Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа по дисциплине «Защита информации»

Тема Алгоритм шифрования RSA и алгоритм хеширования MD5

Студент Светличная А.А.

Группа ИУ7-73Б

Преподаватель Чиж И. С.

Москва — 2023 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

RSA (Rivest-Shamir-Adleman) — это асимметричный криптографический алгоритм, разработанный Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом. Он использует пару ключей: публичный и приватный. Публичный ключ используется для шифрования данных, а приватный — для их расшифровки. RSA широко применяется для обеспечения конфиденциальности и цифровой подписи в коммуникациях и безопасности данных.

MD5 (Message Digest Algorithm 5) представляет собой однонаправленную хеш-функцию, создающую фиксированный 128-битный хеш-код из входных данных произвольной длины. Разработанный Рональдом Ривестом, MD5 широко использовался для проверки целостности данных. Однако, из-за уязвимостей к коллизиям (возможность создания различных входных данных с одинаковым хешем), MD5 стал устаревшим для криптографических целей.

Цель: разработать программную реализацию алгоритма шифрования RSA и алгоритма хеширования MD5.

Задачи:

- исследование общих аспектов алгоритмов;
- анализ алгоритмов RSA и MD5;
- программная реализация алгоритмов.

1 Аналитическая часть

Историческая справка

RSA — криптографический алгоритм, который был представлен в 1977 году Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом. Назван по первым буквам фамилий его создателей. RSA стал первым практически применяемым методом асимметричного шифрования, использующим пару ключей: публичный для шифрования и приватный для расшифровки.

Алгоритм получил широкое распространение благодаря своей эффективности и математической сложности задачи факторизации больших простых чисел, на которой он основан. RSA применяется для обеспечения безопасности в сферах электронной коммерции, цифровой подписи, аутентификации и других областях, где требуется шифрование и защита информации.

Алгоритм **MD** (Message Digest) был впервые представлен Рональдом Ривестом в 1991 году. Этот криптографический хеш-алгоритм создавал фиксированный хеш-код из входных данных произвольной длины. Оригинальная версия MD стала широко использоваться, но со временем её уязвимости к атакам стали очевидными, и были предложены более безопасные версии, такие как MD2, MD4 и, в конечном итоге, MD5.

MD5 был представлен Рональдом Ривестом в 1991 году в качестве улучшения оригинального алгоритма MD. MD5 создает 128-битный хеш-код и быстро стал популярным для проверки целостности данных и хранения хешей паролей. Однако, в 2004 году были обнаружены серьезные уязвимости, в результате чего MD5 был признан небезопасным для криптографического использования. С тех пор рекомендуется использовать более стойкие хеш-алгоритмы, такие как SHA-256 или SHA-3.

2 Конструкторская часть

2.1 Алгоритм шифрования RSA

Генерация ключей. Для использования RSA нужно сгенерировать два ключа — открытый (public) и закрытый (private). Сначала выбираются два простых числа p и q. Затем вычисляется $n = p \times q$. n является второй частью обоих ключей. Далее вычисляется $\phi(n) = (p-1) \times (q-1)$. После этого нужно найти число e, которое является взаимно простым с $\phi(n)$. Число e должно быть таким, чтобы выполнялось неравенство $1 < e < \phi(n)$. Это первая часть открытого ключа. Число d можно найти из формулы $d \times e \ mod \ \phi(n) = 1$. Таким образом, открытый ключ — пара (e, n), а закрытый — (d, n).

Замечание: шифровать с помощью ключа можно только такие значения, которые меньше, чем n.

Шифрование — $c = m^e \ mod \ n$.

Расшифрование — $m = c^d \ mod \ n$.

2.2 Алгоритм хеширования MD5

Алгоритм MD5 использует 4 многократно повторяющиеся преобразования над тремя 32-битными величинами X, Y и Z.

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z). \tag{2.1}$$

$$G(X, Y, Z) = (X \wedge Z) \vee (\neg Z \wedge Y). \tag{2.2}$$

$$H(X, Y, Z) = X \oplus Y \oplus Z. \tag{2.3}$$

$$I(X, Y, Z) = Y \oplus (\neg Z \vee X). \tag{2.4}$$

2.3 Электронная цифровая подпись

Электронная цифровая подпись (ЭЦП) позволяет подтвердить авторство электронного документа. Подпись связана как с автором, так и с самим документом с помощью криптографических методов и не может быть подде-

лана с помощью обычного копирования.

ЭЦП — это реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

С помощью RSA алгоритма можно не только шифровать данные, но и создавать электронную подпись. Разница в том, что при создании ЭП шифрование происходит с помощью закрытого ключа отправителя, а расшифрование — с помощью открытого ключа отправителя.

3 Технологическая часть

3.1 Тестирование

1. Пустой текст

```
Prime number: 8DACF414EA5DEA9F5ED463FFF0A61FFD9BE4764C
E549468555788ED949E15569
Prime number: B2F308968E82492F59DB8631DFFF1D5490FA55A6
07C831516B7A4F619CA9B61B
E: 940197D4F8A7068DF30C935027A3806BBD4296B8F03953A4EB0
E558600D04749
N: 6308C1A2F4A57994DA47A610A4B8ABC2802407D712F812836AC
50FCF168AAFBE328192FEC1ECC8210EA001B308FBC16480C110E0A
BD6CFE68E8365075BCDA813
D: 10A6C94AF48C645DC1B1BD5E22E57E0CE6FDA33DDFBD37A23EF
1F42B11AADF58E810C5D77FE7B9E046F6B4EA0C48A33925F6A06BF
3E81DBA903EDEB52C9A5B19
MD5 hash before encrypting:
d41d8cd98f00b204e9800998ecf8427e
MD5 hash after encrypting:
2635011C8601EB5F50CAF9C13768829A56305F112581A850917FB3
81FDFEF3C12255DA2294199361AEA2ABFED8BCF4326D862080C443
B1E744679C56D30AED27
MD5 hash after decrypting:
d41d8cd98f00b204e9800998ecf8427e
```

2. Текст длинной менее 64 байт: 01234

```
Prime number: D0FAC19F95411468AC84DF4E6CB7E54832F9F17
878401E8311B23C774D85C81D
Prime number: 8734C134CE73D3AFBFCDDBA584540FAE2AB8D75
C0C5D3EA8694619A16F27A573
E: D56B667F4C12FD70515D0F8109EE3C9EFFA3FBE759501AC644
5F51A477B40E15
N: 6E5F4CC2A3909C2D44369D67BE307F153D46C57B8F2B383821
E1639719B94E4F3648273F31359F05C7877DCD67E9562F266173C
```

```
9285241F2336F8EF21C7E9607
D: 8AD981D9F9BEDC4CA44AB0C18F4DBEE1F598C396138C20E412
B7AB3E02D88B121EF312468555716E1934C2B7DB980DA7D347430
B463D2A9A9B12C778CDD6975
MD5 hash before encrypting:
4100c4d44da9177247e44a5fc1546778
MD5 hash after encrypting:
51FDDB576D1C841FE71907E5486071E6AC1A87EF0B3F4A8DB04A
947C5A761C7FDDFC6EF35D94281C9C7AA2F17A1F2F7973BAB267
D883DD11B03591DD7F8BF65F
MD5 hash after decrypting:
4100c4d44da9177247e44a5fc1546778
```

3. Текст длинной более 64 байт: 01234567890123456789

```
Prime number: A3F02FF84F1055844CAC707073549422ACD27F2
44A8B07B74EAC20E64279BD67
Prime number: C880AEABD3A2E53DFB73EC8F49B22278DF37874
DC8BD8A3FD48F6B53E068EB07
E: C82FB9CA7A1FC73DC764D7D20F6915FEAF8A929B9004F5E059
FCB8B837EF2E37
N: 80660D6D437E63E9988DE0AE4AE373F6DC1795BFF4349E8DFF
9879C6D8747334E39E3454922E518BAB838F46127FD25EAFEF06B
0FD860B9588C941E5A409BAD1
D: 3D4424521BACE0A7D358448276879179 47BC92D2BC0C01FBF40
826DEF1FE72F6C7651A1EF2B1516A366957BC1BE9B46226F70FC38
46E466B357C5382C17581E3
MD5 hash before encrypting:
be497c2168e374f414a351c49379c01a
MD5 hash after encrypting:
3772D956BDD2116465C9DBBDF7BF6759F7B191EEC15CE9590CD
F1D7355BDECBE44126571F9D8AAD8576FED8A031F5DA894FF2B
8FD4166DC876E3E620B827DFE5
MD5 hash after decrypting:
be497c2168e374f414a351c49379c01a
```

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы была достигнута поставленная **цель**: разработана программная реализации шифровального алгоритма RSA и алгоритма хеширования MD5.

Все **задачи** лабораторной работы выполнены:

– исследованы общие аспекты алгоритмов;

– проведен анализ алгоритмов RSA и MD5;

– программно реализованы данные алгоритмы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ