1. If Z^*_m has primitive roots, α is a primitive root modulo m if and only if

$$\alpha^{\frac{\varphi(m)}{r}} \neq 1 \, mod \, m$$

for any prime divisor r of $\varphi(m)$.

Consider primes of the form $p = 3 \cdot 2^n + 1$. When $n = 1$, we have $p = 7$; since (2/7) = (1/7) because $7 \equiv -1 \pmod 8$.

i.e., (2/7) = 1

i.e., $2^{(7-1)/2} \equiv 1 \bmod 7$ $\quad \because \left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p$

Hence 2 is not a primitive root modulo 7.

When $n = 2$, $p = 13$, and 2 is a primitive root modulo 13.

Assume that $n \geq 3$. Then $p = 8 \cdot 3 \cdot 2^{n-3} + 1$, and so $p \equiv 1 \pmod 8$.

This implies that $(2/p) = (1/p) = 1$; Because $(2m/n) = (m/n)$ if $n = \pm 1 \bmod 8$.

i.e., 2 is a quadratic residue modulo p.

Hence 2 is not a primitive root modulo p.

2. P('d') = 1- (.5 + .15 + .03) = .05.

If we shift by a multiple of the key length, the probability of coincidence is

$(.5)^2 + (.15)^2 + (.3)^2 + (.05)^2 = 0.365$.

Expected largest number of coincidences = 1000×0.365 = 365

3. For positive integers n, $\varphi(n) \mid n!$

If n is odd, then (by Euler's theorem) $2^{\varphi(n)} \equiv 1 (\bmod n)$

Therefore $2^{n!} \equiv 1 \pmod n \Rightarrow n | 2^{n!} - 1$

4. Since n, n' are public, attacker can compute gcd (n, n'), yielding p, using the Euclidean Algorithm.

Then the attacker computes n/p and n'/p, yielding q, q'.

Finally, the attacker can find d, d' satisfying

$de \equiv 1 \pmod{(p-1)(q-1)}$ and $d'e' \equiv 1 \pmod{(p'-1)(q'-1)}$.

5. let $n = 31803221$, given $2^{n-9} \equiv 27696377 \pmod{31803221}$.

By the Fermat's theorem for any prime number p and $a \in Zp$

$$a^{p-1} \equiv 1 \pmod p,$$

$2^{n-9} \cdot 2^8 \equiv 27696377 \cdot 256 \equiv 29957450 \neq 1 \pmod{31803221}$.

Hence, n is not a prime number!

6. From the given, $\frac{1}{x} + \frac{1}{y} = \frac{1}{n} \Leftrightarrow xy = nx + ny \Leftrightarrow (x-n)(y-n) = n^2$.

Here $n = 84 = 2^2 \cdot 3 \cdot 7$.

Since $x, y > n$, there is a 1-1 correspondence between the solutions in (x, y) and the divisors of 84^2, so the number of solutions is

$$\tau(84^2) = (2 \cdot 2 + 1)(2 \cdot 1 + 1)(2 \cdot 1 + 1) = 45 \text{ Ans}$$

7. $x^8 + x^4 + x^3 + x + 1 = (x^7 + x^3 + x^2 + 1) \cdot x + 1$

   $\Rightarrow x \cdot (x^7 + x^3 + x^2 + 1) \equiv 1 \pmod{x^8 + x^4 + x^3 + x + 1}$

   Hence the inverse of x is $x^7 + x^3 + x^2 + 1 + 1$.

8. Let $P(x_1, y_1)$ be a point in E, then $x_1^{15} = 1$ for all $x_1 \in F_{2^4}$.
   $2P = (x_1^4, 1 + y_1^4) \Rightarrow 4P = (x_1^{16}, 1 + 1 + y_1^{16}) = (x_1^{16}, y_1^{16}) = (x_1, y_1) = P$
   i.e. 4P = P $\Rightarrow$ 3P = 0 $\Rightarrow$ order of P is 3.

9. $Z_3[x]/(x^3 + cx^2 + 1)$ is a field if and only if $x^3 + cx^2 + 1$ is irreducible in $Z_3[x]$.
   Let $f(x) = x^3 + cx^2 + 1$ since $f(x)$ has degree 3, if $f(x)$ is reducible then one of the factors should have degree one i.e.,
   $f(x) = x^3 + cx^2 + 1$ should have a root in Z_3.
   $f(0) = 1 \neq 0 \pmod 3$
   $f(1) = 2 + c$
   $f(2) = 9 + c \equiv c \pmod 3$
   So, if $c \neq 0 \bmod 3$ and $c \neq 1 \bmod 3$ in Z_3, then $f(x)$ is irreducible.
   So, for $c \equiv 2 \bmod 3$, $f(x)$ is irreducible.

10. security strength $s = 256$ bits
    Therefore, capacity $c \geq 512$
    Since $b = r + c$, In keccak $b = 1600$
    $\therefore$ $r = 1600 - 512 = 1088$
    Message length = 4000
    No. of padding bits = 1088 - 4000 (mod 1088) = 352
    Number of 1s is 2. Therefore, number of zeros to be padded is 350.

11. DES is a Feistel-based cipher. In such ciphers, the function S need not be invertible.
    Here's the reason:
    In each round, the following operation is applied:
    for $i = 0, 1, \ldots, n$
    $L_{i+1} = R_i$
    $R_{i+1} = L_i \oplus f(R_i, K_i)$
    Decryption is performed as follows:
    $R_i = L_{i+1}$
    $L_i = R_{i+1} \oplus f(L_{i+1}, K_i)$

As you can see, the decryption does not need f to be invertible. (Since the decryption does not need to compute f^{-1}.)

12. The state in Rijndael is a 4 × 4 matrix with entries in the field of 256 elements. The shift row layer shifts each row to the right a certain amount, wrapping the entries around.

    More precisely, the first row is not shifted, the second row is shifted by one, the third row is shifted by two, and the fourth row is shifted by three.

    The Byte Substitution layer can be viewed as a lookup table. Each matrix entry, represented by an 8-bit byte, is broken into two pieces which index the rows and columns of a 16 × 16 lookup matrix. The byte is replaced by the corresponding entry in the table, which is another 8-bit byte.

    The Byte Substitution layer is applied entry by entry to the state, with all entries treated in the same way.

    The Row Shift layer simply moves the bytes around.

    Thus, it doesn't matter in which order we apply these layers: shifting and substituting is the same as first substituting then shifting.

13. Consider two signatures $[r, s_1]$ and $[r, s_2]$ for messages m_1 and m_2 respectively.
    Here $r = \alpha^k \bmod p$, $s_1 = k^{-1}(m_1 - dr) \bmod (p-1)$ and $s_2 = k^{-1}(m_2 - dr) \bmod (p-1)$
    $s_1 - s_2 = (m_1 - m_2)\, k^{-1} \bmod (p-1)$
    Therefore $k = (s_1 - s_2)^{-1}(m_1 - m_2) \bmod (p-1)$
    Find $d = (m_1 - s_1 \cdot k) \cdot r^{-1} \bmod p - 1$.
    (Because $s_1 = k^{-1}(m_1 - dr) \bmod (p-1) \Rightarrow dr = (m_1 - s_1 \cdot k) \bmod p - 1$).
    Hence interceptor can find d, after knowing m_1, s_1, k, r, p.

14.

    (a) h is pre-image resistant, if for a given $y \in \{1, 2, \ldots, p-1\}$, it is difficult to find $m \in Z$ such that $h(m) = y$.
    Finding such an m is equivalent to solving the congruence $\alpha^m \equiv y \pmod p$.
    i.e., finding m is equivalent to finding Discrete Logarithm modulo p, which is computationally infeasible for large p.
    Hence h is pre-image resistant.

    (b) For numbers m_1, m_2, we know that
    $m_1 \equiv m_2 \pmod{p-1}$ m_1 may not be equal to m_2
    $\Rightarrow \alpha^{m1} \equiv \alpha^{m2} \pmod p$
    $\Rightarrow h(m_1) = h(m_2)$.
    Therefore, since it is easy to pick two distinct numbers m_1 and m_2 with

$m_1 \neq m_2$; $m_1 \equiv m_2 \pmod{p-1}$ but $h(m_1) = h(m_2)$.

It follows that it is easy to find collisions for h.

Hence h is not collision resistant.

15.

(a) "relinearization". The exact complexity of this algorithm is not known, but for sufficiently overdefined systems it was expected to run in polynomial time.

(b) GF(2^8).

(c) Light weight cryptographic algorithms

(d) cryptographic key is linked with user's fingerprint data

(e) Since the sequences produced are periodic.

16. From the assumption we know that the state bits of LFSR are

$s_{2i} = c_i \oplus m_i$ i.e., $s_2 = 1$, $s_4 = 0$, $s_6 = 0$, $s_8 = 0$.

The connection polynomial corresponds to the recursion:

$s_i = s_{i-1} + s_{i-4} \quad i \geq 5$

Hence,

$s_5 = s_4 + s_1$

$s_6 = s_5 + s_2$

$s_7 = s_6 + s_3$

$s_8 = s_7 + s_4$

This gives

$s_5 = s_2 + s_6 = 1$ and therefore $s_1 = s_4 + s_5 = 1$.

$s_7 = s_4 + s_8 = 0$ which gives $s_3 = s_6 + s_7 = 0$.

Thus, the secret state is $(s_1, s_2, s_3, s_4) = (1, 1, 0, 0)$

To find m_5, m_6, m_7 we need to find s_{10}, s_{12}, s_{14}.

Using recursion compute,

$s_9 = s_8 + s_5 = 1$

$s_{10} = s_9 + s_6 = 1$

$s_{11} = s_{10} + s_7 = 1$

$s_{12} = s_{11} + s_8 = 1$

$s_{13} = s_{12} + s_9 = 0$

$s_{14} = s_{13} + s_{10} = 1$

Thus,

$m_5 = c_5 + s_{10} = 1 + 1 = 0$,

$m_6 = c_6 + s_{12} = 1 + 1 = 0$,

$m_7 = c_7 + s_{14} = 0 + 1 = 1$.