowledge
25    v,as,ps;
26
27   goal
28    secrecy_of Nps [V,PS];
29    secrecy_of Npsas [AS,PS];
30    secrecy_of Nas [V,AS];
31    secrecy_of Nv [V,PS];
32    secrecy_of PassportData [V,AS];
33    secrecy_of Userid [V,PS];
34    AS authenticates V on Nas;
35    PS authenticates AS on Npsas;
36    PS authenticates V on Nps;
37    V authenticates PS on Nv;
```

Three interacting parties are described as roles: V, AS, PS (lines 2-3). The identifiers section describes the objects participating in the protocol: interacting parties (line 3), random numbers for authentication, identifiers (line 4). Symmetric keys are specified that will be used for message encryption (line 5). The messages section (lines 7-13) describes the transfer of messages between roles, which data is transmitted, and on which key it encrypted. The knowledge section (lines 15-18) describes roles' data knowledge during the execution of the protocol. In the *session_instances* section (lines 20-22), sessions are described. Among the simulated sessions, 2 are allocated, which allow simulating interaction of two clients with the system. This will detect possible attacks on the parties' authentication and replay-attacks. The *intruder_knowledge* section (lines 24-25) specifies the original knowledge of the intruder. In the goal section (lines 27-37) the secrecy of important values is indicated and the authentication according to the request-response scheme with the transfer of random numbers between the participants. For secrecy of the value, it is necessary that this variable is encrypted and that the encryption key does not come to intruder. In order for one party to authenticate another using the request-response mechanism, it is required that the party wanting to authenticate send a random number to the other party, and that other party in the response message returns this random number. In this protocol there are 4 such actions:

- AS authenticates V by Nas;
- PS authenticates AS by Npsas;
- PS authenticates V by Nps;

- V authenticates the PS to Nv.

As for replay-attacks, protection against them is possible due to the presence of a random number at the beginning of each message, which each side checks when message is received. The results of the check using the OFMC module are shown in Fig. 7. Fig. 8 shows the scheme of interaction between the parties at the stage of registration by steps. Fig. 9 shows the interaction scheme in the presence of an intruder (*Intruder_* side, highlighted in red). This scheme is a visual implementation of the attack man in the middle. When transmitting messages during execution, a transition is made from the «Incoming events» area to «Past events», and the format is the direction of message transfer (from whom and to whom) and the message itself. We can see from the simulation results in the field of intercepted data «Intruder knowledge», all transmitted messages are encrypted on keys which intruder doesn’t know, and it excludes the possibility in any way to get important information, such as the user's passport data or unique identifier. The record «nonce-N» means some data that is not readable. Because of the analysis, it was revealed that the registration protocol is safe, ensures the fulfillment of the security objectives (properties) set in the protocol analysis: securing data, authentication of the parties, protection against replay-attacks.

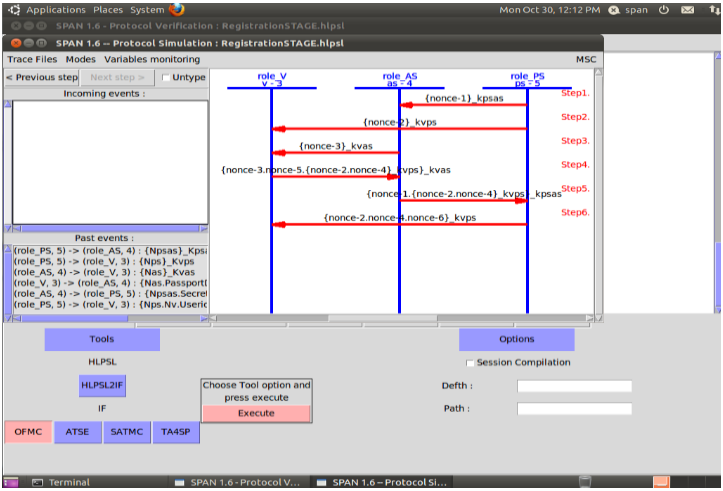


```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/RegistrationSTAGE.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 2.70s
  visitedNodes: 1005 nodes
  depth: 14 plies
```

Fig. 7 – Registration protocol verification using OFMC mode.

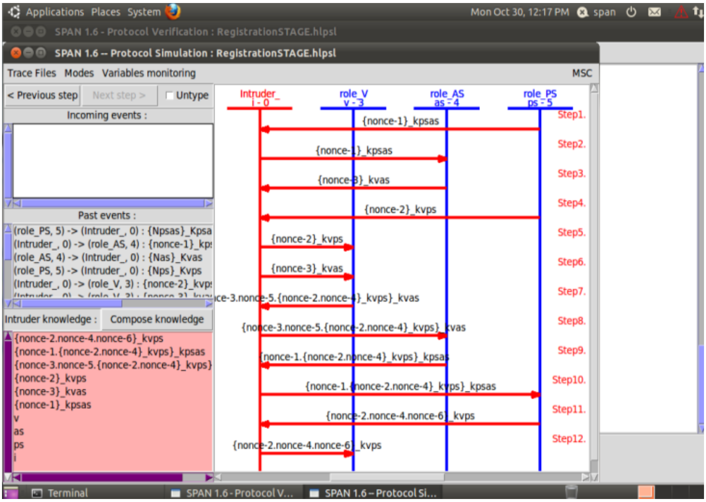
6. Conclusion

The automated security verification tool Avispa was used for security verification of the registration protocol in electronic voting system based on blind intermediaries, in this paper. The protocol was described in the formal languages CAS+ and HLPSL. The secrecy properties of the transmitted data between the interacting parties were analyzed. It was shown that set security objectives: parties’ authentication, verification of data privacy and protection from replay attacks were achieved. The scheme of parties’ interaction with the help of tools’ graphical functional was considered. An analysis of messages that an intruder can intercept was carried out. Based on the graphical representation it was revealed that all transmitted data is secure, because all messages are encrypted on unknown for intruder keys.


Applications Places System
Mon Oct 30, 12:12 PM
span
SPAN 1.6 - Protocol Verification : RegistrationSTAGE.hlpsl
SPAN 1.6 -- Protocol Simulation : RegistrationSTAGE.hlpsl
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
□ Untype
Incoming events :
Past events :
role_V v - 3
role_AS as - 4
role_PS ps - 5
Step1. {nonce-1}_kpsas
Step2. {nonce-2}_kvps
Step3. {nonce-3}_kvas
Step4. {nonce-3.nonce-5.{nonce-2.nonce-4}_kvps}_kvas
Step5. {nonce-1.{nonce-2.nonce-4}_kvps}_kpsas
Step6. {nonce-2.nonce-4.nonce-6}_kvps
Tools
HLPSL
HLPSL2IF
IF
OFMC
ATSE
SATMC
TA4SP
Choose Tool option and press execute
Execute
Options
□ Session Compilation
Defth :
Path :
Terminal
SPAN 1.6 - Protocol V...
SPAN 1.6 – Protocol Si...

```
(role_PS, 5) -> (role_AS, 4) : {Npsas}_Kpsa
(role_PS, 5) -> (role_V, 3) : {Nps}_Kvps
(role_AS, 4) -> (role_V, 3) : {Nas}_Kvas
(role_V, 3) -> (role_AS, 4) : {Nas.Passport(
(role_AS, 4) -> (role_PS, 5) : {Npsas.Secre
(role_PS, 5) -> (role_V, 3) : {Nps.Nv.Useric
```

Fig. 8 – Registration protocol in "Protocol simulation" mode.


Applications Places System
Mon Oct 30, 12:17 PM
span
SPAN 1.6 - Protocol Verification : RegistrationSTAGE.hlpsl
SPAN 1.6 -- Protocol Simulation : RegistrationSTAGE.hlpsl
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
□ Untype
Incoming events :
Past events :
Intruder knowledge :
Compose knowledge
Intruder_ i - 0
role_V v - 3
role_AS as - 4
role_PS ps - 5
Step1. {nonce-1}_kpsas
Step2. {nonce-1}_kpsas
Step3. {nonce-3}_kvas
Step4. {nonce-2}_kvps
Step5. {nonce-2}_kvps
Step6. {nonce-3}_kvas
Step7. {nonce-3.nonce-5.{nonce-2.nonce-4}_kvps}_kvas
Step8. {nonce-3.nonce-5.{nonce-2.nonce-4}_kvps}_kvas
Step9. {nonce-1.{nonce-2.nonce-4}_kvps}_kpsas
Step10. {nonce-1.{nonce-2.nonce-4}_kvps}_kpsas
Step11. {nonce-2.nonce-4.nonce-6}_kvps
Step12. {nonce-2.nonce-4.nonce-6}_kvps
Terminal
SPAN 1.6 - Protocol V...
SPAN 1.6 – Protocol Si...

```
(role_PS, 5) -> (Intruder_, 0) : {Npsas}_Kpsa
(Intruder_, 0) -> (role_AS, 4) : {nonce-1}_kp
(role_AS, 4) -> (Intruder_, 0) : {Nas}_Kvas
(role_PS, 5) -> (Intruder_, 0) : {Nps}_Kvps
(Intruder_, 0) -> (role_V, 3) : {nonce-2}_kvp
```

```
{nonce-2.nonce-4.nonce-6}_kvps
{nonce-1.{nonce-2.nonce-4}_kvps}_kpsas
{nonce-3.nonce-5.{nonce-2.nonce-4}_kvps}
{nonce-2}_kvps
{nonce-3}_kvas
{nonce-1}_kpsas
v
as
ps
i
```

Fig. 9 – Registration protocol verification using OFMC mode.

Acknowledgment

The work was supported by the Ministry of Education and Science of the Russian Federation grant № 2.6264.2017/8.9.

References

[1]. Overview of e-voting systems, NICK Estonia. Estonian National Electoral Commission. Tallinn 2005.

[2]. Dossogne J., Lafitte F. Blinded additively homomorphic encryption schemes for self-tallying voting. Journal of Information Security and Applications, vol. 22, 2015, pp. 40-53.

[3]. Izabachene M. A Homomorphic LWE Based E-voting Scheme. In Proc. of the 7th International Workshop on Post Quantum Cryptography, 2016, pp 245-265.

[4]. Hirt M., Sako K. Efficient receipt-free voting based on homomorphic encryption. In Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques, 2000, pp. 539-556.

[5]. Rivest L. R. et al. Lecture notes 15: Voting, homomorphic encryption. MIT, 2002. Available at http://web.mit.edu/6.857/OldStuff/Fall02/handouts/L15-voting.pdf, accessed 10.06.2018.

[6]. Ben Adida, Mixnets in Electronic Voting, Cambridge University, 2005. Available at http://assets.adida.net/presentations/cambridge-university-voting-2005-01-18.pdf, accessed 10.06.2018.

[7]. Electronic elections: fear of falsification of the results. Kazakhstan today, 2004. Available at http://profit.kz/news/91/Elektronnie-vibori-opasenie-falsifikacii-rezultatov/, accessed 10.06.2018 (in Russian).

[8]. Lipen V.Y., Voronetsky M.A., Lipen D.V., Polevikov E.L. Technology and results of testing electronic voting systems. Ob’edinennyj institut problem informatiki NAN Belarusi [United Institute of Informatics Problems NASB], 2002. Available at http://uiip.bas-net.by/structure/l_kg/results_testing_technology.php/, accessed 10.06.2018 (in Russian).

[9]. David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, vol. 24, no. 2, 1981, pp. 84-90.

[10]. Ali S. T., Murray J. An Overview of End-to-End Verifiable Voting Systems. arXiv preprint arXiv: 1605.08554, 2016.

[11]. Smart M., Ritter E. True trustworthy elections: remote electronic voting using trusted computing. In Proc. of the International Conference on Autonomic and Trusted Computing, 2011, pp.187-202.

[12]. Bruck S., Jefferson D., Rivest R.L. A modular voting architecture («frog voting»). Towards trustworthy elections. LNCS, volume 6000, 2010, pp. 97-106.

[13]. Jonker H., Mauw S., Pang J. Privacy and verifiability in voting systems: Methods, developments and trends. Computer Science Review, vol. 10, 2013, pp. 1-30.

[14]. Shubhangi S. Shinde, Sonali Shukla, Prof. D. K. Chitre. Secure E-voting Using Homomorphic Technology, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 8, 2013, pp. 203-206.

[15]. Neumann S., Volkamer M. Civitas and the real world: problems and solutions from a practical point of view. In Proc. of the Seventh International Conference on Availability, Reliability and Security (ARES), 2012, pp. 180-185.

[16]. Yi X., Okamoto E. Practical remote end-to-end voting scheme. In Proc. of the International Conference on Electronic Government and the Information Systems Perspective, 2011, pp. 386-400.

[17]. The AVISPA team, The High Level Protocol Specification Language. Available at http://www.avispa-project.org/, accessed 10.06.2018

[18]. Ronan Saillard, Thomas Genet, CAS+, March 21, 2011. Available at http://people.irisa.fr/Thomas.Genet/span/CAS_manual.pdf, accessed 10.06.2018

[19]. D. Basin, S. M¨odersheim, and L. Vigan`o. OFMC: A Symbolic Model- Checker for Security Protocols. International Journal of Information Security, vol. 4, issue 3, 2004, pp 181–208.

[20]. L.K. Babenko, I.A. Pisarev, O.B. Makarevich. Secure electronic voting using blinded intermediaries. Isvestiya SFedU. Engineering sciences, no. 5, 2017, pp. 6-15 (in Russian).

Анализ безопасности протокола регистрации в системе электронного голосования на основе слепых посредников с помощью инструмента Avispa

И.А. Писарев <ilua.pisar@gmail.com>
Л.К. Бабенко <lkbabenko@sfedu.ru>
Кафедра информационной безопасности,
Южный Федеральный Университет,
Таганрог, Ростовская область, 347928, Россия

Аннотация. Системы электронного голосования являются будущей альтернативой традиционным способам проведения голосования. Как и для любой системы, важным является верификация ключевых алгоритмов, на которых основана её безопасность. В работе рассматривается анализ безопасности криптографического протокола на этапе регистрации, который используется в созданной авторами системе электронного голосования на основе слепых посредников. Проведено описание протокола регистрации, показаны передаваемые между сторонами сообщения и объяснено их содержимое. При моделировании протоколов предполагается использование модели угроз Долева-Яо. В качестве инструмента для анализа безопасности выбранного протокола используется система Avispa. Протокол описан на языке CAS+ и впоследствии транслирован в специальный язык HLPSL (High−Level Protocol Specification Language), с которым работает используемый инструмент. Описание протокола включает в себя роли, данные, ключи шифрования, порядок передаваемых сообщений между сторонами, знание сторон и злоумышленника, цели проверки. Поставлены цели верификации криптографического протокола на устойчивость к атакам на аутентификацию, секретность и replay-атакам. Установлены данные, которыми может владеть потенциальный злоумышленник. Произведен анализ безопасности протокола регистрации. Анализ показал, что выдвинутые цели проверки были достигнуты. Отображена подробная схема передачи сообщений и их содержимого при наличии злоумышленника, осуществляющего MITM-атаку (Man in the middle). Показана эффективность защиты протокола от действий злоумышленника.

Ключевые слова: электронное голосование; криптографические протоколы; криптографическая защита; верификация безопасности криптографических протоколов

DOI: 10.15514/ISPRAS-2018-30(4)-10

Для цитирования: Писарев И.А., Бабенко Л.К. Анализ безопасности протокола регистрации в системе электронного голосования на основе слепых посредников с помощью инструмента Avispa. Труды ИСП РАН, том 30, вып. 4, 2018 г., стр. 155-168 (на английском языке). DOI: 10.15514/ISPRAS-2018-30(4)-10

Список литературы

[1]. Overview of e-voting systems, NICK Estonia. Estonian National Electoral Commission. Tallinn 2005.

[2]. Dossogne J., Lafitte F. Blinded additively homomorphic encryption schemes for self-tallying voting. Journal of Information Security and Applications, vol. 22, 2015, pp. 40-53.

[3]. Izabachene M. A Homomorphic LWE Based E-voting Scheme. In Proc. of the 7th International Workshop on Post Quantum Cryptography, 2016, pp 245-265.

[4]. Hirt M., Sako K. Efficient receipt-free voting based on homomorphic encryption. In Proc. of the International Conference on the Theory and Applications of Cryptographic Techniques, 2000, pp. 539-556.

[5]. Rivest L. R. et al. Lecture notes 15: Voting, homomorphic encryption. MIT, 2002. Режим доступа: http://web.mit.edu/6.857/OldStuff/Fall02/handouts/L15-voting.pdf, accessed 10.06.2018.

[6]. Ben Adida, Mixnets in Electronic Voting, Cambridge University, 2005. Режим доступа: http://assets.adida.net/presentations/cambridge-university-voting-2005-01-18.pdf, accessed 10.06.2018.

[7]. Электронные выборы: опасение фальсификации результатов. Казахстан сегодня, 2004. Режим доступа: http://profit.kz/news/91/Elektronnie-vibori-opasenie-falsifikacii-rezultatov/, дата обращения 10.06.2018

[8]. Липень В.Ю., Воронецкий М.А., Липень Д.В., Полевиков Э.Л. Результаты апробирования технологий и систем электронного голосования. Объединенный институт проблем информатики НАН Беларуси, 2002. Режим доступа: http://uiip.bas-net.by/structure/l_kg/results_testing_technology.php/, дата обращения 10.06.2018

[9]. David L Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, vol. 24, no. 2, 1981, pp. 84-90.

[10]. Ali S. T., Murray J. An Overview of End-to-End Verifiable Voting Systems. arXiv preprint arXiv: 1605.08554, 2016.

[11]. Smart M., Ritter E. True trustworthy elections: remote electronic voting using trusted computing. In Proc. of the International Conference on Autonomic and Trusted Computing, 2011, pp.187-202.

[12]. Bruck S., Jefferson D., Rivest R.L. A modular voting architecture («frog voting»). Towards trustworthy elections. LNCS, volume 6000, 2010, pp. 97-106.

[13]. Jonker H., Mauw S., Pang J. Privacy and verifiability in voting systems: Methods, developments and trends. Computer Science Review, vol. 10, 2013, pp. 1-30.

[14]. Shubhangi S. Shinde, Sonali Shukla, Prof. D. K. Chitre. Secure E-voting Using Homomorphic Technology, International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 8, 2013, pp. 203-206.

[15]. Neumann S., Volkamer M. Civitas and the real world: problems and solutions from a practical point of view. In Proc. of the Seventh International Conference on Availability, Reliability and Security (ARES), 2012, pp. 180-185.

[16]. Yi X., Okamoto E. Practical remote end-to-end voting scheme. In Proc. of the International Conference on Electronic Government and the Information Systems Perspective, 2011, pp. 386-400.

[17]. The AVISPA team, The High Level Protocol Specification Language. Available at http://www.avispa-project.org/, accessed 10.06.2018

[18]. Ronan Saillard, Thomas Genet, CAS+, March 21, 2011. Available at http://people.irisa.fr/Thomas.Genet/span/CAS_manual.pdf, accessed 10.06.2018

[19]. D. Basin, S. M¨odersheim, and L. Vigan`o. OFMC: A Symbolic Model- Checker for Security Protocols. International Journal of Information Security, vol. 4, issue 3, 2004, pp 181–208.

[20]. Л.К. Бабенко, И.А. Писарев, О.Б. Макаревич. Защищенное электронное голосование с использованием слепых посредников, Известия ЮФУ. Технические науки, №5, 2017 г., стр. 6-15.