Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Дисциплина «Основы защиты информации»

Отчёт по практическому занятию №5

Студент: Коренчук А.В.

ФИТ 2 курс 2 группа

Преподаватель: Ржеутская Н.В.

Минск 2022 г.

**Практическое занятие №5**

**«Криптографическая защита информации»**

Цель: изучить и закрепить умение реализации ЭЦП на примере RSA.

**Теоретическое введение**

Протоколы ЭЦП с одной стороны относят к протоколам аутентификации, т.к. гарантируют, что сообщение поступило от достоверного отправителя, а с другой стороны к протоколам контроля целостности, т.к. гарантируют, что сообщение пришло в неискаженном виде. Более того, получатель в дальнейшем может использовать ЭЦП как доказательство достоверности сообщения третьим лицам (арбитру) в том случае, если отправитель впоследствии попытается отказаться от него.

Говоря о схеме цифровой подписи, обычно имеют в виду следующую классическую ситуацию:

* отправитель знает содержание сообщения, которое он подписывает;
* получатель, зная открытый ключ проверки подписи, может проверить правильность подписи полученного сообщения в любое время без какого-либо разрешения и участия отправителя;
* безопасность схемы подписи гарантируется.

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (Федеральный закон "Об электронной цифровой подписи").

# **Этапы генерации и проверки ЭЦП**

*Этап 1.* Выработка ключей (выполняет отправитель А) - см. практическую работу 6  "Шифрование методом RSA".

*Этап 2.* Отправка сообщения и электронной подписи (выполняет отправитель А).

Отправка сообщения и ЭЦП на базе алгоритма RSA

*Таблица 7.1* – Ключи для алгоритма RSA



*Этап 3*. Получение сообщения и проверка электронной подписи (выполняет получатель B).

Получение сообщения и проверка ЭЦП на базе алгоритма RSA

*Таблица 7.2* – Ключи для алгоритма RSA



# **Контрольные вопросы**

1. Дайте определение понятию "электронная цифровая подпись".

**Электронная цифровая подпись** – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (Федеральный закон "Об электронной цифровой подписи").

1. Опишите последовательность действий участников протокола при отправке и проверке ЭЦП.

При создании цифровой подписи по классической схеме отправитель:

* применяет к исходному сообщению T хеш-функцию h(T) и получает хеш-образ r сообщения;
* вычисляет цифровую подпись s по хеш-образу r с использованием своего закрытого ключа;
* посылает сообщение T вместе с цифровой подписью s получателю.

Получатель, отделив цифровую подпись от сообщения, выполняет следующие действия:

* применяет к полученному сообщению T хеш-функцию h(T)и получает хеш-образ r сообщения;
* расшифровывает хеш-образ r’ из цифровой подписи s с использованием открытого ключа отправителя;
* проверяет соответствие хеш-образов r и r’ и если они совпадают, то отправитель действительно является тем, за кого себя выдает, и сообщение при передаче не подверглось искажению.

Как видно из этой схемы, порядок использования ключей обратный тому, который используется при передаче секретных сообщений. Вначале отправитель использует свой закрытый ключ, а затем получатель применяет открытый ключ отправителя.

1. Какой порядок использования ключей (открытый; закрытый) при отправке и проверке ЭЦП?

Закрытый, открытый.

1. Опишите схему протокола ЭЦП на основе алгоритма RSA.

Пусть открытый текст – m.Используя секретный ключ {d, n} создадим цифровую подпись s по формуле:



**Вывод:** в ходе работы был изучен способ **реализации ЭЦП на примере RSA.**