Протоколи SSL і TLS

Протоколи SSL і TLS

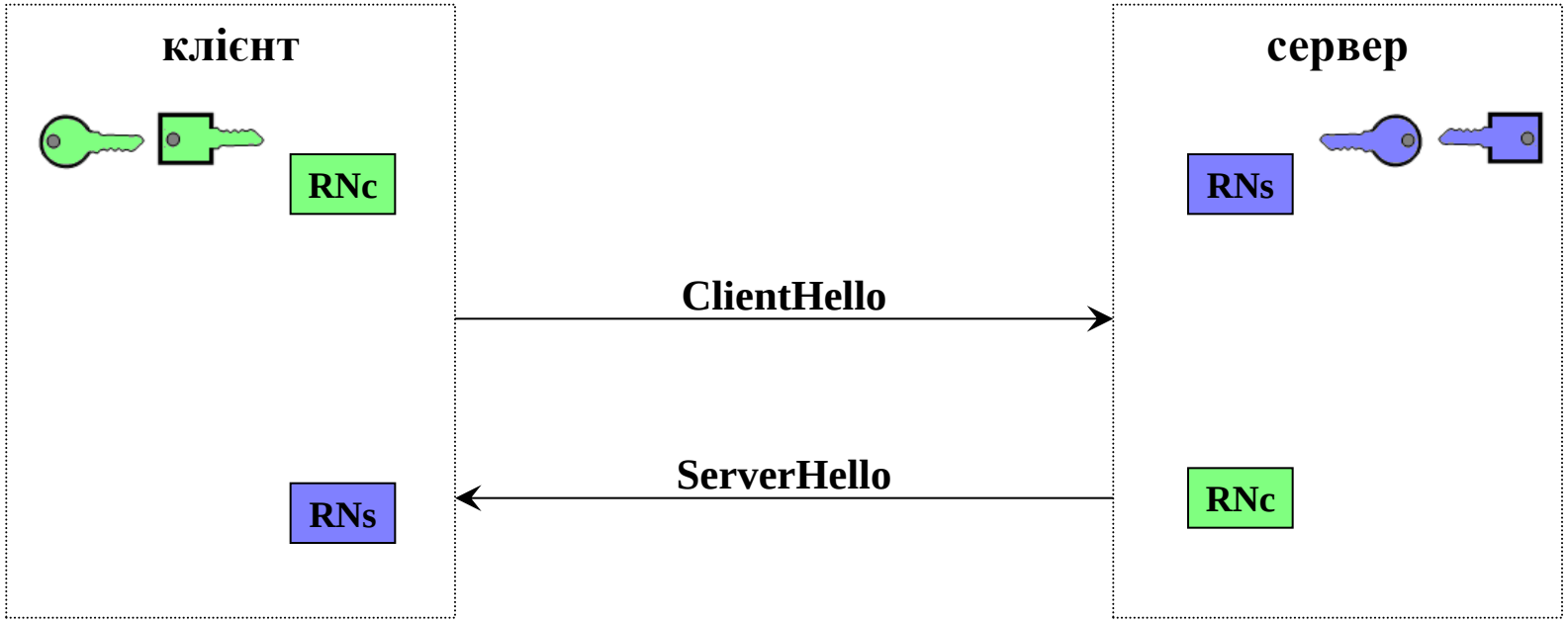
- Протокол безпеки транспортного рівня (Transport Layer Protocol - TLS) забезпечує захист комунікацій між додатками, в основному між веб-клієнтом і веб-сервером. Специфікація TLS базується на популярному протоколі Secure Socket Layer (SSL), розробленому корпорацією Netscape. Ці протоколи створювалися для забезпечення аутентифікації, цілісності та конфіденційності даних, якими обмінюються взаємодіють один з одним додатки. Обидва протоколи мають дворівневу організацію: протокол встановлення з'єднання (Handshake Protocol) і протокол передачі записів (Record Protocol).
- **Протокол встановлення з'єднання** дозволяє серверу та клієнту виконати взаємну аутентифікацію, узгодити застосовуваний алгоритм шифрування і криптографічні параметри перед тим, як протокол прикладного рівня почне передачу даних.
  Протокол передачі записів забезпечує захист протоколів більш високого рівня, включаючи протокол встановлення з'єднання. Протокол передачі записів залежить від надійності транспортного протоколу, такого як TCP.

Історія протоколів SSL/TLS

Коли	Хто	Що	Коментарі
1994	Netscape	Розроблено SSL 1.0	Ніколи не публікувався, оскільки були виявлені недоліки безпеки
1995	Netscape	Опубліковано SSL v2.0	Також було виявлено багато недоліків безпеки
1996	Netscape	Опубліковано SSL v3.0	SSL стає стандартом
1999	IETF	Опубліковано TLS v1.0 (SSL v3.1)	Багато виправлень, зміна імені та право власності на IETF
2006	IETF	Опубліковано TLS v1.1 (SSL v3.2)	Додаткові виправлення та нові можливості
2008	IETF	Опубліковано TLS v1.2 (SSL v3.3)	Зараз використовується на більшості вузлів
2014	IETF	TLS v1.3 проект 1 (SSL v3.4)	
2018	IETF	Опубліковано TLS v1.3	Сучасна версія

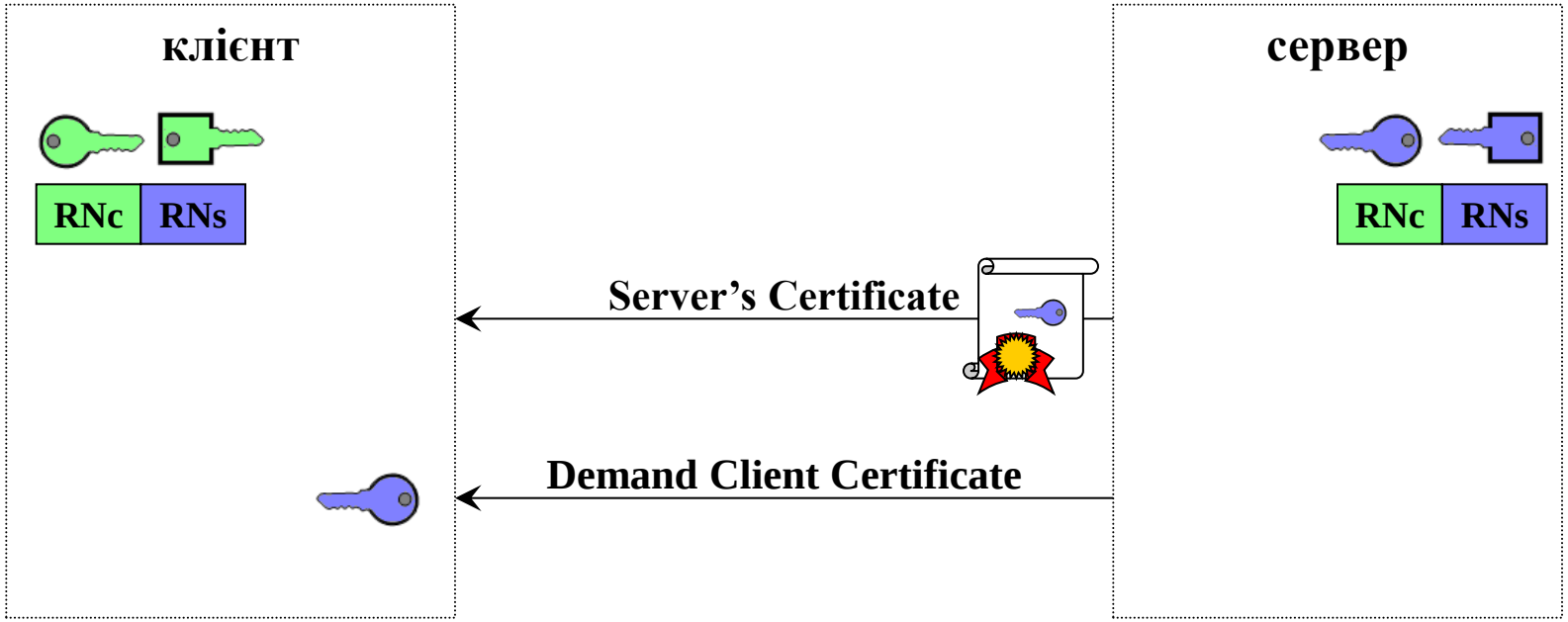
Протокол встановлення з'єднання

- Узгодження алгоритмів
  - Клієнт повідомляє версію протоколу, випадкове число і список підтримуваних алгоритмів – ClientHello
  - Сервер повідомляє обрану версію протоколу, своє випадкове число і вибрані алгоритми – ServerHello



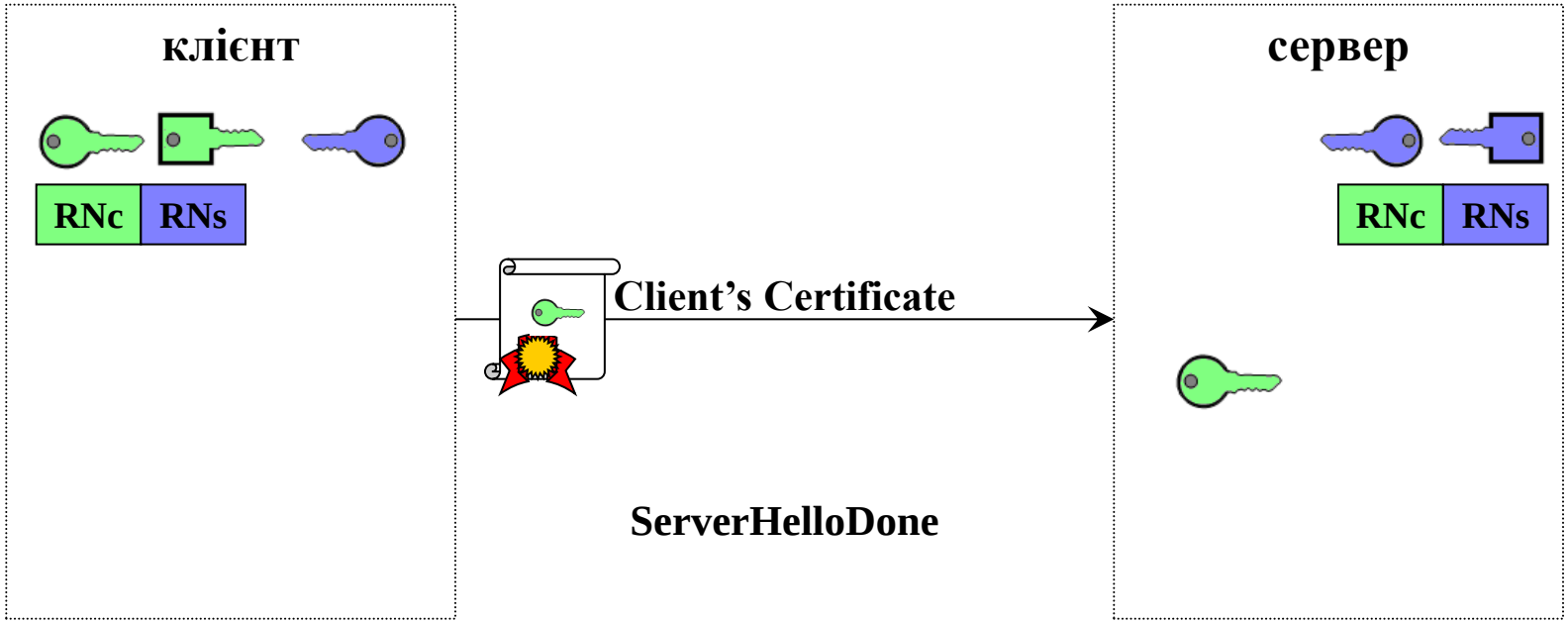
Протокол встановлення з'єднання

- Автентифікація сервера
  - сервер висилає свій сертифікат (X.509 або OpenPGP)
  - сервер може запросити сертифікат клієнта, щоб автентифікувати його
  - клієнт перевіряє сертифікат сервера, використовуючи PKI



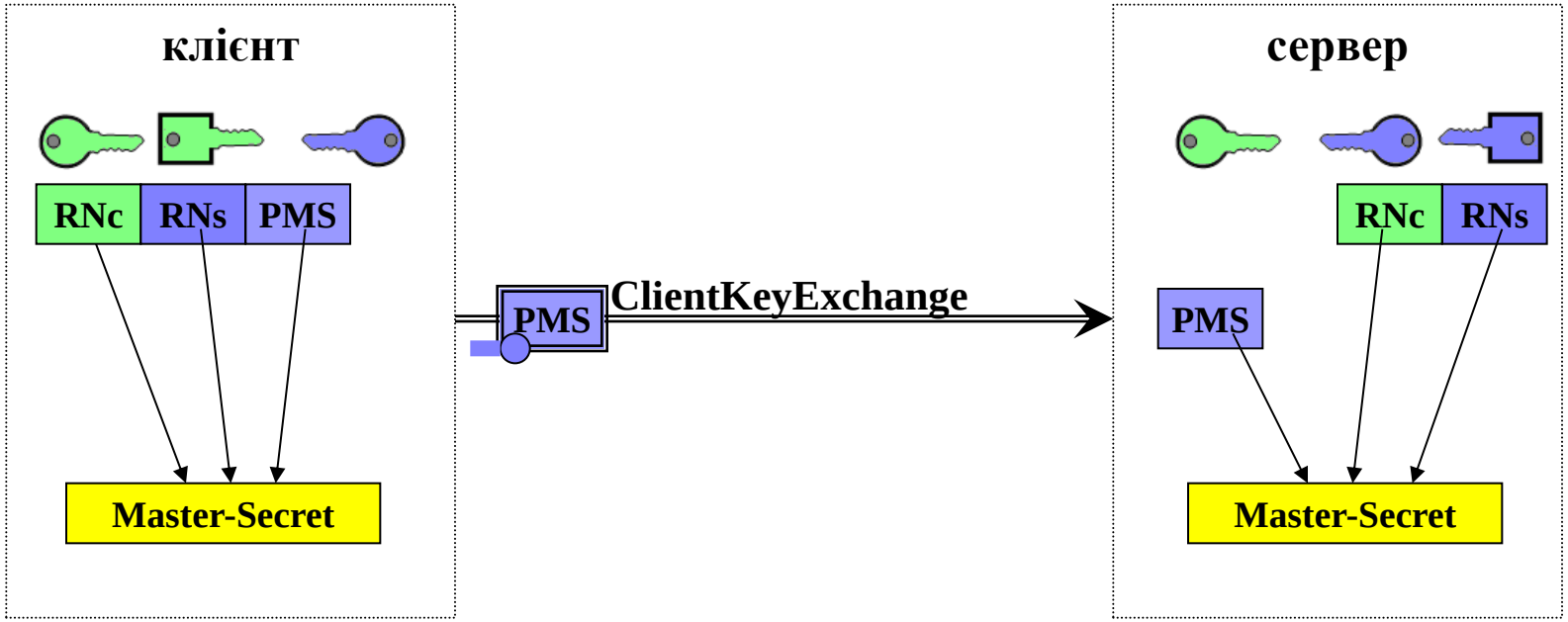
Протокол встановлення з'єднання

- Автентифікація клієнта (опціонально)
  - Клієнт може надати свій сертифікат, тоді сервер, використовуючи PKI, перевіряє автентичність клієнта - так забезпечується взаємна аутентифікація
  - Фаза аутентифікації закінчується повідомленням ServerHelloDone



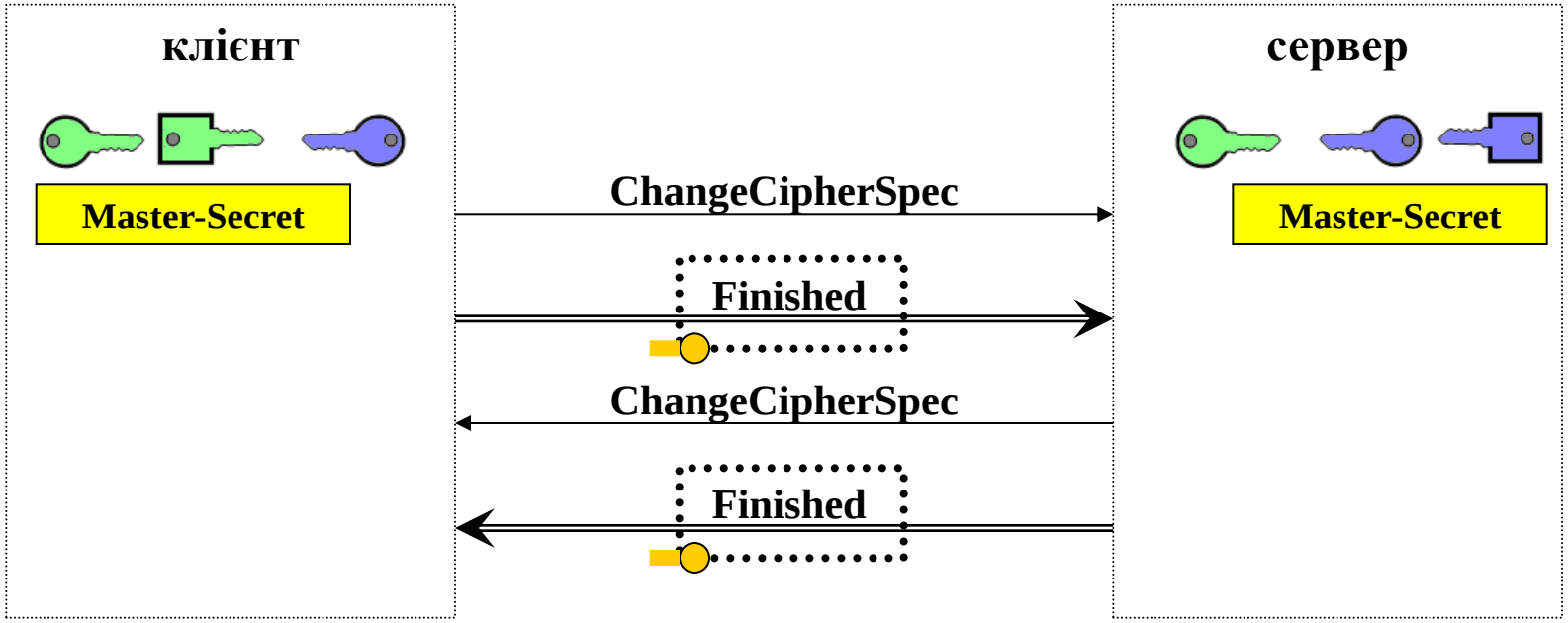
Протокол встановлення з'єднання

- Генерація ключа сесії
  - Клієнт генерує Pre-Master-Secret і пересилає його серверу в повідомленні ClientKeyExchange
  - Клієнт і сервер на основі RNc, RNs і PMS генерують ключ для симетричного криптоалгоритму



Протокол встановлення з'єднання

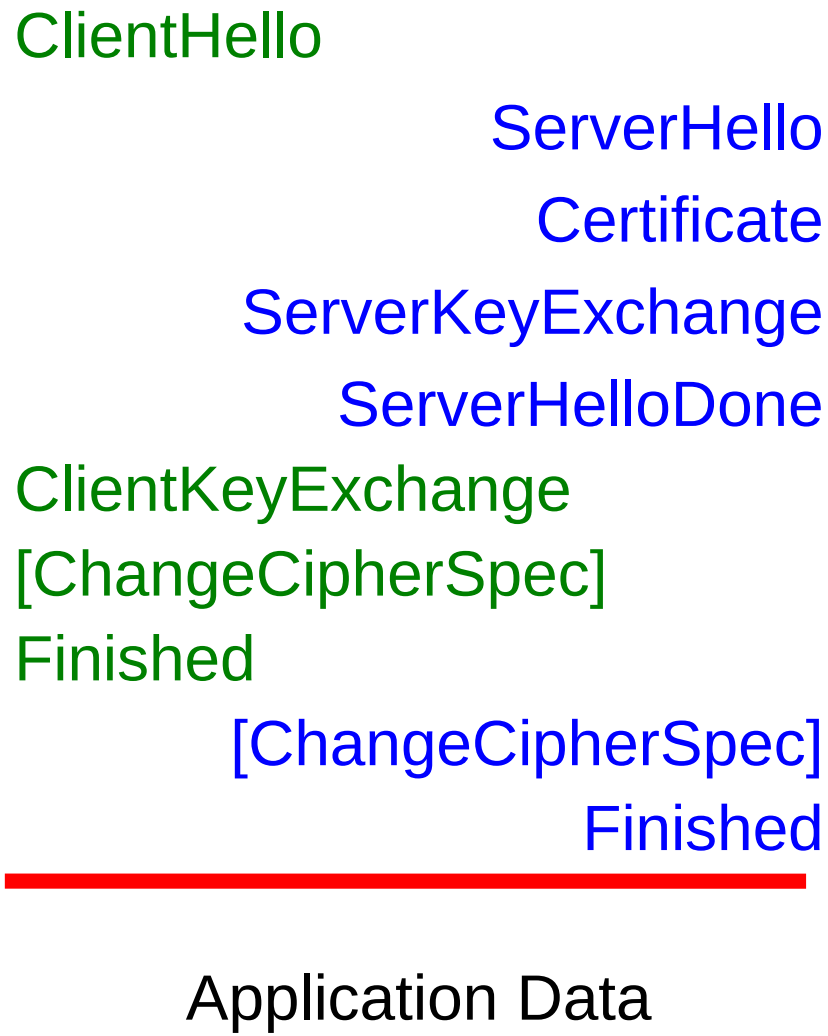
- Завершення узгодження параметрів
  - Клієнт посилає повідомлення про перехід в режим шифрування ChangeCipherSpec і посилає зашифроване повідомлення про завершення узгодження з гешем всіх повідомлень
  - Сервер посилає ChangeCipherSpec і зашифроване повідомлення про завершення узгодження з гешем всіх повідомлень

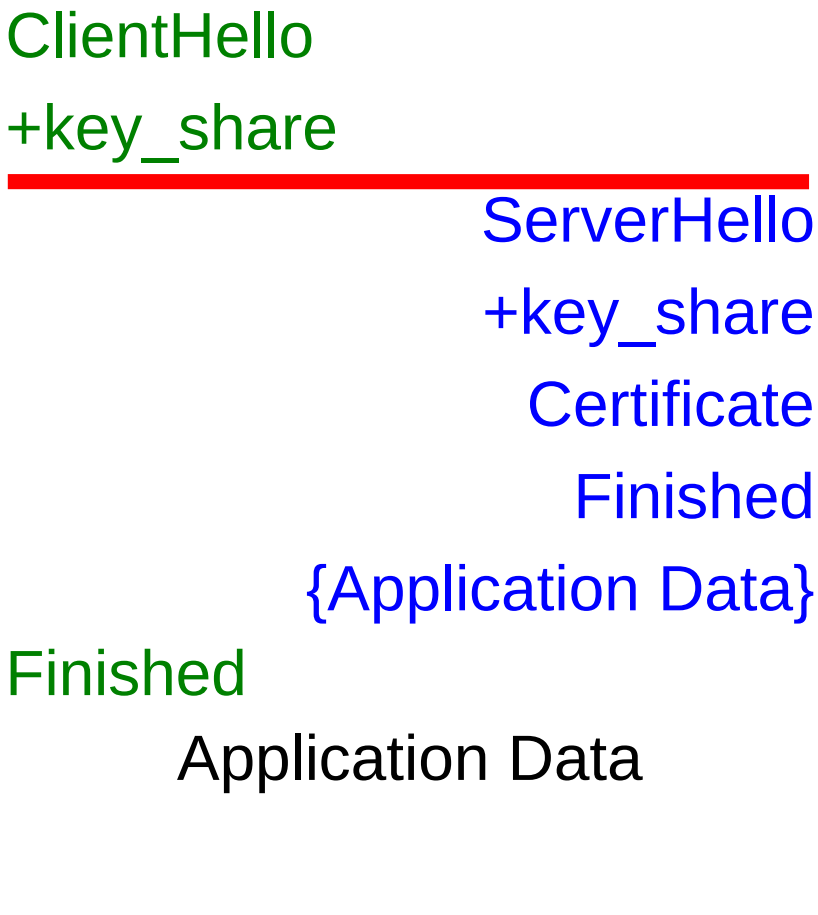


Цілі проектування TLS 1.3

- Кращий мережного трафіку
- Видалити всі небезпечні компоненти
- Зробити TLS швидше

Новий протокол встановлення з'єднання





TLS 1.3 та DPI

- До:
  - Нешифроване ім'я сервера
  - Нешифрований сертифікат
- Зараз:
  - Нешифроване ім'я сервера
  - Зашифрований сертифікат
- Незабаром:
  - Зашифроване ім'я сервера

Аутентифікація та сертифікати

- RSA-PSS замість PKCS1-v1.5
- Більше немає сертифікатів DSA
- Більше немає статичного DH
- Новий режим PSK

Режими шифрування

- AES-128, AES-256, ChaCha
- Тільки режими AEAD:
  - AES-GCM, AES-CCM, ChaCha-Poly1305
- Більше немає:
  - Режим CBC
  - DES/3DES, RC4, ARIA, CAMELLIA
  - SHA1, MD5 ...
- Більше немає стиснення