Набиев Бабак Расим оглы – e-mail: babek@iit.science.az; AZ1141, Азербайджанская Республика, г. Баку, ул. Б. Вахабзаде 9; тел: 994125390167.

Alguliyev Rasim Mahammad – Institute of Information Technology of ANAS; rasim@science.az; AZ1141, B. Vahabzade street, 9A, Azerbaijan Republic, Baku; phone: 994125390167, Active member of ANAS, doctor of technical sciences, Professor.

Imamverdiyev Yadigar Nasib – e-mail: rasim@science.az; phone: 994125104253; dr. of tech. sc., professor.

Nabiyev Babak Rasim – e-mail: rasim@science.az; phone: 994125390167; postgraduate student.

УДК 004.056

Л.К. Бабенко, И.А. Писарев

АНАЛИЗ БЕЗОПАСНОСТИ ПРОТОКОЛА СИСТЕМЫ ЭЛЕКТРОННОГО ГОЛОСОВАНИЯ НА ОСНОВЕ СЛЕПЫХ ПОСРЕДНИКОВ С ПОМОЩЬЮ ИНСТРУМЕНТА AVISPA*

Разработка систем электронного голосования является важной проблемой в современном мире. Такие системы надежнее и удобнее традиционных способов голосования. Однако, их разработка является гораздо более сложной и доказать, что какая-либо система является надежной на достаточном уровне так же крайне сложно. В данной работе рассматривается анализ безопасности криптографического протокола, который используется в созданной авторами системе электронного голосования на основе слепых посредников. Анализируется протокол на самом ключевом этапе системы – голосования. Проведено описание протокола. Показан ход преобразования данных в процессе взаимодействия сторон во время этапа голосования. Указаны уточнения по поводу использования тех или иных техник для обеспечения защищенности информации на всем протяжении этапа голосования. Проверяется защищенность криптографического протокола на этом этапе. В качестве инструмента для верификации безопасности протоколов используется система Avispa. В статье приводится описание протокола на специальном языке CAS+, которое преобразуется в язык HLPSL (High−Level Protocol Specification Language) и анализируется данным инструментом. Поставлены цели анализа безопасности разработанного протокола такие как: аутентификация сторон, проверка секретности данных, защита от replay-атак. Приведены особенности описания протоколов с помощью инструмента Avispa. Произведена проверка безопасности протокола системы электронного голосования на основе слепых посредников, рассмотрена схема взаимодействия сторон, включая анализ сообщений, которые может перехватить злоумышленник. Показана эффективность защиты криптографического протокола от действий злоумышленника. Сделаны выводы по использованию инструмента Avispa для анализа безопасности протоколов.

Электронное голосование; криптографические протоколы; криптографическая защита; верификация безопасности криптографических протоколов.

L.K. Babenko, I.A. Pisarev

PROTOCOL SECURITY ANALYSIS OF ELECTRONIC VOTING SYSTEM BASED ON BLIND INTERMEDIARIES WITH THE AVISPA TOOL

The development of electronic voting systems is an important problem in the modern world. Such systems are more reliable and convenient than traditional methods of voting. However, their development is much more complicated and to prove that any system is reliable at a sufficient level is also extremely difficult. In this paper, we analyze the security of a cryptographic protocol,

* Работа поддержана грантом Министерства образования и науки РФ № 2.6264.2017/8.9.

which is used in the electronic voting system created by the authors on the basis on blind intermediaries. The protocol is analyzed at the most crucial stage of the system - voting. Protocol is described. Data transformation during interaction of the parties in the voting stage is shown. Specifications are given regarding the use of certain techniques to ensure the security of information throughout the voting stage. Cryptographic protocol security is checked at this stage. The Avispa system is used as a tool for protocol security verification. The article describes protocol in the special language CAS+, which is converted to HLPSL (High−Level Protocol Specification Language) and analyzed by this tool. Security analysis goals of the developed protocol are set, such as: parties authentication, data privacy verification, protection against replay-attacks. The description features of protocols with the Avispa tool are given. Protocol security of electronic voting system based on blind intermediaries is checked, parties' interaction scheme is examined, including messages analysis that the attacker could intercept. The effectiveness of cryptographic protocol protection from the attackers' actions is shown. Conclusions are drawn on Avispa tool using for protocol security analyzing.

Electronic voting; cryptographic protocols; cryptographic protection; security verification of cryptographic protocols.

Введение. Создание систем электронного голосования является нетривиальной задачей. Существует ряд готовых систем [1, 2], которые используются на практике, но они далеки от достаточного уровня надежности и присутствия необходимых механизмов, таких как полная анонимность голосующего или возможность проверить свой голос после подсчета голосов. Так же существует множество работ, в которых рассматриваются перспективные методы проведения электронного голосования, основанные на таких принципах как гомоморфное шифрование, включая пороговые схемы, mix-net, схемы разделения секрета и другие [3–18]. Однако в большинстве случаев авторы таких работ показывают теоретические выкладки, из которых не вытекает основная структурная единица взаимодействия сторон, а именно – криптографический протокол. Любой метод, на котором базируется электронное голосование, каким бы хорошим не был, теряет свой смысл, если возможны какие-либо недоработки в структуре криптографического протокола, приводящие к различным атакам злоумышленника. Таким образом, целью данной работы является проверка криптографического протокола на ключевом этапе голосования в системе на основе слепых посредников [19] от различных атак, таких как: атака на аутентификацию сторон, секретность данных и replay-атаки с помощью инструмента Avispa.

Инструмент Avispa. Avispa [20] представляет собой инструмент для автоматизированного анализа безопасности криптографических протоколов. Реализована под ОС linux. Архитектура включает в себя: транслятор HLPSL2IF, который переводит описание протокола из HLPSL в специальный IF (Intermediate format), 4 модуля верификации OFMC (on-the-fly Model-Checker), AtSe (CL-based Attack Searcher), SATMC (SAT-based Model-Checker), TA4SP (Tree Automata-based Protocol Analyser). Принцип работы применяемого в данной работе модуля OFMC [21] основан на использовании символьной проверки моделей с предварительно описанными в спецификации целями проверки и знаниями сторон-участников. Для описания криптографических протоколов и их свойств безопасности используется модульный формальный язык HLPSL [20] или более высокоуровневый язык спецификаций CAS+ [21]. В Avispa входит подсистема SPAN (Security Protocol ANimator), которая имеет транслятор для преобразования кода из языка CAS+ в HLPSL и позволяет визуализировать схемы взаимодействия сторон.

Язык спецификаций CAS+ является более простым в синтаксисе и позволяет быстро описать протокол. Пример описания части протокола на языке CAS+ приведен ниже:

```
1    protocol NeedhamSchroederPublicKey;
2    identifiers
3    A,B             : user;
4    Na,Nb           : number;
5    KPa,KPb : public_key;
6
7    messages
8    1. A -> B       : {Na, A}KPb
9    2. B -> A       : {Na, Nb}KPa
10   3. A -> B       : {Nb}KPb
11
12   knowledge
13   A       : A,B,KPa,KPb;
14   B       : A,B,KPa,KPb;
15   ...
```

В данном фрагменте описаны роли пользователей A, B, те данные, которые каждый пользователь знает (случайные числа `Na`, `Nb`, ключи `KPa`, `KPb`), передача сообщений (`messages`) и знания (`knowledge`) взаимодействующих сторон A,B.

Язык HLPSL – это язык, с которым работает непосредственно Avispa. Пример описания части протокола, которая описана выше, на языке HLPSL приведен ниже:

```
1     role Alice (A, B: agent,
2                 KPa, KPb: public_key,
3                 SND, RCV: channel (dy))
4         played_by A def=
5         transition
6
7         0.  State  = 0 /\ RCV(start) =|>
8             State':= 2 /\ Na' := new()
9                        /\ SND({Na'.A}_KPb)
10        2.  State  = 2 /\ RCV({Na.Nb'}_KPa) =|>
11            State':= 4 /\ SND({Nb'}_KPb)
12
13    end role
14    ...
```

В отличие от CAS+, где передаваемые сообщения и данные для всех ролей описываются в отдельной секции, в HLPSL они описываются для каждой роли отдельно. В данном фрагменте при описании ролей используются: обозначение ролей, участвующих во взаимодействии (строка 1), ключи шифрования (строка 2), канал передачи по модели Долева-Яо (строка 3), указывается какая роль «играется» из указанных (строка 4). Сообщения протокола в области `transition` (строки 5–11) имеют формат, привязанный к состояниям, например, в строках 7–8 описывается переход из состояния 0 в 2. При этом вначале получаются некоторые данные с помощью `RCV` (получить), затем переменной `Na'` присваивается новое значение и происходит отправка `Na'к B` с помощью `SND` (переслать). Далее протокол описывается аналогичным образом.

Синтаксис языка HLPSL более сложный и наилучшим способом описать протокол, особенно если он довольно объемный, является описание его на CAS+, а затем с помощью транслятора Avispa преобразовать его в HLPSL. Чем сложнее и больше протокол, тем более вероятно возникновение ошибок при преобразовании, так что после этого необходимо вручную поправить некоторые фрагменты в HLPSL. Так же не стоит описывать цели проверки на CAS+, а лучше дописать их непосредственно в HLPSL.

Для использования модуля OFMC (рис. 1) требуется дополнительная спецификация для всех данных в сообщении для проверки на секретность, а также области сообщений, где требуется провести проверку на аутентификацию сторон.

В результате проверки будет выдан соответствующий результат. В случае обнаружения атак, покажется тип атаки и её ход в виде соответствующих изменений сообщений стороной злоумышленником. Если атак не обнаружено, то в выводе программы будет соответствующее сообщение, что все в порядке. С помощью кнопки «Protocol simulation» можно посмотреть схему взаимодействия сторон в протоколе.

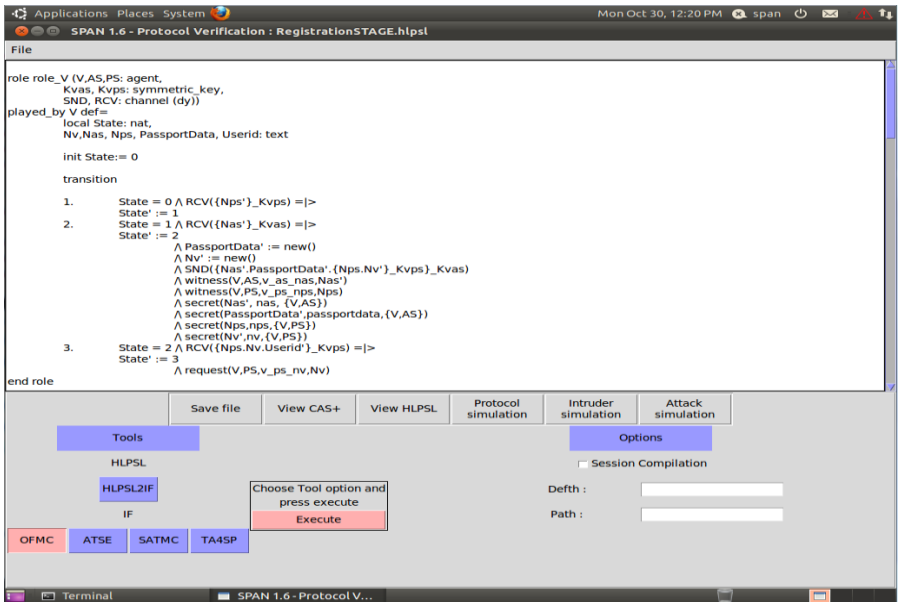

Рис. 1. Интерфейс программы

С помощью кнопки «Intruder simulation» (моделирование злоумышленника) можно посмотреть подобную схему, только с участием стороны злоумышленника, в которой отобразятся перехваченные им данные. С помощью кнопки «Attack simulation» можно увидеть схему атаки с участием злоумышленника, при условии, что в протоколе обнаружилась какая-либо атака.

Этапы проведения электронного голосования в контексте разработанной системы на основе слепых посредников:

1. *Подготовка.* На данном этапе создается база данных голосующих и бюллетень. Эти данные шифруются, и уполномоченные лица доставляют их на соответствующие компоненты-серверы системы.

2. *Регистрация.* На этом этапе пользователи регистрируются в системе, используя свои идентификационные данные, в настоящий момент это паспортные данные, и получают свой анонимный идентификатор.
3. *Голосование.* Пользователи получают бюллетень, делают свой выбор и отправляют его вместе со своим анонимным идентификатором. Если такой идентификатор есть, то голос принимается, а пользователю пересылается проверочный идентификатор, с помощью которого он может проверить свой голос после подсчета голосов.
4. *Подсчет результатов и проверка голосов*. На последнем этапе голоса подсчитываются, результаты публикуются в открытом доступе, и любой проголосовавший пользователь может проверить свой голос с помощью проверочного идентификатора.

Проверка протокола с помощью инструмента Avispa. С помощью Avispa в контексте разработанных протоколов возможны проверки: аутентификации сторон, секретности данных и защиты от replay-атак. Осуществить проверку целостности, в частности использующийся режим выработки CMAC (Cipher-based message authentication code, имитовставки) с помощью инструмента Avispa нельзя. Протокол не подразумевает использование временных меток в их классическом варианте отслеживания свежести сообщения. Вместо этого в разработанной системе используется временной контроль сессии сервером, в котором в случае если сессия длится слишком долго, то она разрывается.

В статье анализируется этап голосования. Моделируется три стороны: пользователь, сервер-посредник и основной сервер. Анализ протокола будет производиться после фазы выработки общего сессионного ключа между сторонами. При связи пользователя с сервером-посредником и основным сервером, серверы вначале отсылают свою часть секрета, подписанную своим закрытым ключом и свой сертификат. На данном этапе пользователь аутентифицирует серверы. Серверы при взаимодействии между собой также обмениваются частями секрета, подписанными своими закрытыми ключами**,** и сертификатами. Таким образом серверы аутентифицируют друг друга.

Протокол будет описан на языке CAS+ [3], затем с помощью транслятора Avispa переведен в HLPSL [2]. Проверка будет осуществляться с помощью модуля OFMC, где цели проверки – секретность передаваемых в сообщениях данных и аутентификация сторон.

Этап голосования. На рис. 2 изображена упрощенная схема этапа голосования.

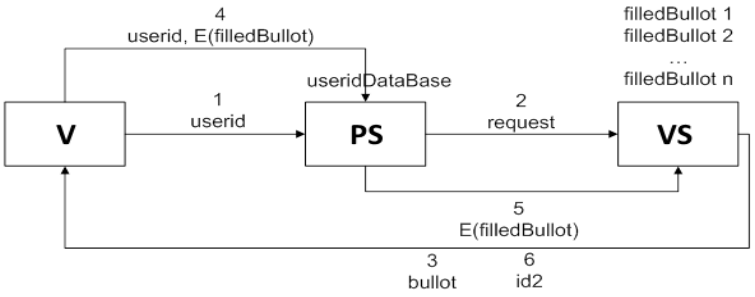

Рис. 2. Упрощенная схема этапа голосования

Описание протокола

Подготовка этапа

Осуществляется генерация общих секретных ключей V, VPS, VVS аналогично с помощью протокола выработки общего сессионного ключа. Стороны-серверы генерируют случайные числа и отправляют своему получателю сообщения (1), (2), (3).

Процесс голосования

V генерирует N_v,. Далее формирует сообщение (4) со своим уникальным идентификатором userid, шифрованными случайными числами на общем секретном ключе VVS, имитовставкой I_{4vvs}, все это шифруется на ключе VPS, высчитывается и прикладывается имитовставка I_{4vps}, после чего сообщение (4) отправляется PS. PS в данном случае является слепым посредником, что является основой всего протокола. Это означает, что PS может прочесть только первую часть сообщения с userid, а последующую не может. Он проверяет значение имитовставки, наличие userid в БД и в случае успеха перенаправляет другую часть в виде сообщения (5) VS. VS проверяет значение имитовставки, случайное число и в случае успеха отсылает V бюллетень в виде сообщения (6). Голосующий расшифровывает сообщение, проверяет значение имитовставки, случайных чисел, заполняет бюллетень, формирует сообщение (7) и отправляет PS. PS расшифровывает, проверяет значение имитовставки и значение userid ещё раз, в случае успеха перенапрявляет оставшуюся часть сообщения (8) VS. VS расшифровывает, проверяет значение имитовставки, случайное число и принимает голос. Далее генерирует уникальный идентификатор id2, формирует сообщение (9) и отправляет его пользователю. Пользователь расшифровывает, проверяет значение имитовставки, случайные числа и запоминает свой id2, по которому он может проверить свой голос. Второй идентификатор нужен для усиления анонимности голоса.

Протокол этапа голосования

Подготовка этапа:

ECDHE (V, PS) – vps – общий секретный ключ для обмена
PS: генерирует N_{ps}
(1) PS -> V: $E_{vps}(N_{ps})$
ECDHE (V, VS) – vvs – общий секретный ключ для обмена
VS: генерирует N_{vs}
(2) VS -> V: $E_{vvs}(N_{vs})$
ECDHE (PS, VS) – psvs – общий секретный ключ для обмена
VS: генерирует N_{psvs}
(3) VS -> PS: $E_{psvs}(N_{psvs})$

Процесс голосования:

V: генерирует N_v
(4) V -> PS: E_{vps} (N_{ps}, *userid,* $E_{vvs}(N_{vs}, N_v)$, I_{4vvs}), I_{4vps}
(5) PS -> VS: E_{psvs} (N_{psvs}, $E_{vvs}(N_{vs}, N_v)$, I_{4vvs})), I_{5psvs}
(6) VS -> V: $E_{vvs}(N_{vs}, N_v,$ *ballot)),* I_{6vvs}
(7) V -> PS: E_{vps} (N_{ps}, *userid,* E_{vvs} (N_{vs}, N_v, *filledBullot),* I_{7vvs}), I_{7vps}
(8) PS -> VS: E_{psvs} (N_{psvs}, E_{vvs} (N_{vs}, N_v, *filledBullot),* I_{7vvs}), I_{8psvs}
VS: запоминает голос
VS: генерирует id2
(9) VS -> V: $E_{vvs}(N_{vs}, N_v,$ *id2),* I_{9vvs}

Запись ECDHE означает использование протокола Диффи-Хеллмана на эллиптических кривых с использованием эфемерных ключей. В нашем случае используется доработанная версия ECDHE-RSA, где для аутентификации используется подпись с помощью шифра RSA и сертификат сервера, которая позволяет защититься от MIMT (man in the middle, атак человек по середине) атак. Описание протокола следующее.

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) Обе стороны на своей стороне вырабатывают общий сессионный ключ K для дальнейшего взаимодействия с помощью симметричного шифра.

Здесь V – клиент, S – доверенный сервер, у которого есть сертификат, DHs – часть секрета сервера, DHv – часть секрета клиента, $Sign_{SKs}$(DHs) – подпись с помощью закрытого ключа сервера SKs, Certificate – сертификат сервера.

При генерации общего сессионного ключа между серверами используется такой же протокол, за исключением того, что обе стороны обмениваются сертификатами и если последние действительны, то генерируется общий сессионный ключ.

Проверка данного протокола не является целесообразной в рамках нашей работы, поскольку это общепринятый стандарт и его надежность доказана. Проверка безопасности протокола голосования будет осуществляться после этого этапа.

Анализ безопасности протокола на этапе голосования с помощью инструмента Avispa. Рассмотрим описание протокола на языке CAS+ на этапе голосования.

```
protocol EVoting;
identifiers
V,PS,VS                                    : user;
Nps,Nvs,Npsvs,Nv,Userid,Id2,Ballot,FilledBallot     : number;
Kvps,Kvvs,Kpsvs                            : symmetric_key;

messages
1. VS -> V    : {Nvs}Kvvs
2. VS -> PS   : {Npsvs}Kpsvs
3. PS -> V    : {Nps}Kvps
4. V -> PS    : {Nps,Userid,{Nvs,Nv}Kvvs}Kvps
5. PS -> VS   : {Npsvs,{Nvs,Nv}Kvvs}Kpsvs
6. VS -> V    : {Nvs,Nv,Ballot}Kvvs
7. V -> PS    : {Nps,Userid,{Nvs,Nv,FilledBallot}Kvvs}Kvps
8. PS -> VS   : {Npsvs,{Nvs,Nv,FilledBallot}Kvvs}Kpsvs
9. VS -> V    : {Nvs,Nv,Id2}Kvvs

knowledge
V      : V,PS,VS,Nvs,Nv,Userid,Id2,Ballot,FilledBallot,Kvps,
Kvvs;
PS     : V,PS,VS,Nps,Npsvs,Kvps,Kpsvs,Userid;
VS     : V,PS,VS,Nvs,Nv,Npsvs,Kpsvs,Kvps,Id2,Ballot,
FilledBallot;

session_instances
 [V:v,PS:ps,VS:vs,Kvps:kvps,Kvvs:kvvs,Kpsvs:kpsvs]
 [V:v,PS:ps,VS:vs,Kvps:kvps,Kvvs:kvvs,Kpsvs:kpsvs];

intruder_knowledge
 v,ps,vs;

goal
 secrecy_of Nps [V,PS];
 secrecy_of Nvs [V,VS];
```

```
35    secrecy_of Npsvs [PS,VS];
36    secrecy_of Nv [V,VS];
37    secrecy_of Userid [V,PS];
38    secrecy_of Id2 [V,VS];
39    secrecy_of FilledBallot [V,VS];
40    secrecy_of Ballot [V,VS];
41    PS authenticates V on Nps;
42    VS authenticates PS on Npsvs;
43    VS authenticates V on Nvs;
44    V authenticates VS on Nv;
```

Описаны 3 взаимодействующие стороны с помощью ролей: V, PS, VS (строки 2–3). В секции `identifiers` описаны объекты, участвующие в протоколе: взаимодействующие стороны (строка 3), случайные числа для аутентификации, идентификаторы (строка 4). Указаны симметричные ключи, которые будут использоваться для шифрования сообщений (строка 5). В секции `messages` (строки 7–16) описана передача сообщений между ролями, какие данные передаются и на каком ключе шифруются. В секции `knowledge` (строки 18–23) описаны данные, которые известны той или иной роли во время выполнения протокола. В секции `session_instances` (строки 25–27) описаны сессии. Среди моделируемых сессий выделено 2, которые позволяют моделировать одновременное взаимодействие двух клиентов с системой. Это позволит обнаружить возможные атаки на аутентификацию сторон и replay-атаки. В секции `intruder_knowledge` (строки 29–30) указаны изначальные знания злоумышленника. В секции `goal` (строки 32–44) указаны секретность ключевых значений и аутентификация по схеме запрос-ответ с передачей между участниками случайных чисел. Для обеспечения секретности значения необходимо, чтобы эта переменная была зашифрована, и чтобы ключ шифрования не оказался у злоумышленника. Для того, чтобы одна сторона могла аутентифицировать другую с использованием механизма запрос-ответ, требуется, чтобы сторона, желающая аутентифицировать послала случайное число другой стороне, а эта другая сторона в ответном сообщении вернула это случайное число. В данном протоколе есть 4 таких действия:

1. PS аутентифицирует V по Nps
2. VS аутентифицирует PS по Npsvs
3. VS аутентифицирует V по Nvs
4. V аутентифицирует VS по Nv

В 1 и 3 случаях сервер аутентифицирует клиента. Напомним (см. пункт Процесс голосования), что клиент аутентифицировал сервер на подготовительном этапе при генерации сессионных ключей. Во 2 – ом случае происходит дополнительная аутентификация только стороны PS, поскольку именно сообщение от этой стороны для нас важно. Взаимная аутентификация обоих серверов произведена на этапе подготовки. В 4 – ом случае так же происходит дополнительная аутентификация стороны VS. Это важно для сообщения (9) где VS присылает проверочный идентификатор id2. Что касается replay-атак, то защита от них возможна благодаря наличию случайного числа в начале каждого сообщения, которое каждая сторона проверяет при получении сообщения. Результаты проверки с помощью модуля OFMC представлены на рис. 3.

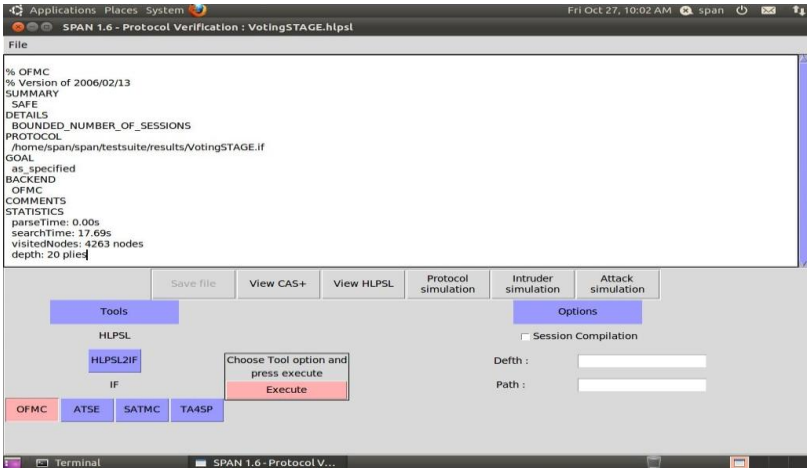

Рис. 3. Проверка протокола этапа голосования с помощью модуля OFMC

На рис. 4 представлена схема взаимодействия сторон на этапе голосования по шагам. На рис. 5 изображена схема взаимодействия при наличии злоумышленника (Intruder_ сторона, выделенная красным цветом). Такая схема является наглядной реализацией атаки человек посередине. При передаче сообщений в ходе выполнения осуществляется переход из области «Incoming events» в «Past events», причем формат представляет собой направление передачи сообщений (от кого и кому) и само сообщение. Как видно из результатов моделирования в области перехваченных данных «Intruder knowledge», все передаваемые сообщения зашифрованы на ключах, не известных злоумышленнику, что исключает возможность каким-либо образом узнать ключевую информацию, такую, как голос пользователя или его проверочный идентификатор. Запись «nonce-N» означает какие-то данные, недоступные для чтения.

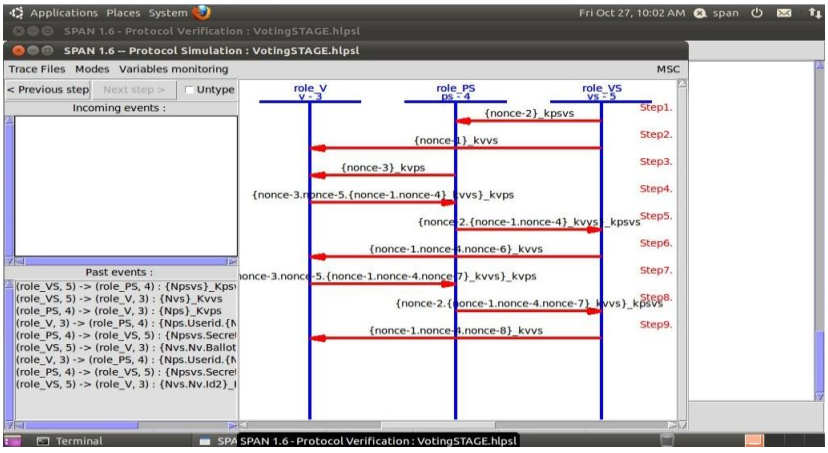

Рис. 4. Схема протокола голосования в режиме просмотра «Protocol simulation»

В результате анализа было выявлено, что протокол голосования является безопасным, обеспечивает выполнение целей (свойств) безопасности, поставленных при анализе протокола: обеспечение секретности данных, аутентификации сторон, защиту от replay-атак.

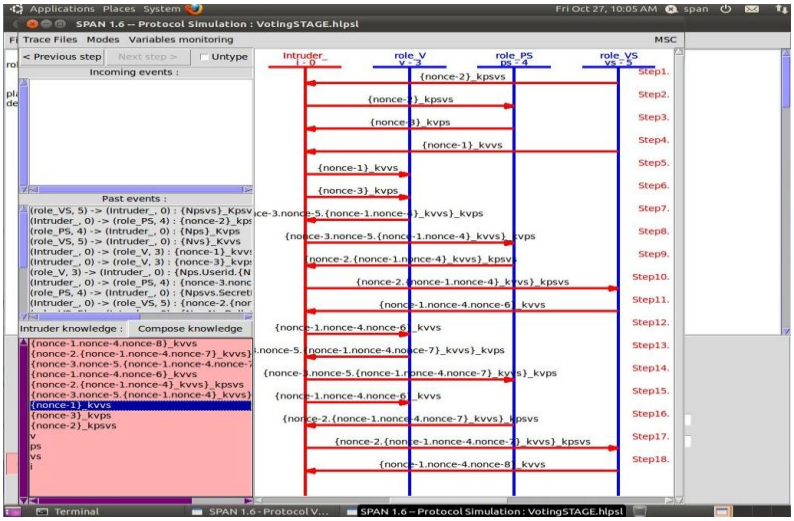

Рис. 5. Схема взаимодействия сторон в протоколе голосования при наличии злоумышленника в режиме просмотра «Intruder simulation»

Заключение. В работе рассмотрено применение инструмента автоматизированной верификации безопасности криптографических протоколов Avispa для анализа безопасности протокола голосования, разработанного авторами. Выполнено описание протокола на формальных языках CAS+ и HLPSL. Проанализированы свойства обеспечения безопасности передаваемых данных между взаимодействующими сторонами. Показано, что поставленные цели обеспечения безопасности: аутентификация сторон, проверка секретности данных, защита от replay-атак достигнуты. Рассмотрена схема взаимодействия сторон с помощью графического функционала используемого инструмента. Проведен анализ сообщений, которые может перехватить злоумышленник. На основе графического представления хода выполнения протокола при наличии предполагаемого злоумышленника выявлено, что он не сможет узнать передаваемые между сторонами данные, поскольку все сообщения зашифрованы на неизвестных ему ключах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

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. – 2015.
3. *Ben Adida.* Mixnets in Electronic Voting. – Cambridge University, 2005.
4. Electronic elections: fear of falsification of the results. – Kazakhstan today, 2004.
5. *Lipen V.Y., Voronetsky M.A.* Lipen DV technology and results of testing electronic voting systems. – United Institute of Informatics Problems NASB, 2002.
6. *David L Chaum.* Untraceable electronic mail, return addresses, and digital pseudonyms // Communications of the ACM. – 1981. – Vol. 24 (2). – P. 84-90.
7. *Ali S. T., Murray J.* An Overview of End-to-End Verifiable Voting Systems // arXiv preprint arXiv: 1605.08554. – 2016.
8. *Smart M., Ritter E.* True trustworthy elections: remote electronic voting using trusted computing // International Conference on Autonomic and Trusted Computing. – Springer Berlin Heidelberg, 2011. – S. 187-202.
9. *Bruck S., Jefferson D., Rivest R.L.* A modular voting architecture ("frog voting"). Toward strustworthy elections. – Springer Berlin Heidelberg, 2010.

10. *Jonker H., Mauw S., Pang J.* Privacy and verifiability in voting systems: Methods, developments and trends // Computer Science Review. – 2013.
11. *Shubhangi S. Shinde, Sonali Shukla, Prof. D.K. Chitre*. Secure E-voting Using Homomorphic Technology // International Journal of Emerging Technology and Advanced Engineering. – 2013.
12. *Neumann S., Volkamer M.* Civitas and the real world: problems and solutions from a practical point of view // Availability, Reliability and Security (ARES), 2012. Seventh International Conference on. – IEEE, 2012. – P. 180-185.
13. *Yi X., Okamoto E.* Practical remote end-to-end voting scheme // International Conference on Electronic Government and the Information Systems Perspective. – Springer Berlin Heidelberg, 2011. – S. 386-400.
14. *Hirt M., Sako K.* Efficient receipt-free voting based on homomorphic encryption // International Conference on the Theory and Applications of Cryptographic Techniques. – Springer Berlin Heidelberg, 2000. – P. 539-556.
15. *Rivest L. R. et al.* Lecture notes 15: Voting, homomorphic encryption. – 2002.
16. *Izabachene M.* A Homomorphic LWE Based E-voting Scheme // Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016.
17. *Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine J. Alex Halderman*. Security Analysis of the Estonian Internet Voting System, University of Michigan, Ann Arbor, MI, U.S.A. 2014.
18. *Basin D., M¨odersheim S., and Vigan`o L.* OFMC: A Symbolic Model-Checker for Security Protocols // International Journal of Information Security. – 2004.
19. *Бабенко Л.К., Писарев И.А., Макаревич О.Б.* Защищенное электронное голосование с использованием слепых посредников // Известия ЮФУ. Технические науки. – 2017. – № 5 (190). – С. 6-15.
20. The AVISPA team, The High Level Protocol Specification Language. – http://www.avispa-project.org/. – 2006.
21. *Ronan Saillard, Thomas Genet.* CAS+, March 21, 2011.

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*, 2015.
3. *Ben Adida.* Mixnets in Electronic Voting. Cambridge University, 2005.
4. Electronic elections: fear of falsification of the results. Kazakhstan today, 2004.
5. *Lipen V.Y., Voronetsky M.A.* Lipen DV technology and results of testing electronic voting systems. United Institute of Informatics Problems NASB, 2002.
6. *David L Chaum.* Untraceable electronic mail, return addresses, and digital pseudonyms, *Communications of the ACM*, 1981, Vol. 24 (2), pp. 84-90.
7. *Ali S. T., Murray J.* An Overview of End-to-End Verifiable Voting Systems, *arXiv preprint arXiv: 1605.08554*, 2016.
8. *Smart M., Ritter E.* True trustworthy elections: remote electronic voting using trusted computing, *International Conference on Autonomic and Trusted Computing*. Springer Berlin Heidelberg, 2011, pp. 187-202.
9. *Bruck S., Jefferson D., Rivest R.L.* A modular voting architecture ("frog voting"). Toward strustworthy elections. Springer Berlin Heidelberg, 2010.
10. *Jonker H., Mauw S., Pang J.* Privacy and verifiability in voting systems: Methods, developments and trends, *Computer Science Review*, 2013.
11. *Shubhangi S. Shinde, Sonali Shukla, Prof. D.K. Chitre*. Secure E-voting Using Homomorphic Technology, *International Journal of Emerging Technology and Advanced Engineering*, 2013.
12. *Neumann S., Volkamer M.* Civitas and the real world: problems and solutions from a practical point of view, *Availability, Reliability and Security (ARES), 2012. Seventh International Conference on*. IEEE, 2012, pp. 180-185.
13. *Yi X., Okamoto E.* Practical remote end-to-end voting scheme, *International Conference on Electronic Government and the Information Systems Perspective*. Springer Berlin Heidelberg, 2011, pp. 386-400.

14. *Hirt M., Sako K.* Efficient receipt-free voting based on homomorphic encryption, *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer Berlin Heidelberg, 2000, pp. 539-556.
15. *Rivest L. R. et al.* Lecture notes 15: Voting, homomorphic encryption. 2002.
16. *Izabachene M.* A Homomorphic LWE Based E-voting Scheme, *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016*.
17. *Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine J. Alex Halderman.* Security Analysis of the Estonian Internet Voting System, University of Michigan, Ann Arbor, MI, U.S.A. 2014.
18. *Basin D., M¨odersheim S., and Vigan`o L.* OFMC: A Symbolic Model-Checker for Security Protocols, *International Journal of Information Security*, 2004.
19. *Babenko L.K., Pisarev I.A., Makarevich O.B.* Zashchishchennoe elektronnoe golosovanie s ispol'zovaniem slepykh posrednikov [Protected electronic voting system with the use of blind intermediaries], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 5 (190), pp. 6-15.
20. The AVISPA team, The High Level Protocol Specification Language. Available at: http://www.avispa-project.org/. – 2006.
21. *Ronan Saillard, Thomas Genet.* CAS+, March 21, 2011.

Статью рекомендовал к опубликованию д.т.н., профессор И.А. Калмыков.

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: lkbabenko@sfedu.ru; 347928, г. Таганрог, ул. Чехова, 2, тел.: 88634361518; кафедра безопасности информационных технологий; профессор.

Писарев Илья Александрович – e-mail: ilua.pisar@gmail.com; г. Таганрог, ул. Котлостроительная, 7, кв. 35; тел.: 89885350837; кафедра безопасности информационных технологий; аспирант.

Babenko Liudmila Klimentevna – Southern Federal University; e-mail: lkbabenko@sfedu.ru; 2, Chehov street, Taganrog, 347928, Russia; phone: +78634361518; the department of security of information technologies; professor.

Pisarev Ilya Aleksandrovich – e-mail: ilua.pisar@gmail.com; 7, Kotlostroitelnaia street, Apt. 35, Taganrog, Russia; phone: +79885350837; the department of information technology security; postgraduate.