

NanoMess

- [Этапы работы](#)

Этапы работы

Пакеты имеют вид:

* Шифрованный: `encrypt#L3eNIEq3LmbHEtC7Wml3uRQQq`

- `encrypt` - показатель, что сообщение зашифровано ключом сервера (см.)
- далее пакет имеет то же самое, что и нешифрованный пакет

* Нешифрованный (после регистрации):

`none#sender#2025_07_01_20_00#8t9SFvRgcV6jOldEbWyq6LCMg#someuser#o0H2FoSOmhG`

- `sender` - ник отправителя
- `2025_07_01_20_00` - время подписи
- `8t9SFvRgcV6jOldEbWyq6LCMg` - строка с датой подписанная ассиметричным ключом пользователя
- `someuser` - ник получателя
- `o0H2FoSOmhG` - зашифрованное сообщение

* При регистрации: `reg#nick#rk4TNhaThMoRUPZEj6vJHY3FZ`

- `nick` - ник
- `rk4TNhaThMoRUPZEj6vJHY3FZ` - ассиметричный публичный ключ пользователя

Ключ сервера - один из пары ключей, которые заранее сгенерированы и поставляются вместе с сервером и клиентом (для защиты от MITM)

1. Регистрация в системе

1. Пакет шифруется ассиметричным ключом сервера (см. шифр.пакет)
2. Передача публичного ключа (для подписи) и ника серверу (см. пакет при регистрации)

3. Принятие данных сервером, сохранение в
4. Сервер отвечает "accepted"
2. Вычисление общего симметричного ключа по Диффи-Хеллману
 1. Передача данных шифруется ассиметричным ключом сервера (см. шифр.пакет), пока не будет вычислен общий симметричный ключ клиент-клиент.
 2. Отправка запроса на генерацию общего ключа.
 3. *алгоритм Диффи-Хеллмана*
3. Передача сообщения
 1. Шифрование сообщения (см. нешифрованный пакет)
 2. Отправка на сервер и принятие его.
 3. Ожидание N времени пока получатель не спросит: "нет ли чего для меня?"
 1. Если не спросил - удалить пакет из RAM.
 2. Если спросил - передать пакет в таком же виде и занести в RAM сообщение для отправителя, что пакет принят.