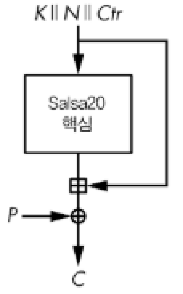
Salsa20

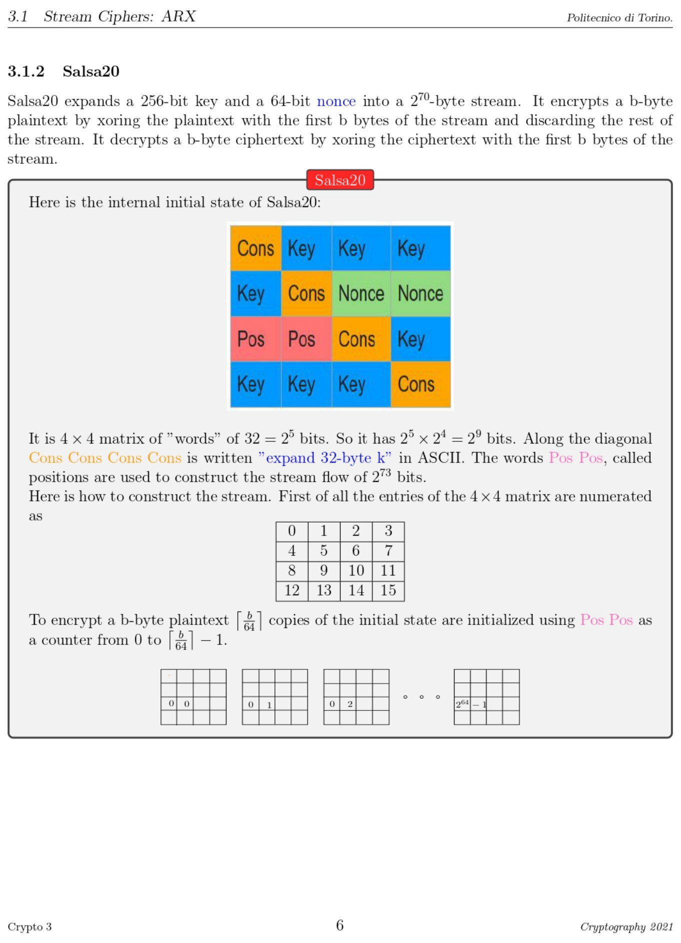
March 23, 2021

1 How it is formed

Salsa20 is a counter-based stream cipherit generates its keystream by repeatedly processing a counter incremented for each block. As you can see in Figure 5-10, the Salsa20 core algorithm transforms a 512-bit block using a key (K), a nonce (N), and a counter value (Ctr). Salsa20 then adds the result to the original value of the block to produce a keystream block. (If the algorithm were to return the cores permutation directly as an output, Salsa20 would be totally insecure, because it could be inverted. The final addition of the initial secret state K , N , Ctr makes the transform key-to-keystream-block non-invertible.)



(sorry for the foreign language; coudn't find a picture in english but that sentence means SALSA20 CORE): REFERENCE IN THE BOOK: SERIOUS CRYPTOGRAPHY(CHAPTER 5)



3.1.2 Salsa20

Salsa20 expands a 256-bit key and a 64-bit nonce into a 2^{70}-byte stream. It encrypts a b-byte plaintext by xoring the plaintext with the first b bytes of the stream and discarding the rest of the stream. It decrypts a b-byte ciphertext by xoring the ciphertext with the first b bytes of the stream.

Salsa20

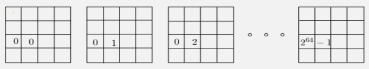
Here is the internal initial state of Salsa20:

Cons	Key	Key	Key
Key	Cons	Nonce	Nonce
Pos	Pos	Cons	Key
Key	Key	Key	Cons

It is 4×4 matrix of "words" of $32 = 2^5$ bits. So it has $2^5 \times 2^4 = 2^9$ bits. Along the diagonal Cons Cons Cons Cons is written "expand 32-byte k" in ASCII. The words Pos Pos, called positions are used to construct the stream flow of 2^{73} bits.
Here is how to construct the stream. First of all the entries of the 4×4 matrix are numerated as

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

To encrypt a b-byte plaintext $\lceil \frac{b}{64} \rceil$ copies of the initial state are initialized using Pos Pos as a counter from 0 to $\lceil \frac{b}{64} \rceil - 1$.



1.1 observation from the page taken from notes

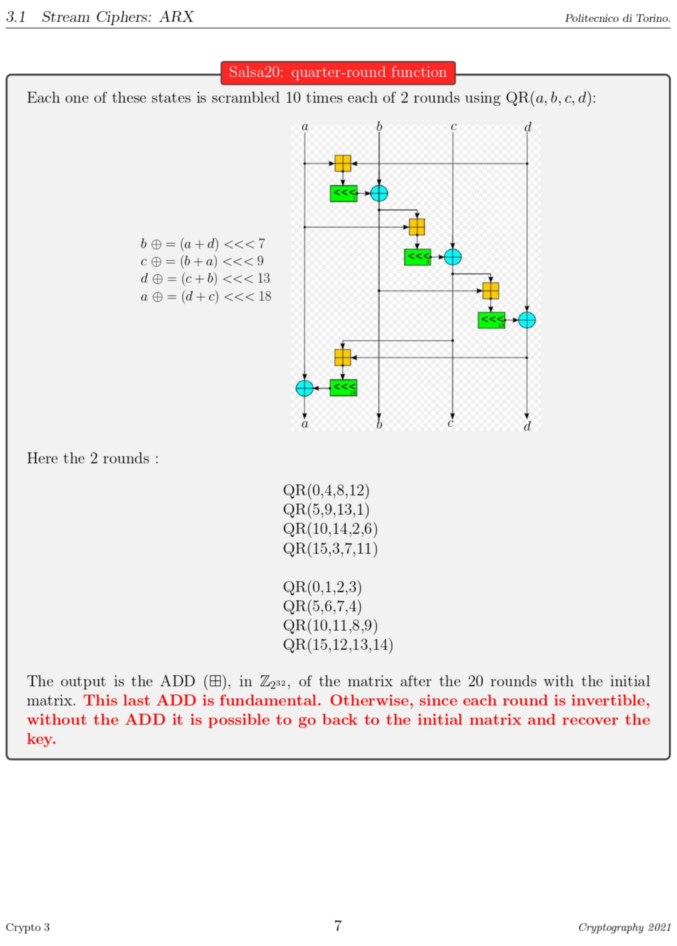
This is a matrix of 4x4 word(4 bytes each); meaning that each block give a keystream of 64 bytes(4*4*4). That is why we have to take (b/64) to get the total number of blocks needed to make a valid key stream for a plaintext of b bits.

2 The quarter-round function

Salsa20s core permutation uses a function called quarter-round (QR) to transform four 32-bit words (a, b, c, and d), as shown here:

b=b xor [(a+d) <<< 7]
c=c xor [(b+a) <<< 9]
d=d xor [(c+b) <<< 13]
a=a xor [(d+c) <<< 18]

These four lines are computed from top to bottom, meaning that the new value of b depends on a and d, the new value of c depends on a and on the new value of b (and thus d as well), and so on. The operation <<< is wordwise left-rotation by the specified number of bits, which can be any value between 1 and 31 (for 32-bit words).

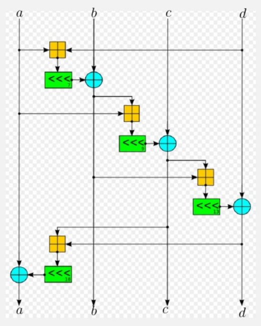


3.1 Stream Ciphers: ARX *Politecnico di Torino.*

Salsa20: quarter-round function

Each one of these states is scrambled 10 times each of 2 rounds using QR(a, b, c, d):

$b \oplus= (a+d) <<< 7$
$c \oplus= (b+a) <<< 9$
$d \oplus= (c+b) <<< 13$
$a \oplus= (d+c) <<< 18$



Here the 2 rounds :

QR(0,4,8,12)
QR(5,9,13,1)
QR(10,14,2,6)
QR(15,3,7,11)

QR(0,1,2,3)
QR(5,6,7,4)
QR(10,11,8,9)
QR(15,12,13,14)

The output is the ADD (⊞), in $\mathbb{Z}_{2^{32}}$, of the matrix after the 20 rounds with the initial matrix. **This last ADD is fundamental. Otherwise, since each round is invertible, without the ADD it is possible to go back to the initial matrix and recover the key.**

Crypto 3 7 *Cryptography 2021*

2.1 Observation from the teacher and students

1)In case of the syncronized mode,how is the IV initialized

Answer:It depends; u can make it public if u wanted; it depends on what you choose or u can even choose some random thing and share it.

2)Key whiting: it is the addition done at the end of the quarter-round function and the key; since if we don't do it; it would be reversible hence insecure.

3)the meaning of (a,b,c,d) in the matrix: Prof's answer: It means that when we pass those 4 numbers in the qr function; they are going to be updated according to the schema above.

4)with a certain number of registers each with a part of the stream with bytes in it; in what order do we xor with the plain text? Prof answer: In the bernstein paper, they had in mind the little endian notation in the lower level but on higher level like in python, it depends on how u implemented the bytes(for example in python u use lists)

5)where does the rotation number coming from(7,9,13,18): Answer: They are some ratio in the berstein paper to some explanation they used those numbers

6)Salsa works on plaintext of variable size because only the number of registers change

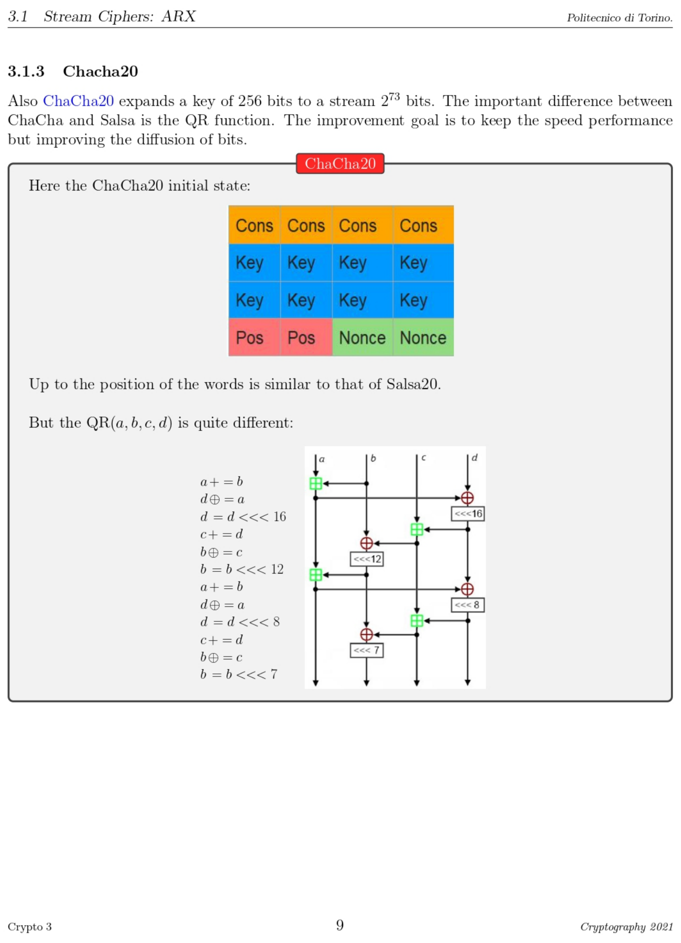
7)what do we do at decryption side?: answer: You just do the same; no need of the reverse since it is just a xor with the cyphertext.

2.2 Salsa20 implementation example in python

```
from salsa20 import Xsalsa20xor
    from os import urandom
    IV = urandom(24)
    KEY = b'*secret**secret**secret**secret*'
    ciphertext = XSalsa20xor(b"IT'S A YELLOW SUBMARINE", IV, KEY)
    print(ciphertext)
    print(XSalsa20xor(ciphertext, IV, KEY).decode())
```

3 CHACHA20

The only difference between chacha20 is the quarter-round.



3.1.3 Chacha20

Also ChaCha20 expands a key of 256 bits to a stream 2^{73} bits. The important difference between ChaCha and Salsa is the QR function. The improvement goal is to keep the speed performance but improving the diffusion of bits.

ChaCha20

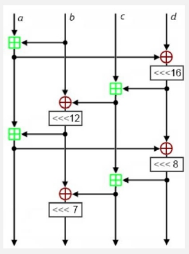
Here the ChaCha20 initial state:

Cons	Cons	Cons	Cons
Key	Key	Key	Key
Key	Key	Key	Key
Pos	Pos	Nonce	Nonce

Up to the position of the words is similar to that of Salsa20.

But the QR(a, b, c, d) is quite different:

$$\begin{aligned}
&a += b\\
&d \oplus= a\\
&d = d <<< 16\\
&c += d\\
&b \oplus= c\\
&b = b <<< 12\\
&a += b\\
&d \oplus= a\\
&d = d <<< 8\\
&c += d\\
&b \oplus= c\\
&b = b <<< 7
\end{aligned}$$



3.0.1 what is added

confusion: (Shanon theory) It is related with permutation; (This refers to change the position of values in the register)

difusion: register and matrix; the register multiplied with matrix; (this is a way of calculating the media since the matrix perse is populated with 1s and 0s)