УДК 681.3.06

ТЕХНОЛОГИЯ ПЕРЕДАЧИ ДАННЫХ В ЗАЩИЩЕННОМ РЕЖИМЕ

Алексей Остапченко
НПЦ "Битис"

Аннотация: **Сформулировано предложение по передаче данных в TCP/IP сетях в защищенном режиме с использованием протокола защищенной передачи.**
Summary: **Secure data transfer in TCP/IP net with the Secure transfer protocol usage formulate.**
Ключевые слова: **Информация, защита информации.**

I Введение

Организация передачи данных в защищенном режиме в IP сети осуществляется *программными* и (или) *аппаратными* средствами при помощи набора функций криптографических преобразований. Функции выполняют действия по идентификации сторон передачи данных (клиента и сервера), генерации случайного ключа сессии (или пары ключей) для сплошного шифрования данных, шифрования данных симметричными и несимметричными методами, выработку цифровой подписи для подтверждения целостности передаваемых данных и некоторые другие.

Программные средства, которые передают и (или) принимают данные в защищенном режиме, подразделяются на два класса – прикладные и непосредственные.

– **Прикладные**. Средства, выполняющие свою прикладную задачу в IP сети. Для сокрытия передаваемых данных и взаимной идентификации сторон, участвующих в процессе (клиента и сервера), используются криптографические преобразования и протоколы защищенной передачи данных (например, интернет браузер обменивается данными с Веб сервером в защищенном режиме по протоколу SSL).

– **Непосредственные**. Это средства, выполняющие задачу сокрытия передаваемых данных и взаимную идентификацию сторон обмена. Выполняют перехват незащищенных данных от клиента или сервера, взаимную идентификацию сторон передачи данных (клиента и сервера), шифрование и контроль целостности.

Непосредственные – это средства, основная задача которых – обеспечить аутентификацию, шифрование и целостность передаваемых данных в IP сети при взаимодействии с пользовательскими прикладными программами – клиентами и серверами (например, клиент электронной почты, интернет браузер, почтовый сервер и другие). Эти программные средства являются межсетевыми криптографическими экранами, которые технологически выполняются как *криптографический файервол* (firewall) или как *криптографический сервер, использующий прокси технологию*.

Криптографический файервол – это программное средство, которое зависит от особенности реализации конкретной операционной системы (OS dependent). На транспортном уровне осуществляется перехват, криптографическое преобразование, замена содержимого с добавлением цифровой подписи и зашифрованного разового ключа IP пакета, следующего от источника к приемнику данных. На приемнике выполняется анализ и восстановление в исходное состояние каждого полученного IP пакета.

Криптографический сервер, использующий прокси технологию – это программное средство, которое выступает как посредник между клиентом и сервером, который получает данные через TCP/IP соединение от пользовательских прикладных программ – клиентов, выполняет действия по транспортировке данных к серверу от имени клиента и обратно, а также криптографические преобразования над данными. *Криптографические сервера* работают на уровне приложений и технологически могут быть выполнены мультиплатформными.

Криптографический сервер использует для передачи данных в защищенном режиме:

– маршрутизацию IP пакетов, над которыми выполнено криптографическое преобразование и (или) трансляцию адресов (address translation);

– протокол защищенной передачи данных.

Набор общеизвестных протоколов защищенной передачи данных, таких как SSL, TLS, PCT, осуществляет криптографические преобразования над данными и маршрутизацию IP пакетов между клиентом и сервером. Криптографические преобразования осуществляются на основе алгоритмов DES, MD5 и других. Протоколы защищенной передачи данных не предполагают работу с алгоритмами на основе отечественных ГОСТов. Кроме этого, существуют экспортные ограничения на длину ключей шифрования для программных средств, использующих криптографические преобразования.

Для использования криптографических преобразований, основанных на отечественных ГОСТах, и длины ключей, свободных от экспортных ограничений, предлагается *криптографический сервер* с организацией таблиц маршрутизации IP пакетов и протокол защищенной передачи данных, который предполагает любую длину ключей шифрования и любые алгоритмы криптографических преобразований.

Ниже разработана схема организации таблиц маршрутизации IP пакетов и правила трансляции адресов для криптографического сервера. Разработан принцип определения типа клиентского соединения с криптографическим сервером и протокол защищенной передачи данных между криптографическими серверами.

II Передача данных между клиентом и сервером

Технологическое расположение *криптографического сервера*, как посредника в передаче данных между клиентом и сервером, представлено на рис. 1.

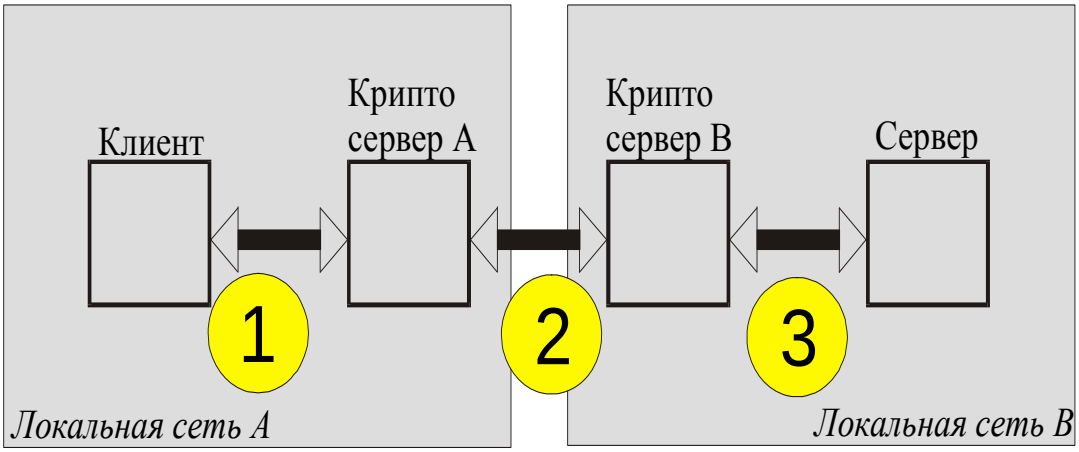

Рисунок 1 – Передача данных между клиентом и сервером при использовании посредника – криптографического сервера

Данные, передаваемые между клиентом и сервером, проходят через три TCP/IP соединения, отмеченные на рис. 1 цифрами 1, 2, 3.

1 – Соединение, устанавливаемое клиентом с *криптографическим сервером А*.

2 – Соединение, устанавливаемое между двумя *криптографическими серверами А и В*. Данные передаются в защищенном режиме с взаимной аутентификацией клиента и сервера.

3 – Соединение, устанавливаемое *криптографическим сервером В* с сервером, к которому обращается клиент.

Соединения 1, 3 относятся к соединениям, которые устанавливаются с внутренними сервисами (internal services) *криптографических серверов*, клиентских серверов и являются незащищенными. Соединения 3 устанавливаются с внешними сервисами (external services) *криптографических серверов* и являются защищенными.

Соединение 1. Основная задача заключается в определении типа клиентского соединения и удаленного адреса для последующей передачи данных.

Криптографический сервер открывает соединения с клиентами и выступает как посредник для обслуживания соединений следующих типов:

– *Сокс (Socks)*. Соединения клиентов с *криптографическим сервером* по протоколу сокс (socks), который содержит адрес сервера, с которым необходимо связаться для обмена данными между клиентом и сервером.

– *Прокси (Proxy)*. Соединения клиентов с криптографическим сервером; в заголовках коммуникационного протокола содержится адрес сервера, с которым необходимо связаться для обмена данными между клиентом и сервером.

– *Передача (Transfer)*. Соединения клиентов с криптографическим сервером; не содержится адрес сервера, к которому обращается клиент. Удаленный ресурс для клиента представляется в виде *криптографического сервера*.

При установлении *сокс* соединения клиента с *криптографическим сервером* устанавливается адрес сервера, с которым необходимо связаться для обмена данными между клиентом и сервером. На основе

адреса сервера, типа и порта соединения из таблицы удаленных сервисов (Remote Services Table), извлекается правило и адрес для передачи данных.

При установлении *прокси* соединения клиента с *криптографическим сервером* устанавливается адрес сервера, с которым необходимо связаться для обмена данными между клиентом и сервером (определяется из строки в заголовке запроса Host : <адрес сервера> или CONNECT <адрес сервера>). На основе адреса сервера, типа и порта соединения из таблицы удаленных сервисов (Remote Services Table) извлекается правило и адрес для передачи данных (подобный тип клиентского соединения устанавливается по протоколу HTTP и др.).

При установлении соединения клиента типа *передача* с *криптографическим сервером* по номеру порта соединения из таблицы удаленных сервисов (Remote Services Table) извлекается правило и адрес для передачи данных (подобный тип клиентского соединения устанавливается по протоколу POP3, SMTP и др.).

После установки соединения клиента с внутренним сервисом *криптографического сервера* определяется адрес внешнего сервиса удаленного *криптографического сервера*, который является локальным для клиентского сервера. Выполняется взаимная аутентификация *криптографических серверов* и шифрование всего трафика обмена данными. Соединение закрывается по инициативе клиента или сервера.

Соединение 2. Основная задача заключается в восстановлении исходных данных и определении типа клиентского соединения с *криптографическим сервером*.

Криптографический сервер открывает соединения с удаленными *криптографическими серверами* и выступает как посредник для обслуживания соединений клиентов с серверами.

Криптографический сервер открывает соединения типа *Защищенный (Secure)* по протоколу защищенной передачи данных (Secure Transfer Protocol), который содержит тип клиентского соединения, определенный при соединении 1.

Данные поступают в защищенном режиме через открытые внешние сервисы *криптографического сервера*.

Соединение 3. Основная задача заключается в открытии соединения с сервером клиента для последующей передачи данных в открытом виде. Из Соединения 2 определяется тип клиентского соединения (из заголовков протокола защищенной передачи данных) и адрес клиентского сервера из таблицы локальных сервисов (Local Services Table) и эти данные используются для установки Соединения 3. Соединение закрывается по инициативе клиента или сервера.

III Маршрутизация IP пакетов

Для маршрутизации IP пакетов и открытия соединений с клиентами и серверами *криптографические сервера* используют таблицы сервисов.

- Таблица сервисных серверов (Service Servers Table). Назначение – открытие соединений клиентов (как к внутренним сервисам) и удаленных *криптографических серверов* (как к внешним сервисам) с *криптографическим сервером*. Внутренние сервисы определяются локальным именем или IP адресом сервера в локальной сети, а внешние сервисы определяются внешним, по отношению к локальной сети, именем или IP адресом. Используется для обслуживания первого типа соединений.
- Таблица локальных сервисов (Local Services Table). Назначение – маршрутизация IP пакетов, следующих от *криптографического сервера* к клиентскому серверу. Используется для обслуживания второго типа соединений.
- Таблица удаленных сервисов (Remote Services Table). Назначение – для маршрутизации IP пакетов, следующих от клиента через *криптографический сервер* к удаленному *криптографическому серверу*. Используется для обслуживания третьего типа соединений.

Поля таблиц сервисов:

ServiceID – идентификатор сервиса или порядковый номер в таблице сервиса;

ServiceName – имя сервиса, является ключевым идентификатором для передачи данных по адресу сетевого ресурса, связывает поля из разных таблиц сервисов;

ServiceType – тип сервиса и обслуживания протокола соединения (*Socks, Proxy, Transfer, Secure*);

HostAddress – адрес сервера сервиса;

DestHostAddress – адрес удаленного сервера, к которому адресуются данные (нет в таблице Service Servers Table);

ServiceServerType – тип сервера сервиса (*internal, external*) (только для таблицы Service Servers Table);

ServerSocket – служебные данные (только для таблицы Service Servers Table).

Таблица сервисных серверов определяет по номеру порта имя сервиса и тип клиентского соединения и из таблиц локальных и удаленных сервисов определяется адрес сервера, которому будут переданы данные. При

открытии *криптографическим сервером* соединения типа Secure удаленным *криптографическим сервером* передается имя сервиса, по которому определяется тип клиентского соединения.

Таким образом обслуживается любое клиентское соединение по практически любому из существующих протоколов.

Адреса DestHostAddress, HostAddress в таблицах маршрутизации представляются тремя типами: EMPTY (пустой), ANY (любой), ADDRESS (адрес хоста).

Правила маршрутизации определяются (в зависимости от типов DestHostAddress, HostAddress) как:

- ADDRESS_TO_ADDRESS (передавать адрес по адресу);
- ADDRESS_TO_ANY (передавать адрес);
- ANY_TO_ANY (передавать любой адрес);
- EMPTY_TO_ANY = ANY_TO_ANY;
- ANY_TO_ADDRESS (все потоки данных передавать по адресу);
- EMPTY_TO_ADDRESS = ANY_TO_ADDRESS;
- ANY_TO_EMPTY (блокировать все потоки данных для всех адресов);
- ADDRESS_TO_EMPTY (блокировать потоки данных для адреса).

Правила маршрутизации IP пакетов и открытия соединений с клиентами и серверами представлены ниже:

ServiceType	**DestHostAddress**	**HostAddress**
Transfer, Secure, Socks, Proxy	EMPTY	ADDRESS
Transfer, Secure, Socks, Proxy	EMPTY	ADDRESS
Socks, Proxy	ANY	ANY
Socks, Proxy	ANY	ADDRESS
Socks, Proxy	ADDRESS	ANY
Socks, Proxy	ADDRESS	ADDRESS
Socks, Proxy	ADDRESS или ANY	EMPTY

- Для сервисов типа *Transfer* указывается адрес в поле HostAddress для перенаправления потока данных по этому адресу.
- Для сервисов типа *Secure* указывается адрес HostAddress, с которым клиенты и удаленные *криптографические сервера* открывают соединения с *криптографическим сервером*.
- Для сервисов типа *Socks, Proxy* допускается трансляция адресов для перенаправления потока данных в зависимости от содержимого поля DestHostAddress и HostAddress.
- Допускается использование ключевого слова *any* для обозначения любого имени или IP адреса в заголовках протокола обмена данными.
- Допускается блокирование адресов указанием адреса в DestHostAddress или ключевого слова *any* и пустого поля HostAddress для блокировки одного или всех адресов для данного имени сервиса.
- Трансляция адресов означает, что для адреса DestHostAddress данные определенного типа сервиса будут переданы по адресу HostAddress.

Примеры таблиц маршрутизации IP пакетов при помощи таблиц сервисов приведены ниже.

Таблица 1 – Таблица сервисных серверов

ServiceID	**ServiceName**	**ServiceType**	**HostAddress**	**ServiceServerType**	**ServerSocket**
0	STP	Secure	192.117.67.29:10440	External	Null
1	HTTP	Proxy	10.2.1.100:3128	Internal	Null
2	SMTP	Transfer	127.0.0.1:25	Internal	Null

ServiceID – идентификатор сервиса
ServiceName – имя сервиса
ServiceType – тип сервиса и обслуживания протокола соединения (*Socks, Proxy, Transfer, Secure*)
HostAddress – адрес сервера сервиса
ServiceServerType – тип сервера сервиса (*internal, external*)
ServerSocket – служебные данные

Таблица 2 – Таблица локальных сервисов

ServiceID	**ServiceName**	**ServiceType**	**DestHostAddress**	**HostAddress**
1	HTTP	Proxy	yahoo.com:80	10.2.1.100:3128
2	HTTP	Proxy	*Any*	10.1.1.10:3128
3	HTTP	Proxy	ukr.net:80	*any*
4	HTTP	Proxy		*any*
5	HTTP	Proxy	local.net:80	
6	SMTP	Transfer		10.1.1.10:25

ServiceID – идентификатор сервиса
ServiceName – имя сервиса
ServiceType – тип сервиса и обслуживания протокола соединения
DestHostAddress – адрес удаленного сервера к которому адресуются данные
HostAddress – адрес локального сетевого ресурса

Таблица 3 – Таблица удаленных сервисов

ServiceID	**ServiceName**	**ServiceType**	**DestHostAddress**	**HostAddress**
1	HTTP	Proxy	yahoo.com:80	192.117.67.29:10440
2	HTTP	Proxy	*any*	192.117.67.30:10440
3	HTTP	Proxy	ukr.net:80	*any*
5	SMTP	Transfer		192.117.67.29:10440

ServiceID – идентификатор сервиса
ServiceName – имя сервиса
ServiceType – тип сервиса и обслуживания протокола соединения
DestHostAddress – адрес удаленного сервера к которому адресуются данные
HostAddress – адрес удаленного сетевого ресурса

IV Протокол защищенной передачи данных

Протокол защищенной передачи данных технологически разделяется на две составляющие:

– протокол аутентификации, получения сертификата и ключа сессии для сплошного шифрования данных (**Authentication and Keys Exchange**), см. табл. 4;

– протокол защищенной передачи блоков данных (**Secure Data Transfer Protocol**), см. табл. 5.

Таблица 4 – Команды протокола аутентификации, получения сертификата и ключа сессии для сплошного шифрования данных

Формат запроса	**Формат ответа**
Получить сертификат от сертификационного центра по его номеру ID **GET_SERTIFICATE** <CRLF> SertificateID: <ID> <CRLF> <CRLF>	Сертификат не найден 201 **SERTIFICATE_NOT_FOUND** <CRLF> <CRLF>
	Сертификат блокирован 210 **SERTIFICATE_IS_BLOCKED** <CRLF> <CRLF>
	Сертификат найден 200 **OK** <CRLF> ContentLength: <длина в байтах> <CRLF> <CRLF> <Содержимое сертификата>
Получение номера сертификата ID от удаленного владельца **GET_SERTIFICATE_ID** <CRLF> <CRLF>	Сертификат не найден 201 **SERTIFICATE_NOT_FOUND** <CRLF> <CRLF>

Продолжение таблицы 4

	Сертификат найден 200 **OK** <CRLF> SertificateID: <ID> <CRLF> <CRLF>
Получение даты и времени создания сертификата удаленного владельца от сертификационного центра по его номеру ID **GET_SERTIFICATE_DATETIME** <CRLF> SertificateID: <ID> <CRLF> <CRLF>	Сертификат не найден 201 **SERTIFICATE_NOT_FOUND** <CRLF> <CRLF>
	Сертификат найден 200 **OK** <CRLF> Datetime: <Время в мсек, прошедшее с 1 января 1970 г.> <CRLF> <CRLF>
Получение номера ID сертификата, ключа сессии, синхропосылки и времени их генерации от удаленного сервера по номеру сертификата ID **GET_SESSION_PARAM** <CRLF> SertificateID: <ID> <CRLF> <CRLF>	Параметры не сформированы 202 **SESSION_PARAM_NOT_FOUND** <CRLF> <CRLF>
	Необходимо воспользоваться параметрами сформированными ранее 200 **OK** <CRLF> Parameters: old <CRLF> <CRLF>
	Новые параметры сформированы 200 **OK** <CRLF> Parameters: new <CRLF> SertificateID: <ID> <CRLF> Datetime: < Время в мсек, прошедшее с 1 января 1970 г.> <CRLF> ContentLength: <Длина Буфера параметров в байтах> <CRLF> SignLength: <Длина цифровой подписи в байтах> <CRLF> <CRLF> <Сформированные параметры>
Получение номера ID сертификата, ключа сессии, синхропосылки и времени их генерации от удаленного сервера по номеру сертификата ID с аутентификацией **GET_SESSION_PARAM** <CRLF> SertificateID: <ID> <CRLF> AuthenticationLength: <Длина Буфера аутентификации в байтах> <CRLF> AuthenticationSignLength: <Длина цифровой подписи в байтах> <CRLF> * Methods: <методы криптографических преобразований> <CRLF> <CRLF> <Буфер аутентификации>	Параметры не сформированы 202 **SESSION_PARAM_NOT_FOUND** <CRLF> <CRLF>

Продолжение таблицы 4

	Необходимо воспользоваться параметрами сформированными ранее 200 **OK** <CRLF> Parameters: old <CRLF> <CRLF>
	Ошибки аутентификации 221 **AUTHENTICATION_FAILED** <CRLF> <CRLF>
	Новые параметры сформированы 200 **OK** <CRLF> Parameters: new <CRLF> SertificateID: <ID> <CRLF> Datetime: < Время в мсек, прошедшее с 1 января 1970 г.> <CRLF> ContentLength: <Длина Буфера параметров в байтах> <CRLF> SignLength: <Длина цифровой подписи в байтах> <CRLF> AuthenticationLength: <Длина Буфера аутентификации в байтах> <CRLF> AuthenticationSignLength: <Длина цифровой подписи в байтах> <CRLF> * Methods: <методы криптографических преобразований> <CRLF> <CRLF> <Буфер аутентификации> <Сформированные параметры>
Получение даты и времени создания параметров сессии от удаленного сервера по номеру сертификата ID **GET_SESSION_PARAM_DATETIME** <CRLF> SertificateID: <ID> <CRLF> <CRLF>	Параметры не сформированы 202 **SESSION_PARAM_NOT_FOUND** <CRLF> <CRLF>
	Параметры сформированы 200 **OK** <CRLF> Datetime: <Время в мсек, прошедшее с 1 января 1970 г.> <CRLF> <CRLF>

Таблица 5 – Команды протокола защищенной передачи блоков данных

Формат запроса	**Формат ответа**
Передавать блоки данных **TRANSFER_DATA** <CRLF> Service: <Имя сервиса> <CRLF> Host: <Адрес сервера> <CRLF> <CRLF> <Передаваемые данные>	
Передавать блоки данных с аутентификацией **TRANSFER_DATA** <CRLF> Service: <Имя сервиса> <CRLF> Host: <Адрес сервера> <CRLF> AuthenticationLength: <Длина Буфера аутентификации в байтах> <CRLF> AuthenticationSignLength: <Длина цифровой подписи в байтах> <CRLF> * Methods: <методы криптографических преобразований> <CRLF> <CRLF> <Буфер аутентификации> <Передаваемые данные>	При возникновении ошибки аутентификации 221 **AUTHENTICATION_FAILED** возникает разрыв соединения без уведомления

Команды протокола защищенной передачи данных допускают взаимодействие двух сторон как с взаимной аутентификацией, так и без нее. Размер блока параметров сессии и буфера аутентификации устанавливается равным 40 байт по умолчанию.

* Параметр Methods (методы криптографических преобразований) указывает список криптографических преобразований, которые выполнены над данными. По умолчанию используются государственные стандарты.

Управляющая команда **TRANSFER_DATA** указывает имя сервиса, которому предназначены данные, а также адрес сервера, с которого эти данные поступают. За заголовком следуют защищенные блоки данных.

Блоки передаваемых данных имеют следующий формат:

DataLength [длина 4 байта] – длина блока передаваемых данных;

Data [длина, указанная в DataLength] – блок данных;

SignLength [длина 4 байта] – длина цифровой подписи;

Sign [длина, указанная в SignLength] – цифровая подпись.

Формат выполнения протокола защищенной передачи данных с взаимной аутентификацией представлен в табл. 6.

Таблица 6 – Формат выполнения протокола защищенной передачи данных с взаимной аутентификацией

Криптографический сервер А	**Данные**	**Криптографический сервер В**
		Генерация случайного блока данных В. Получение сертификата А (командой GET_SERTIFICATE). Шифрование блока данных В сертификатом А.
	 Пересылка запроса GET_SESSION_PARAM с зашифрованным блоком данных В.	
Расшифровка блока данных В. Генерация случайного блока данных А (параметры сессии). Получение сертификата В (командой GET_SERTIFICATE). Шифрование блока данных А сертификатом В. Шифрование блока данных В сертификатом В.		
	 Пересылка ответа на запрос GET_SESSION_PARAM с зашифрованным блоком данных А и В.	
		Расшифровка блоков данных А и В. Проверка блока данных В (аутентификация А). Шифрование блока данных А сертификатом А.
	 Пересылка запроса TRANSFER_DATA с зашифрованным блоком данных А.	

Продолжение таблицы 6

Расшифровка блока данных A. Проверка блока данных A (аутентификация B).		
	 Пересылка данных с указанием имени сервиса.	

В табл. 6 представлен также порядок работы криптографических серверов A и B при обращении сервера B в локальную область A, закрытую сервером A. При повторном обращении сервера B к серверу A сервер A принимает решение о генерации новых параметров сессии и взаимной аутентификации, а также о временном интервале доверия к сертификату.

V Заключение

Протоколы передачи данных в защищенном режиме SSL, TLS, PCT и другие в общем случае предназначены для использования западных стандартов на криптографические преобразования над данными. Кроме того, реализации этих протоколов содержат экспортные ограничения на длину ключей, а поэтому не могут быть использованы для защиты передаваемых данных на уровне конфиденциальной связи. Кроме того, протоколы SSL, TLS, PCT необходимы для интеграции различных программных продуктов, зачастую выполненных различными производителями. Предложенный Протокол защищенной передачи данных предназначен для использования любой длины ключей, базируется на использовании отечественных стандартов криптографии и предназначен для использования в однородных *криптографических серверах*.

УДК 681.3.06:006.354

ИССЛЕДОВАНИЕ СВОЙСТВ ПОДСТАНОВОК ГОСТ 28147-89, ПОСТРОЕННЫХ НА ОСНОВЕ АНАЛИЗА СВОЙСТВ КООРДИНАТНЫХ ФУНКЦИЙ

Роман Олейников, Ирина Лисицкая, Александр Шумов
Харьковский национальный университет радиоэлектроники

Аннотация: **Показано, что известные требования к подстановкам ГОСТ 28147-89 позволяют использовать заполнение узлов замены, при которых шифр является уязвимым к дифференциальному криптоанализу.**
Summary: **It is shown, that known requirements to S boxes of GOST 28147-89 allow to use permutations, which make cipher vulnerable to differential cryptanalysis.**
Ключевые слова: **ГОСТ 28147-89, заполнение узлов замены, дифференциальный криптоанализ.**

В настоящее время в Украине официально рекомендован к применению только один блочный симметричный алгоритм шифрования – ГОСТ 28147-89. Его особенностью является отсутствие стандартного заполнения узлов замены (S-блоков). Разработчики шифра не определили долговременный ключ, также как и не опубликовали в открытой печати правила его построения. Тем не менее, известно, что именно свойства подстановок в значительной мере определяют криптостойкость всего шифра [1].

К нынешнему моменту опубликованы несколько подходов к обоснованию требований отбора подстановок ГОСТ 28147-89, один из которых основан на анализе криптографических свойств координатных (булевых) функций, образующих S-блок [2]. В работе предлагается использование четырёх требований, первое из которых определяет формирование перестановки, а три остальных определяют свойства булевых функций, отображающих входные биты S-блока в выходные. Кратко напомним эти требования:

1) все преобразования π_i ($i = \overline{1,8}$), образующие блок замены π, должны быть перестановками;

2) у каждой из координатных функций $f_j^{(i)}$ перестановок π_i ($i = \overline{1,8}$, $j = \overline{1,4}$) должен отсутствовать аффинный статистический аналог, совпадающий с ней с вероятностью большей, чем $3/4$: