

# Алгоритмы шифрования в OpenSSL



# Цель

01

Описать основные  
типы шифрования в  
OpenSSL

02

Проанализировать  
основные  
преимущества и  
недостатки  
алгоритмов

03

На основе  
полученных  
результатов  
предложить  
рекомендованный  
алгоритм

# Типы шифрования

Симметричное

Асимметричное

# Асимметричное шифрование

Асимметричное шифрование предполагает использование открытого и закрытого ключа

Для асимметричного шифрования в OpenSSL есть два вида алгоритма

1. RSA
2. ECDH

## Алгоритм Ривест-Шамир-Адлеман (RSA)

---

RSA относится к асимметричным алгоритмам шифрования.

---

Где открытый ключ используется для шифрования данных, а закрытый для их дешифрования.

# Алгоритм работы RSA

Алиса хочет отправить сообщение Бобу

Боб создает два ключа - открытый и закрытый

Боб отправляет открытый ключ Алисы

Алиса шифрует сообщение открытым ключом Боба

Алиса отправляет зашифрованное сообщение Бобу

Боб закрытым ключом расшифровывает сообщение

# Основной недостаток использование алгоритма **RSA** для шифрования сообщения

В связи с тем что алгоритм RSA является медленным (по сравнению с симметричными методами шифрования). Данный алгоритм не используется для шифрования сообщений.

# Протокол Диффи–Хеллмана на эллиптических кривых (ECDH)

ECDH позволяет двум сторонам, имеющим пары открытого и закрытого ключа на эллиптических кривых, получить общий секретный ключ, используя незащищенный от прослушивания канал связи.





# Алгоритм ECDH

Алиса и Боб хотят создать общий секретный ключ

Алиса и Боб создают пару открытого/закрытого ключа

Алиса и Боб обмениваются открытыми ключами

Алиса вычисляет общий секрет за счет открытого ключа  
Боба

Боб вычисляет общий секрет за счет открытого ключа  
Алисы

# Недостаток ECDH

- ECDH используется только для создания общего секрета (ключа), который потом будет является секретным ключом для симметричного шифрования.

# Типы ключей для асимметричного шифрования

Эфемерный

Статичный

# Статичный ключ

Статичный ключ - это такая пара ключей в которой Алиса и Боб используют одни и те же ключи для каждой сессии связи.

# Эфемерный ключ

Эфемерный ключ - это ключ который использует разные закрытые ключи для каждого сеанса.

# Преимущество эфемерного ключа в отличие от статического

Основное преимущество эфемерного ключа заключается в том что если третья сторона сможет найти секретный ключ за время одной сессии, он не сможет использовать его для других.

# Симметричное шифрование

Симметричное шифрование - это способ шифрования/дешифрования при котором применяется один и тот же криптографический ключ.

Существует два вида симметричного шифрования

- 1. Блочный
- 2. Поточковый

# Блочные шифры в OpenSSL



AES

DES

3DES

RC2

RC5

CAST

IDEA

Camellia





# AES

- AES (*Advanced Encryption Standard*) симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES.

# DES и 3DES

- DES - алгоритм для симметричного шифрования, разработанный фирмой IBM и утверждённый правительством США в 1977 году как официальный стандарт. Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 циклами и ключом, имеющим длину 56 бит. Алгоритм использует комбинацию нелинейных и линейных преобразований.
- 3DES - симметричный блочный шифр, созданный Уитфилдом Диффи, Мартином Хеллманом и Уолтом Тачманном в 1978 году на основе алгоритма DES с целью устранения главного недостатка последнего — малой длины ключа (56 бит), который может быть взломан методом полного перебора ключа. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше.

# RC2 и RC5

- RC2 – блочный шифр с длиной блока 64 бита и переменной длиной ключа, разработанный Роном Ривестом в конце 1980-х годов. Алгоритм является более быстрым, чем алгоритм DES.
- RC5 – это блочный шифр, разработанный Роном Ривестом из компании RSA Security с переменным количеством раундов, длиной блока и длиной ключа. Это расширяет сферу использования и упрощает переход на более сильный вариант алгоритма.

# CAST

- CAST-128 (или CAST5) в криптографии – блочный алгоритм симметричного шифрования на основе сети Фейстеля, который используется в целом ряде продуктов криптографической защиты, в частности некоторых версиях PGP и GPG.

# IDEA

- IDEA – симметричный блочный алгоритм шифрования данных, запатентованный швейцарской фирмой Ascom. Известен тем, что применялся в пакете программ шифрования PGP.

# Camellia

- Camellia – алгоритм симметричного блочного шифрования (размер блока 128 бит, ключ 128, 192, 256 бит), один из финалистов европейского конкурса NESSIE (наряду с AES и Shacal-2), разработка японских компаний Nippon Telegraph and Telephone Corporation и Mitsubishi Electric Corporation (представлен 10 марта 2000 г.).

# Основные преимущества блочного шифрования



Проще реализовать

Некоторые блочные шифры могут обеспечить защиту целостности.

# Основные недостатки блочного шифрования

Блочное шифрование медленное по сравнению с потоковым

Требует больше памяти

Чувствительны к шумам при передаче



# Потоковое шифрование в OpenSSL

В OpenSSL есть поддержка только одного потокового шифра

- ChaCha20

# ChaCha20

- ChaCha20 - это потоковый шифр, разработанный Дэниелом Бернштейном. Он работает путем перестановки 128 фиксированных битов, 128 или 256 бит ключа, 64-битного одноразового номера и 64-битного счетчика в 64 байта выходных данных. Этот вывод используется в качестве ключевого потока, при этом все неиспользуемые байты просто отбрасываются.


# Совместное использование с Poly1305

- Poly1305 – это криптографический код аутентификации сообщений (MAC), созданный Дэниелом Дж. Бернштейном . Его можно использовать для проверки целостности данных и подлинности сообщения.
- ChaCha20-poly1305 объединяет эти два примитива в аутентифицированный режим шифрования. Используемая конструкция основана на конструкции, предложенной для TLS Адамом Лэнгли, но отличается расположением данных, передаваемых на MAC, и добавлением шифрования длин пакетов .


# Скорость работы ChaCha20 против AES

	ChaCha20	AES-128-GCM	AES-256-GCM	AES-128-CBC	AES-256-CBC	Total	Score
AMD Ryzen 7 1800X	573	3006	2642	1513	1101	=	8835
Intel W-2125	565	2808	2426	1698	1235	=	8732
Intel i7-6700	585	2607	2251	1561	1131	=	8135
Intel Gold 5217	598	2344	2018	1396	1014	=	7370
AMD EPYC 7702	410	2464	2175	1241	904	=	7194
AMD EPYC 7551	355	2213	1962	1114	811	=	6455
AMD EPYC 7402P	493	2478	2184	1244	907	=	6062
Intel i5-6500	410	1729	1520	1078	783	=	5520
Intel i7-4750HQ	369	1556	1353	688	499	=	4465
AMD FX 8350	367	1453	1278	716	514	=	4328
AMD FX 8150	347	1441	1273	716	515	=	4292
Intel E5-2650 v4	404	1479	1286	652	468	=	4289
Intel i7-2700K	382	1353	1212	763	552	=	4262
Intel i7-3840QM	373	1279	1143	725	520	=	4040
Intel i5-2500K	358	1274	1140	728	522	=	4022
AMD FX 6100	326	1344	1186	671	481	=	4008
AMD A10-7850K	321	1303	1176	685	499	=	3984
AMD A8-7600 Kaveri	306	1246	1108	648	470	=	3778
Intel E5-2640 v3	303	1286	1126	585	419	=	3719
AMD Opteron 6380	293	1203	1063	589	423	=	3571
AMD Opteron 6378	282	1138	986	561	406	=	3373
AMD Opteron 6274	232	1054	926	524	376	=	3112
Intel Xeon E5-2630	247	962	864	541	394	=	3008
Intel Xeon E5645	262	817	717	727	524	=	3047
Intel i7-2635QM	151	989	881	564	404	=	2989
Intel Xeon L5630	225	701	610	626	450	=	2612
Intel E5-2603 v4	236	866	754	382	274	=	2512
AMD Opteron 2382	249	651	485	215	150	=	1750
Intel i7-950	401	256	218	358	257	=	1490
Intel Xeon X5550	287	205	175	305	219	=	1191
AMD Phenom 965	404	84	63	282	198	=	1031
Intel Core2 Q9300	231	126	133	221	161	=	872
AMD X4 610e	225	59	44	198	139	=	665
Intel Core2 Q6600	173	141	79	108	77	=	578
Intel P4 3Ghz Will	109	26	23	55	43	=	256
Intel ATOM D525	98	51	43	28	20	=	240
Snapdragon S4 Pro	131	41	-	-	-	=	172

# AES vs ChaCha20



Как было видно на  
предыдущем слайде  
chacha20 работает заметно  
быстрее на процессорах без  
аппаратной поддержки AES



Но если процессор  
поддерживает аппаратное  
ускорение то получаем  
обратный результат

# Преимущество потокового шифрования

Быстрее по сравнению с блочным шифрованием (при условии что на процессоре нет аппаратного ускорения)

Требует меньше памяти для работы

Менее чувствительны к шумам

# Недостатки потокового шифрования

Намного сложнее реализовать по сравнению с блочным шифрованием



Не обеспечивают целостность и аутентификацию

# Основной вывод во блочным и потоковым шифрам



Потоковые шифры лучше подходят когда объем данных либо неизвестен либо непрерывен например, сетевые потоки.



Блочные шифры лучше подходят если объем данных заранее известен - например, файл, поля данных или протоколы запроса/ответа, такие как HTTP, где длина всего сообщение известна уже в начале.



# Вывод

- На основе сделанной презентации можно сделать вывод что для оптимальной работы с шифрование данных лучше использовать сразу несколько видов шифрования (асимметричный и симметричный).
- То есть за счет асимметричного метода мы шифруем сообщение в котором храниться ключ для симметричного шифрования и передаем его клиенту или серверу
- Что касается выбора уже определенных методов для асимметричного (RSA и т.д.) и симметричного (AES и т.д.) шифрование. То здесь надо уже смотреть на то где будет использоваться данные методы т.к. все зависит от того что для нас является основным в шифрование
- Также в случае использование асимметричного метода необходимо определиться какой тип ключа будет использоваться эфемерный или статичный ключ.