

АЛГОРИТМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ аутентификации абонентского доступа имитатора сети ПД учебного лабораторного стенда

Р. Бельфер, доцент кафедры "Информационная безопасность"
МГТУ им. Н.Э.Баумана, к.т.н. / a.belfer@yandex.ru

Е. Глинская, ст. преподаватель кафедры "Информационная безопасность"
МГТУ им. Н.Э. Баумана,

А.Кравцов, с.н.с. НИИЦ ЦНИИ ВВКО

УДК 621.392, DOI: 10.22184/2070-8963.2018.70.1.64.69

Предложены алгоритмы программного обеспечения взаимной аутентификации на абонентском доступе и аутентификации только оконечного пункта. Статья дополняет изложенный в предыдущей работе алгоритм ПО установления коммутируемого виртуального канала на абонентском доступе имитатора сети ПД с учетом обеспечения информационной безопасности.

ВВЕДЕНИЕ

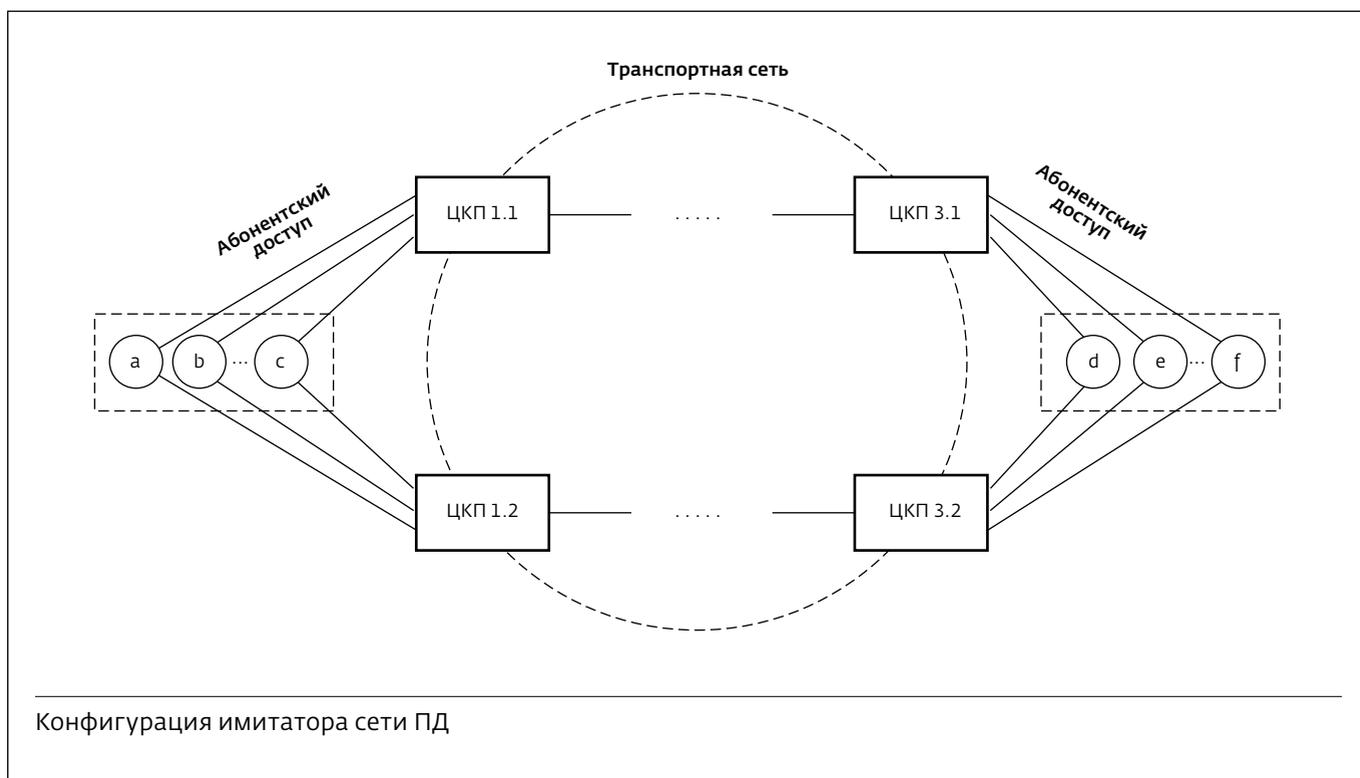
В работе [1] приводится алгоритм программного обеспечения, выполняющий функцию установления соединения на участках абонентского доступа имитатора сети ПД. При этом отмечается, что такая сеть разрабатывается на кафедре "Информационная безопасность" МГТУ им. Н.Э.Баумана в научно-образовательном плане в рамках создания учебного лабораторного стенда (УЛС). Отмечена актуальность УЛС в целях подготовки кадров для создания отечественных сетей ПД категории специального назначения с высокими требованиями по надежности, информационной безопасности и другим характеристикам. Показано, что это относится, в частности, к сетям для ОПК.

В указанной работе отмечается необходимость разработать алгоритм ПО в части выполнения функций взаимной аутентификации оконечной станции и узла коммутации на абонентских участках сети. Настоящая статья посвящена этой разработке.

Приведем конфигурацию имитатора сети, используемую в работе [1].

КОНФИГУРАЦИЯ ИМИТАТОРА СЕТИ

На рисунке приведена конфигурация имитатора сети с адресацией оконечных пунктов и центров коммутации пакетов (ЦКП). Физические адреса: оконечных пунктов – 101 (а), 102 (b), 103 (с) и ЦКП абонентских доступов этих оконечных пунктов – 11 (ЦКП 1.1), 21 (ЦКП 2.1); оконечных



пунктов – 601 (f), 602 (d), 603 (e) и ЦКП абонентских доступов этих оконечных пунктов – 31 (ЦКП 3.1), 32 (ЦКП 3.2).

Описание алгоритма ПО в [1] приводится на примере установления коммутируемого виртуального канала между оконечными пунктами a и f. Алгоритм ПО аутентификации в настоящей работе приводится на абонентских доступах с этими оконечными пунктами. Оконечный пункт a является источником установления соединения с оконечным пунктом f. Будем называть f оконечным пунктом назначения.

ПО включает диспетчер программ и описание алгоритма шести программ. Диспетчер

программ DISP управляет последовательным выполнением программ P_1, P_2, \dots, P_6 .

Программа P_1 отвечает за установление адресов оконечных пунктов и ЦКП сети ПД. **Программа P_2** – за создание имитаторов сертификатов на абонентском доступе оконечного пункта источника. Для этого создадим имитаторы сертификатов оконечного пункта a и ЦКП 1.1. Примем одноуровневую структуру удостоверяющих центров. Для упрощения изложения не рассматриваем аутентификацию с ЦКП 1.2 и в открытой части сертификатов оставим только один параметр – открытый ключ. Обозначим ЦКП 1.1 через I, оконечный пункт a через R. ЦКП 1.1 и a

получили сертификат в общем удостоверяющем центре (УЦО). Введем значения открытого и закрытого ключа, а также параметра расчета принятого метода RSA для: общего удостоверяющего центра УЦО (соответственно $P_{УЦО} = 7$, $S_{УЦО} = 23$, $n = 187$); удостоверяющего центра ЦКП 1.1 ($P_1 = 3$, $S_1 = 7$, $n = 31$); удостоверяющего центра оконечного пункта а ($P_R = 5$, $S_R = 77$, $n = 119$). Имитатор сертификата ЦКП 1.1, полученный в общем удостоверяющем центре, - УЦО $\ll I \gg = P_1 S_{УЦО} (h(P_1))$. Имитатор сертификата оконечного пункта а, полученный в общем удостоверяющем центре, - УЦО $\ll R \gg = P_R S_{УЦО} (h(P_R))$.

Программа P₃ предназначена для взаимной аутентификации оконечной станции источника виртуального соединения и ЦКП абонентского доступа, а также создания головного ключа. Она посылает сообщения.

Сообщение 1. ЦКП 1.1. посылает в оконечный пункт а (I в R) сообщения:

- сертификат I, полученный в УЦО (УЦО $\ll I \gg = P_1 S_{УЦО} (h(P_1))$). На приеме этого сообщения в оконечном пункте а убеждаемся в достоверности открытого ключа P_1 . Для расчетов функции хэширования используется библиотека по адресу <https://www.cryptopp.com>, включающая несколько стандартных протоколов. Для упрощения изложения поставленной в настоящей работе задачи описания алгоритма программного обеспечения в каждом случае задаем численное значение хэша открытого ключа. Примем $h(P_1) = 88$, не производя хэширования стандартным алгоритмом. Тогда $S_{УЦО} (h(P_1)) = 88^{23} \text{ mod } 187 = 11$. Считаем, что открытый ключ удостоверяющего центра ($P_{УЦО}$), выдавшего сертификат оконечному пункту, достоверно известен. Сравниваем принятое значение $h(P_1) = 88$ с полученным значением $h(P_1)$ с помощью открытого ключа $P_{УЦО} = 7$, т.е. $P_{УЦО} S_{УЦО} (h(P_1)) = (88^{23} \text{ mod } 187)^7 \text{ mod } 187 = 88$. Совпадение этих двух значений $h(P_1)$ показывает достоверность открытого ключа P_1 , т.е. открытого ключа ЦКП 1.1;

- случайное число R_1 , сгенерированное I для защиты от угрозы "повтор аутентификации".

Сообщение 2. Оконечный пункт а посылает в ЦКП 1.1 (R в I) сообщения:

- сертификат R, полученный в УЦО (УЦО $\ll R \gg = P_R S_{УЦО} (h(P_R))$). На приеме в оконечном пункте а убеждаемся в достоверности открытого ключа P_R . Примем, что $h(P_R) = 75$, не производя хэширования стандартным алгоритмом. Тогда $S_{УЦО} (h(P_R)) =$

$75^{23} \text{ mod } 187 = 80$. Считаем, что открытый ключ удостоверяющего центра ($P_{УЦО}$), выдавшего сертификат ЦКП 1.1, достоверно известен. Сравниваем принятое значение $h(P_R) = 75$ с полученным значением $h(P_R)$ с помощью открытого ключа $P_{УЦО} = 7$, т.е. $P_{УЦО} S_{УЦО} (h(P_R)) = (75^{23} \text{ mod } 187)^7 \text{ mod } 187 = 75$. Совпадение этих двух значений $h(P_R)$ показывает достоверность открытого ключа P_R , т.е. открытого ключа оконечного пункта а;

- сгенерированное случайное число R_R (для защиты от угрозы "повтор аутентификации");
- $P_1 [K_{и}]$ - головной секретный ключ $K_{и}$, зашифрованный открытым ключом I (т.е. P_1). $K_{и}$ - случайное число, сгенерированное R. Примем $K_{и} = 37$, $P_1 [K_{и}] = 37^3 \text{ (mod } 31) = 30$;
- $S_R [h(R_1, R_R, K_{и})]$ - хэш-функция ($R_1, R_R, K_{и}$), зашифрованная закрытым ключом получателя S_R .

На приеме в ЦКП 1.1 (I) производится: дешифрация головного секретного ключа $K_{и}$ закрытым ключом источника S_1 . $K_{и} = 30^7 \text{ (mod } 31) = 37$; с помощью открытого ключа получателя P_R проверка целостности значений $R_1, R_R, K_{и}$. Примем $h(R_1, R_R, K_{и}) = 108$, не производя хэширования.

Тогда $S_R [h(R_1, R_R, K_{и})] = 108^{77} \text{ (mod } 119) = 61$. На приеме в I с помощью открытого ключа получателя $P_R = 5$ производится проверка целостности значений $R_1, R_R, K_{и}$. $S_R [h(R_1, R_R, K_{и})] = h(R_1, R_R, K_{и}) = 61^5 \text{ (mod } 119) = 108$. Совпадение значений $h(R_1, R_R, K_{и})$ показывает целостность значений $R_1, R_R, K_{и}$. Проверка, является ли значение R_1 тем же самым, которое было отправлено в сообщении 1 (защита от угрозы "повтор аутентификации"). Успешная проверка принятых сообщений завершает аутентификацию оконечного пункта.

Сообщение 3. ЦКП 1.1. посылает в оконечный пункт а (I в R) сообщение:

- $S_1 [h(R_R, K_{и})]$ - хэш-функцию ($R_R, K_{и}$), зашифрованную закрытым ключом источника S_1 . На приеме в R с помощью открытого ключа приемника P_R производится проверка целостности значений $R_R, K_{и}$. Примем $h(R_R, K_{и}) = 29$. $S_1 [h(R_R, K_{и})] = 29^7 \text{ (mod } 31) = 27$. На приеме в R с помощью открытого ключа приемника P_R производится проверка целостности значений $R_R, K_{и}$ - $P_1 S_1 [h(R_R, K_{и})] = h(R_R, K_{и}) = 27^3 \text{ (mod } 31) = 29$. Совпадение значений $h(R_R, K_{и})$ показывает целостность значений $R_1, R_R, K_{и}$. Кроме того, на приеме в R производится проверка, является ли значение R_R тем же самым, которое было отправлено ранее в

ЦКП 1.1. (защита от угрозы "повтор аутентификации"). Успешная проверка принятого сообщения завершает аутентификацию источника I (ЦКП 1.1).

Программа P₄ отвечает за установление ассоциации безопасности на абонентском доступе оконечного пункта назначения. Программа P₅ - за создание имитаторов сертификатов на абонентском доступе оконечного пункта назначения. Для этого создадим имитаторы сертификатов оконечного пункта и ЦКП 3.1. В отличие от принятой структуры для абонентского доступа оконечного пункта а и ЦКП 1.1. примем двухуровневую структуру удостоверяющих центров. Для упрощения изложения не рассматриваем аутентификацию с ЦКП 3.2. Обозначим ЦКП 3.1 через I, оконечный пункт f через R. ЦКП 3.1 получил сертификат в удостоверяющем центре I (УЦИ). Оконечный пункт f получил сертификат в удостоверяющем центре R (УЦР).

Введем значения открытого, закрытого ключа и параметра расчета принятого метода RSA для: общего удостоверяющего центра УЦО (соответственно $P_{УЦО} = 7$, $S_{УЦО} = 23$, $n = 187$); имитатора сертификата УЦИ, полученного в общем удостоверяющем центре УЦО ($P_{УЦИ} = 17$, $S_{УЦИ} = 53$, $n = 77$); имитатора сертификата УЦР, полученного в общем удостоверяющем центре УЦО ($P_{УЦР} = 3$, $S_{УЦР} = 67$, $n = 55$); имитатора сертификата ЦКП 3.1 (I), полученного в удостоверяющем центре УЦИ (соответственно $P_I = 3$, $S_I = 7$, $n = 31$); имитатора сертификата оконечного пункта f, полученного в удостоверяющем центре УЦР ($P_R = 5$, $S_R = 77$, $n = 119$).

Для упрощения изложения в открытой части сертификатов оставим только один параметр - открытый ключ. Имитатор сертификата ЦКП 3.1, полученный в удостоверяющем центре УЦИ, - УЦИ << I >> = $P_I S_{УЦИ}(h(P_I))$; имитатор сертификата оконечного

пункта f, полученный в УЦР, - УЦР << R >> = $P_R S_{УЦР}(h(P_R))$; имитатор сертификата удостоверяющего центра УЦИ, выданный в общем удостоверяющем центре УЦО, - УЦО << УЦИ >> = $P_{УЦИ} S_{УЦО}(h(P_{УЦИ}))$; имитатор сертификата удостоверяющего центра УЦР, выданный в общем удостоверяющем центре УЦО, - УЦО << УЦР >> = $P_{УЦР} S_{УЦО}(h(P_{УЦР}))$.

Программа P₆ предназначена для взаимной аутентификации оконечного пункта назначения виртуального соединения и ЦКП абонентского доступа, а также создания головного ключа. Она также посылает сообщения.

Сообщение 1. ЦКП 3.1. в адрес оконечного пункта f (I в R) посылает:

- сертификат удостоверяющего центра УЦИ, полученный в общем удостоверяющем центре УЦО (УЦО << УЦИ >> = $P_{УЦИ} S_{УЦО}(h(P_{УЦИ}))$). На приеме в оконечном пункте f убеждаемся в достоверности открытого ключа удостоверяющего центра УЦИ ($P_{УЦИ}$). Примем $h(P_{УЦИ}) = 72$, не производя хэширования стандартным алгоритмом. Тогда $S_{УЦО}(h(P_{УЦИ})) = 72^{23} \pmod{187} = 183$. Считаем, что открытый ключ удостоверяющего центра ($P_{УЦО}$), выдавшего сертификат оконечному пункту, достоверно известен. Сравниваем принятое значение $h(P_{УЦИ}) = 183$ с полученным значением $h(P_I)$ с помощью открытого ключа $P_{УЦО} = 7$, т.е. $P_{УЦО} S_{УЦО}(h(P_I)) = 72^{23} \pmod{187}^7 \pmod{187} = 72$. Совпадение обоих значений $h(P_{УЦИ}) = 72$ означает подлинность открытого ключа удостоверяющего центра $P_{УЦИ}$, как принято для приема $P_{УЦИ} = 17$.
- сертификат ЦКП 3.1 (I), полученный в удостоверяющем центре УЦИ (УЦИ << I >> = $P_I S_{УЦИ}(h(P_I))$). На приеме в оконечном пункте f убеждаемся в достоверности открытого ключа P_I ЦКП 3.1 (I). Примем $h(P_{УЦИ}) = 29$, не производя хэширования стандартным алгоритмом. Тогда $S_{УЦИ}$

$(h(P_I)) = 29^7 \pmod{31} = 27$. Сравниваем принятое значение $h(P_{УЦП}) = 29$ с полученным значением $h(P_{УЦП}) = 29$ с помощью открытого ключа $P_{УЦП} = 17$, т.е. $P_{УЦП} S_{УЦП}(h(P_I)) = 29^{53} \pmod{31}^{17} \pmod{31} = 29$. Совпадение обоих значений $h(P_{УЦП}) = 29$ означает подлинность открытого ключа – убеждаемся в достоверности открытого ключа ЦКП 3.1 (I), как принято для примера $P_I = 3$.

- R_I – случайное число, сгенерированное I для защиты от угрозы "повтор аутентификации".

Описание алгоритма взаимной аутентификации и формирования головного ключа далее следует без примеров расчета, как это приводилось выше на обоих абонентских доступах. Это объясняется тем, что принцип работы этих алгоритмов такой же.

Сообщение 2. Оконечный пункт f в адрес ЦКП 3.1: (R в I) посылает:

- сертификат удостоверяющего центра УЦР, полученный в общем удостоверяющем центре УЦО (имитатор сертификата удостоверяющего центра УЦР, выданный в общем удостоверяющем центре УЦО, – УЦО \ll УЦР $\gg = P_{УЦР} S_{УЦО}(h(P_{УЦР}))$), а на приеме в конечном пункте f убеждаемся в достоверности открытого ключа удостоверяющего центра УЦР ($P_{УЦР} = 3$);
- сертификат окончного пункта f , полученный в удостоверяющем центре УЦР ($УЦР \ll R \gg = P_R S_{УЦР}(h(P_R))$), а на приеме в конечном пункте f убеждаемся в достоверности открытого ключа P_R окончного пункта f ;
- случайное число R_R , сгенерированное R для защиты от угрозы "повтор аутентификации".

Далее алгоритм взаимной аутентификации и формирования головного ключа аналогичен алгоритму, изложенному в программе P_3 для абонентского доступа окончного пункта a (источника соединения коммутируемого виртуального канала). Отличаются только подлежащие в примере аутентификации окончный пункт (f), ЦКП (ЦКП 3.1) абонентского доступа и головной секретный ключ K_n .

ОСОБЕННОСТИ АЛГОРИТМА АУТЕНТИФИКАЦИИ

За основу приведенного алгоритма принят стандартизированный протокол Европейского института стандартизации ETSI ETS 300 841 мультимедийной сети ISDN [2]. Он был реализован, в частности, на сети связи общего пользования немецкого оператора Deutsche Telecom.

Приведем основные изменения предложенного в настоящей работе алгоритма.

Во-первых, изложен не алгоритм аутентификации, а алгоритм программного обеспечения реализации алгоритма аутентификации с использованием имитаторов сертификатов.

Во-вторых, предложена не только двухуровневая иерархия сертификатов, но и одноуровневая.

В-третьих, в настоящем разделе рассмотрена аутентификация только окончного пункта. Для некоторых областей использования сетей ПД специального назначения может отсутствовать необходимость при каждом установлении соединения производить аутентификацию узла коммутации транспортной части сети. Даже в системе сигнализации ОКС №7, используемой в сетях связи общего пользования стандартов ISDN, IN, UMTS, механизм аутентификации MTPSec используется только один раз при пуске сети в эксплуатацию. В остальных случаях используется созданный им головной ключ [3,4].

Основное изменение в приведенной программе взаимной аутентификации P_3 для выполнения аутентификации только окончного пункта – не используется сообщение 3.

Выводы

Предложенные алгоритмы ПО взаимной аутентификации на абонентском доступе и аутентификации только окончного пункта могут быть использованы при реализации имитатора сети ПД в рамках учебного лабораторного стенда и лабораторных работ студентов.

ЛИТЕРАТУРА

1. Басараб М.А., Бельфер Р.А., Глинская Е.В., Кравцов А.В.. Алгоритм ПО установления коммутируемого виртуального канала на абонентском доступе имитатора сети ПД с учетом обеспечения информационной безопасности // ПЕРВАЯ МИЛЯ. 2017. №8. С.64–69.
2. Бельфер Р.А. Сети и системы связи (технологии, безопасность) /Учеб. пособие по дисциплине "Сети и системы связи": электронное учеб. изд. – М.: МГТУ им. Н.Э.Баумана. 2012. 723 с.
3. Sengar H., Wijesekera D., Jajodia S. Authentication and Integrity in telecommunication Signaling Network. Engineering of Computer-Based Systems, 12th IEEE International Conference and Workshops, 2005. P. 163-170.
4. Yang Y., He W., Feng S. Security Analysis and Amendment of 3G Core Network Based on MTPsec, IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008. P. 519-523.