Stream Ciphers

1. The keystream of a synchronous stream cipher depends on the key and an initialization vector IV. If the key is fixed and the IV has only 24 bits, then the keystream repeats after $2^{24} = 16,777,216$ initializations at most. Uniform random 24-bit IVs are likely to have a collision after only $2^{12} = 4096$ values. This is a vulnerability, since encryption with identical keystreams is completely insecure.
2. $p(x) \in GF(2)[x]$ is irreducible. However, $p(x)$ is not primitive since $x^5 \equiv 1 \bmod p(x)$. The period of each nonzero output sequences is $\operatorname{ord}(p(x)) = 5$.
3. Let $n = \deg(c(x))$. Then $p(x) = x^n c(\frac{1}{x})$ and $c(x) = x^n p(\frac{1}{x})$. It is sufficient to show only one direction of the statements. Suppose that $c(x)$ is reducible and $c(x) = c_1(x)c_2(x)$. Then $\deg(c_1(x)) + \deg(c_2(x)) = n$ and
$$x^n c\left(\frac{1}{x}\right) = x^n c_1\left(\frac{1}{x}\right) c_2\left(\frac{1}{x}\right) = x^{\deg(c_1)} c_1\left(\frac{1}{x}\right) x^{\deg(c_2)} c_2\left(\frac{1}{x}\right).$$
This gives a factorization of $p(x)$. For the second claim, suppose that $N \in \mathbb{N}$ and $p(x) = x^n c(\frac{1}{x}) \mid x^N - 1$. Then $c(x) = x^n p(\frac{1}{x}) \mid 1 - x^N$. Therefore, if $c(x)$ is primitive, then $p(x)$ is primitive, too.
4. We obtain ten keystream bits: $m \oplus c = 10010\ 10110$. The first five bits give the state
$$st = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$
The subsequent states are $(1,0,1,0,0)^T$, $(0,1,0,1,0)^T$, $(1,0,1,0,1)^T$, $(1,1,0,1,0)^T$, $(0,1,1,0,1)^T$, and we have a system of linear equations modulo 2:
$$\begin{aligned} 1 &= x_2 + x_5 \\ 0 &= x_1 + x_3 \\ 1 &= x_2 + x_4 \\ 1 &= x_1 + x_3 + x_5 \\ 0 &= x_1 + x_2 + x_4 \end{aligned}$$
The unique solution is $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 1$, $x_5 = 1$. The connection polynomial is $c(x) = 1 + x + x^3 + x^4 + x^5$ and the corresponding characteristic polynomial is $p(x) = x^5 + x^4 + x^2 + x + 1$. The polynomials are primitive and the period is $2^5 - 1 = 31$. The complete 31-bit output is:
$$1,0,0,1,0,1,0,1,1,0,0,0,0,1,1,1,0,0,1,1,0,1,1,1,1,1,0,1,0,0,0.$$
Subsequently, the output bits recur.
5. We have
$$\begin{aligned} &maj(0,0,0) = maj(0,0,1) = maj(0,1,0) = maj(1,0,0) = 0, \\ &maj(0,1,1) = maj(1,0,1) = maj(1,1,0) = maj(1,1,1) = 1. \end{aligned}$$
For each of the three input bits, in six out of eight combinations the input bit coincides with the majority bit.

6. The three connection polynomials of A5/1 are primitive:

```
sage: R.<x> = PolynomialRing(GF(2))
sage: R(x^19+x^18+x^17+x^14+1).is_primitive()
True
sage: R(x^22+x^21+1).is_primitive()
True
sage: R(x^23+x^22+x^21+x^8+1).is_primitive()
True
```

   The periods of the three registers are $2^{19}-1$, $2^{22}-1$ and $2^{23}-1$, respectively. Their product is an upper bound of the period of A5/1:

$$(2^{19}-1)(2^{22}-1)(2^{23}-1) \approx 2^{64}$$

7. Suppose $K[0] + K[1] \equiv 0 \bmod 256$. Then the first two iterations of the key scheduling algorithm give

$$S[0] = K[0],\ S[1] = K[0] + K[1] + 1 \equiv 1 \bmod 256.$$

   The next iteration probably yields

$$S[2] = K[0] + K[1] + K[2] + 3 \equiv K[2] + 3 \bmod 256.$$

   If $S[1]$ and $S[2]$ do not change later during the key scheduling, the first first output byte is

$$B = S[S[1] + S[S[1]]] = S[1 + S[1]] = S[2] \equiv K[2] + 3 \bmod 256.$$

8. The Salsa20 quarter-round is invertible; applying the four XOR operations of a quarter-round in reverse order gives the inverse map:

$$\begin{aligned} a &= a \oplus ((d+c) \lll 18) \\ d &= d \oplus ((c+b) \lll 13) \\ c &= c \oplus ((b+a) \lll 9) \\ b &= b \oplus ((a+d) \lll 7) \end{aligned}$$

9. Let $S = \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 & y_7 \\ y_8 & y_9 & y_{10} & y_{11} \\ y_{12} & y_{13} & y_{14} & y_{15} \end{pmatrix}$, then

$$\textit{row-round}\ (S^T) = \begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 & z_7 \\ z_8 & z_9 & z_{10} & z_{11} \\ z_{12} & z_{13} & z_{14} & z_{15} \end{pmatrix}, \text{ where}$$

$$\begin{aligned} (z_0, z_1, z_2, z_3) &= \textit{quarter-round}\ (y_0, y_4, y_8, y_{12}), \\ (z_5, z_6, z_7, z_4) &= \textit{quarter-round}\ (y_5, y_9, y_{13}, y_1), \\ (z_{10}, z_{11}, z_8, z_9) &= \textit{quarter-round}\ (y_{10}, y_{14}, y_2, y_6), \\ (z_{15}, z_{12}, z_{13}, z_{14}) &= \textit{quarter-round}\ (y_{15}, y_3, y_7, y_{11}). \end{aligned}$$

It follows that

$$\textit{column-round}\ (S) = (\textit{row-round}\ (S^T))^T = \begin{pmatrix} z_0 & z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 & z_7 \\ z_8 & z_9 & z_{10} & z_{11} \\ z_{12} & z_{13} & z_{14} & z_{15} \end{pmatrix}, \text{ where}$$

$$(z_0, z_4, z_8, z_{12}) = \textit{quarter-round}\ (y_0, y_4, y_8, y_{12}),$$
$$(z_5, z_9, z_{13}, z_1) = \textit{quarter-round}\ (y_5, y_9, y_{13}, y_1),$$
$$(z_{10}, z_{14}, z_2, z_6) = \textit{quarter-round}\ (y_{10}, y_{14}, y_2, y_6),$$
$$(z_{15}, z_3, z_7, z_{11}) = \textit{quarter-round}\ (y_{15}, y_3, y_7, y_{11}).$$

10. We obtain:

    $$\textit{quarter-round}\ (1,0,0,0) = (08008145, 00000080, 00010200, 20500000),$$
    $$\textit{quarter-round}\ (0,1,0,0) = (88000100, 00000001, 00000200, 00402000),$$
    $$\textit{quarter-round}\ (0,0,0,1) = (00048044, 00000080, 00010000, 20100001).$$

11. The addition modulo 2^{32} is not $GF(2)$-linear.
12. Without the final addition, one could invert the ten *double-round* operations and recover the state which includes the key.
13. Without the *diagonal-round* operations, ChaCha would transform each column independently from each other. In this case, there would be insufficient diffusion among the state entries and ChaCha would consist of four stream ciphers with 64-bit keys.