Definition:
<u>Private-key Encryption Scheme</u>: Specify a message space $\mathcal{M}$, and Gen, Dec, and Enc algorithms. Gen is a probabilistic algorithm that outputs a key k. Enc takes k and $m \in \mathcal{M}$, and outputs ciphertext c. Notation: $c = \mathsf{Enc}_k(m)$. Dec takes k and c, and outputs m. Notation: $m = \mathsf{Dec}_k(c)$. Must have $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ for all k. The set of valid keys is $\mathcal{K}$. WLOG, assume Gen chooses a uniform $k \in \mathcal{K}$.

Definition: <u>Kerchoffs' Principle</u>: An encryption scheme should be designed to be secure *even if* an eavesdropper knows all the details of the scheme, so long as the attacker doesn't know the key being used.

Definition:
<u>Sufficient Key-space Principle</u>: Any secure encryption scheme must have a key space that is sufficiently large to make an exhaustive-search attack infeasible. Necessary, but not sufficient.

Theorem: <u>Bayes Theorem</u>: $\Pr[A|B] = \frac{\Pr[B|A]\cdot\Pr[A]}{\Pr[B]}$.

Definition: <u>Perfect Secrecy</u>: An encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ such that for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$, for which $\Pr[C = c] > 0$, $\Pr[M = m \mid C = c] = \Pr[M = m]$. Equivalently, $\forall m, m' \in \mathcal{M}, c \in \mathcal{C}$, $\Pr[\mathsf{Enc}_K(m) = c] = \Pr[\mathsf{Enc}_K(m') = c]$.

Definition: <u>$\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$</u>: (Adversarial Indistinguishability Experiment): The Adversary $\mathcal{A}$ outputs $m_0, m_1 \in \mathcal{M}$. $k \leftarrow \mathsf{Gen}$, $b \in \{0,1\}$ uniformly. $c \leftarrow \mathsf{Enc}_k(m_b)$ is given to $\mathcal{A}$, called challenge ciphertext. $b' \leftarrow \mathcal{A}$. Output is 1 ("success") iff $b' = b$, notated $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1$.

Definition:
<u>Perfect Indistinguishability</u>: An encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with $\mathcal{M}$, such that $\forall \mathcal{A}$, $\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1] = \frac{1}{2}$.

Theorem: Perfect Indistinguishability $\Leftrightarrow$ Perfect Secrecy.

Definition: <u>One-time Pad</u>: Fix $\ell > 0$. $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^\ell$ (binary strings length ℓ). Gen: uniform $k \in \mathcal{K}$. Enc: $c = k \oplus m$, $\oplus$ is bitwise xor. Dec: $m = k \oplus c$.

Theorem: The one-time pad encryption scheme is perfectly secret.

Theorem: In a perfectly secure encryption scheme, $|\mathcal{K}| \geq |\mathcal{M}|$. ($|X|$ denotes magnitude/size of X.)

Theorem: <u>Shannon's Theorem</u>: Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme with $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$. It is perfectly secret iff all $k \in \mathcal{K}$ are chosen with probability $1/|\mathcal{K}|$ by Gen, and $\forall m \in \mathcal{M}, c \in \mathcal{C}$, $\exists! k \in \mathcal{K}$ such that $c = \mathsf{Enc}_k(m)$.

Definition: A cryptographic scheme is <u>(t, ε)-secure</u> if any adversary running for time at most t succeeds in breaking the scheme with probability at most ε.

Definition: <u>PPT</u> (Probabilistic Polynomial Time): An adversary which runs for time at most $p(n)$, where n is the security parameter (length of key), and p is a polynomial.

Definition: <u>$\mathtt{negl}$</u> (Negligible): A function f from the natural numbers to the non-negative real numbers such that for every positive polynomial p there is an $N \in \mathbb{N}$ such that $\forall n > N$, $f(n) < \frac{1}{p(n)}$.

Definition: <u>Secure</u>: A scheme where any PPT adversary succeeds in breaking the scheme with at most negligible probability.

Definition: <u>Probabilistic</u>: An algorithm that can "toss a coin" - access unbiased random bits - as necessary.

Theorem: Let $\mathtt{negl}_1, \mathtt{negl}_2$ be negligible functions, p a polynomial. Then $\mathtt{negl}_1(n) + \mathtt{negl}_2(n)$ and $p(n) \cdot \mathtt{negl}_1(n)$ are both negligible.

Definition: <u>Secure</u>: A scheme for which every PPT adversary $\mathcal{A}$ carrying out an attack of some formally specified type, the probability that $\mathcal{A}$ succeeds is negligible.

Definition: We denote an error from Dec by $\perp$ (bottom), when it is asked to decrypt a non-valid ciphertext.

Definition:
<u>Fixed-length encryption scheme</u>: An encryption scheme such that for a $k \leftarrow \mathsf{Gen}(1^n)$, Enc_k is only defined for messages $m \in \{0,1\}^{\ell(n)}$ (fixed length messages).

Definition: <u>$\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$</u> (Adversarial Indistinguishability Experiment-EAV): , where n is the security parameter, and success is defined as before. However, $\mathcal{A}$ is a PPT adversary, $|m_0| = |m_1|$, but the guessing for $b = b'$ is identical.

Definition: <u>EAV-secure</u> (indistinguishable encryptions in the presence of an eavesdropper): A private key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such than for all PPT adversaries $\mathcal{A}$, for all n, $\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathtt{negl}(n)$, where the probability is taken over the randomness used by $\mathcal{A}$ and the encryption scheme Π. Equivalently: $\left|\Pr[\mathsf{out}_{\mathcal{A}}(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n, 0)) = 1] - [\mathsf{out}_{\mathcal{A}}(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n, 1)) = 1]\right| \leq \mathtt{negl}(n)$.

Theorem: Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be a fixed-length private-key encryption scheme for messages of length ℓ that has indistinguishable encryptions in the presence of an eavesdropper. Then for all PPT adversaries $\mathcal{A}$ and any $i \in \{1, \ldots, \ell\}$, there is a negligible function $\mathtt{negl}$ such that $\Pr[\mathcal{A}(1^n, \mathsf{Enc}_k(m)) = m^i] \leq \frac{1}{2} + \mathtt{negl}(n)$, where m^i is the i^{th} bit of m.

Theorem: Let $(\mathsf{Enc}, \mathsf{Dec})$ be a fixed-length private key encryption scheme for messages of length ℓ that is EAV-secure. Then for any PPT algorithm $\mathcal{A}$ there is a PPT algorithm $\mathcal{A}'$ such that for any $S \subseteq \{0,1\}^\ell$ and any function $f : \{0,1\}^\ell \to \{0,1\}$, $|\Pr[\mathcal{A}(1^n, \mathsf{Enc}_k(m)) = f(m)] - \Pr[\mathcal{A}'(1^n) = f(m)]| \leq \mathtt{negl}(n)$. That is, $\mathcal{A}$ cannot determine any function f of the original message m, given the ciphertext, with more than negligible probability better than when not given the ciphertext.

Definition: <u>PRG</u> (Pseudo-Random Generator): Let ℓ be a polynomial and G be a deterministic polynomial-time algorithm such that for any n and any input $s \in \{0,1\}^n$, the result $G(s)$ is a string of length $\ell(n)$. The following must hold: For every n, $\ell(n) > n$. For any PPT algorithm D, there is a negligible function $\mathtt{negl}$ such that $|\Pr[D(G(s)) = 1] - \Pr[D(r) = 1]| \leq \mathtt{negl}(n)$. ℓ is the <u>expansion factor</u> of G.

Construction: <u>Stream Cipher:</u> Let G be a pseudorandom generator with expansion factor ℓ. Let $\mathsf{Gen}(1^n)$ output a uniform $k \in \{0,1\}^n$. Let $c = \mathsf{Enc}_k(m) = G(k) \oplus m$. Let $\mathsf{Dec}_k(c) = G(k) \oplus c$. This is an EAV-secure private-key encryption scheme.

Definition: <u>$\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$</u>: The EAV experiment, except $\mathcal{A}$ presents 2 equal length lists of equal length messages, $\vec{M}_0 = (m_{0,1}, \ldots, m_{0,t})$ and $\vec{M}_1 = (m_{1,1}, \ldots, m_{1,t})$, the challenger chooses one of the lists and returns the ciphertext of all messages from that list, and $\mathcal{A}$ attempts to determine which list was chosen.

Definition: <u>Multiple-EAV-Secure</u>: Same as EAV secure, except with the $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$ experiment.

Theorem: There are private-key encryption schemes which are EAV-secure but not multiple-EAV-secure.

Theorem: Any multiple-EAV-secure private-key encryption scheme is also EAV-secure.

Theorem: If Π is a stateless encryption scheme in which Enc is deterministic, then Π cannot be multiple-EAV-secure.

Definition: <u>$\mathsf{PrivK}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(n)$</u>: $k \leftarrow \mathsf{Gen}(1^n)$, then adversary $\mathcal{A}$ is given 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$. $\mathcal{A}$, after making its oracle calls, outputs m_0, m_1, a pair of same length messages. b is chosen, $c = \mathsf{Enc}_k(m_b)$ is computed and returned, and $\mathcal{A}$ outputs b'. $\mathcal{A}$ "succeeds" if $b' = b$, and the experiment outputs 1. Else, the experiment outputs 0.

Definition: <u>CPA-secure</u> (Chosen Plaintext Attack): A private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that for all PPT adversaries $\mathcal{A}$, $\Pr\left[\mathsf{PrivK}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathtt{negl}(n)$.

Definition: <u>$\mathsf{PrivK}^{\mathsf{LR-CPA}}_{\mathcal{A},\Pi}(n)$</u>: Same as $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$, but for CPA-security. Extends to <u>Multiple-CPA-security</u>.

Theorem: CPA-secure $\Rightarrow$ multiple-CPA-secure.

Definition: <u>Keyed Function</u>: A function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$, with first input called key k. It is <u>efficient</u> if there is a polynomial-time algorithm that computes $F(k, x)$ given k and x.

Definition: <u>Length-Preserving</u>: A keyed function such that $\ell_{key}(n) = \ell_{in}(n) = \ell_{out}(n)$.

Definition: <u>$\mathtt{Func}_n$</u>: The set of all functions mapping n-bit strings to n-bit strings. $|\mathtt{Func}_n| = 2^{n \cdot 2^n}$.

Definition: <u>PRF</u> (Pseudo-Random Function): An efficient, length-preserving keyed function such that for all PPT distinguishers D, $\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathtt{negl}(n)$, where f is chosen uniformly from $\mathtt{Func}_n$. $|\text{PRF}| = 2^n$, there are at most that many distinct functions. (Some are not secure.)

Definition: <u>Permutation</u>: A keyed function F such that $\ell_{in} = \ell_{out}$, and for all $k \in \mathcal{K}$, $F_k\{0,1\}^{\ell_{in}(n)} \to \{0,1\}^{\ell_{out}(n)}$ is one-to-one. F_k is <u>efficient</u> if $F_k(x)$ and $F_k^{-1}(x)$ are computable with a polynomial-time algorithm.

Definition: <u>PRP</u> (Pseudo-Random Permutation): Same as a PRF, except F must be indistinguishable from a random $f \in \mathtt{Perm}_n$, the set of truly random permutations.

Theorem: If F is a PRP and $\ell_{in}(n) \geq n$, F is a PRF.

Definition: <u>Strong PRP</u>: Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed permutation such that for all PPT distinguishers D, $\left|\Pr[D^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot), f^{-1}(\cdot)}(1^n) = 1]\right| \leq \mathtt{negl}(n)$, where $f \in \mathtt{Perm}_n$ uniformly. Any strong PRP is a PRP.

Definition: <u>Synchronized</u>: Stream cipher mode where sender and receiver must know how much plaintext has been encrypted/decrypted so far. Typically used in a single session between parties.

Definition: <u>Unsynchronized</u>: Stream cipher mode which is stateless, taking a new IV each time.

Definition: <u>ECB Mode</u> (Electronic Code Book): Here, $c := \langle F_k(m_1), F_k(m_2), \ldots, F_k(m_\ell) \rangle$, where $m = m_1, m_2, \ldots, m_\ell$ is the message and F is a block cipher of length n. Deterministic, and therefore not CPA-secure. Should not be used, only included for historical significance.

Definition: <u>CBC Mode</u> (Cipher Block Chaining): Choose an IV of length n. Then, $c_0 = IV$, $c_1 = F_k(c_0 \oplus m_1)$, $c_2 = F_k(c_1 \oplus m_2)$, and so on. To decrypt, compute $m_\ell = F_k^{-1}(c_\ell) \oplus c_{\ell-1}$, $m_{\ell-1} = F_k^{-1}(c_{\ell-1}) \oplus c_{\ell-2}$, and so on. If F is a PRP, and IV is chosen uniformly at random, then CBC mode is CPA-secure. It cannot be computed in parallel, since encrypting c_i requires c_{i-1} for $i > 0$. Using c_ℓ as IV for the next encryption is <u>not secure</u>.

Definition: <u>OFB Mode</u> (Output FeedBack): Let IV be uniformly chosen of length n. Then $c_0 = IV$, $c_1 = F_k(IV) \oplus m_1$, $c_2 = F_k(F_k(IV)) \oplus m_2$, $\ldots c_\ell = F_k^\ell(IV) \oplus m_\ell$, where F_k^ℓ denotes F_k applied ℓ times. F need not be invertible, and m_ℓ need not be of length n, the message may be truncated to match its length. OFB mode is CPA-secure. Using $F_k^\ell(IV)$ as the next IV, producing a synchronized stream cipher, it remains secure.

Definition: <u>CTR Mode</u> (CounTeR): Pick an $\mathsf{ctr} = IV$, then $c_0 = \mathsf{ctr}$, $c_1 = F_k(\mathsf{ctr} + 1) \oplus m_1$, $\ldots$, $c_\ell = F_k(\mathsf{ctr} + \ell) \oplus m_\ell$. F need not be invertible. Here, the encryption can be fully parallelized. CTR mode is CPA-secure, assuming F is a PRF. The stateful variant, where $F_k(\mathsf{ctr} + \ell)$ is used as the new IV, remains secure.

Note: None of these schemes achieve message integrity in the sense of chapter 4.

Definition: <u>$\mathsf{PrivK}^{\mathsf{CCA}}_{\mathcal{A},\Pi}(n)$</u>: The adversary $\mathcal{A}$ is given access to a decryption oracle in addition to an encryption oracle, then outputs m_0, m_1, gets $c := \mathsf{Enc}_k(m_b)$, the challenge ciphertext, and tries to determine b'. $\mathcal{A}$ again has oracle access, but cannot query the decryption oracle with c. Success is as defined in previous experiments.

Definition: <u>CCA-Secure</u> (Chosen Ciphertext Attack): A private-key encryption scheme Π such that for all PPT adversaries $\mathcal{A}$, $\Pr[\mathsf{PrivK}^{\mathsf{CCA}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathtt{negl}(n)$. Any CCA-secure scheme is also multiple-CCA-secure.

Note: NOTHING above this point is CCA-secure.

Definition: <u>Non-Malleability</u>: An encryption scheme with the property that if the adversary tries to modify a given ciphertext, the result is either an invalid ciphertext or one whose corresponding plaintext has no relation to the original plaintext.

Definition: <u>MAC</u> (Message Authentication Code): Three probabilistic polynomial-time algorithms $\mathsf{GenMacVrfy}$ such that Gen takes 1^n, outputs k with $|k| \geq n$. Mac, the tag-generation algorithm, takes k and $m \in \{0,1\}^*$ and outputs tag t. Deterministic Vrfy takes k, m, t, and outputs b, where $b = 1$ means t is a valid tag for m with key k, and $b = 0$ means it is not. It must be that $\mathsf{Vrfy}_k(m, \mathsf{Mac}_k(m)) = 1$. If Mac_k is only defined for $m \in \{0,1\}^{\ell(n)}$, we call it a fixed-length MAC.

Definition: <u>Canonical Verification</u>: Deterministic MACs (Mac is deterministic), where $\mathsf{Vrfy}_k(m, t)$ computes $\tilde{t} := \mathsf{Mac}_k(m)$, and out puts 1 iff $\tilde{t} = t$.

Definition: <u>$\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)$</u>: $k \leftarrow \mathsf{Gen}(1^n)$. $\mathcal{A}$ is given 1^n and oracle access to $\mathsf{Mac}_k(\cdot)$. Eventually outputs (m, t). Success is defined as $\mathsf{Vrfy}(m, t) = 1$, and m is not a message previously queried from the oracle.

Definition: <u>Secure MAC</u> (Existentially Unforgeable Under an Adaptive Chosen-Message Attack): A MAC such that for all PPT adversaries $\mathcal{A}$, $\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathtt{negl}(n)$. Note: This definition offers no protection against replay attacks.

Definition: <u>Strong MAC</u>: MAC such that (m, t) cannot have been previously output by the oracle, but using (m, t') is a valid guess.

Theorem: With a canonical Vrfy, Strong Mac $\Leftrightarrow$ Secure MAC.

Construction: : $\mathsf{Mac}_k(m) = F_k(m)$, where $k \in \{0,1\}^n$, $m \in \{0,1\}^n$, and F a PRF. Vrfy is canonical. If $|m| \neq |k|$, Mac outputs nothing, and Vrfy outputs 0. This construction is a secure fixed-length MAC.

Construction: Another Mac_k: Chop m into n-bit blocks, $m_1, m_2, \ldots, m_d$, let $t_i = \mathsf{Mac}'(m_i)$, and use $(t_1, t_2, \ldots, t_d)$ as the tag.
This is bad. This can be easily broken using a reordering attack. Present $m = m_1, m_2$, get tag t_1, t_2. Then message $m' = m_2, m_1$ will have tag t_2, t_1, which will pass Vrfy.
To combat this attack, break m into $\frac{n}{2}$-bit blocks $m_1, \ldots, m_d$, then $t_i = \mathsf{Mac}'_k(\langle i \rangle \mid\mid m_i)$, where $\langle i \rangle$ is the $\frac{n}{2}$-bit binary encoding of i. This prevents the reordering attack.
This scheme is still insecure. Since we have an arbitrary-length message Mac, we can use a truncation attack, and present $m = m_1, m_2, m_3$, get (t_1, t_2, t_3). Then we can present $m' = m_1, m_2$. The tag (t_1, t_2) will be valid for m'.
To prevent the truncation attack, we will include the length ℓ of the full message in the calculation. We will chop our message into $\frac{n}{3}$-bit blocks. Then $t_i = \mathsf{Mac}'_k(\langle \ell \rangle \mid\mid \langle i \rangle \mid\mid m_i)$. Note: We pad the last block with $0's$ if necessary. The tag will be $(t_1, \ldots)$. By this point, we are sending 4ℓ bits.
Unfortunately, even this scheme is still insecure. It can be attacked with a "mix and match" attack. For example, get tag $t = (t_1, t_2, t_3)$ for $m = m_1, m_2, m_3$. Take another message, same length, $m' = m_4, m_5, m_6$, get tag $t' = (t_4, t_5, t_6)$. Then (t_1, t_5, t_6) is a valid tag for m_1, m_5, m_6, which has never been queried from the oracle before.
Finally, let's fix all of this! We'll chop our message $m = m_1, \ldots, m_d$ into $\frac{n}{4}$ bit blocks, and pick a random $\frac{n}{4}$-bit value r for the entire message, and $t_i = \mathsf{Mac}'_k(r \mid\mid \langle \ell \rangle \mid\mid \langle i \rangle \mid\mid m_i)$. $\mathsf{Mac}_k(m) = (r, t_1, \ldots, t_d)$. At this point, this is not a deterministic Mac, so Vrfy has to behave slightly differently, taking into account the random r passed to it. It can reconstruct the tag as above, with this slight extra step.
This is secure!
But it does produce a tag of 4x the length of the message.

Construction: <u>CBC-MAC</u>: Used widely in practice. On input a key $k \in \{0,1\}^n$, m of length $\ell(n) \cdot n$, let $\ell = \ell(n)$, parse $m = m_1, \ldots, m_\ell$, set $t_0 := 0^n$, then for $i \in \{1, \ell\}$, $t_i := F_k(t_i - 1)$, where F is a PRF. Output t_ℓ only as the tag. Vrfy is done in the canonical way.
To extend this to arbitrary length messages, *prepend* the message with length $|m|$, encoded as an n-bit string.
Alternatively, have keys k_1, k_2, compute CBC-MAC using k_1, then output tag $\hat{t} := F_{k_2}(t)$.

Definition: <u>$\mathsf{Enc\text{-}Forge}_{\mathcal{A},\Pi}(n)$</u>: Run $\mathsf{Gen}(1^n)$ to obtain k. Adversary $\mathcal{A}$ is given input 1^n and access to an encryption oracle $\mathsf{Enc}_k(\cdot)$. They output ciphertext c. Let $m := \mathsf{Dec}_k(c)$, and let Q denote the set of all queries that $\mathcal{A}$ asked its encryption oracle. The output of the experiment is 1 iff $m \neq \perp$ and $m \notin Q$.

Definition: <u>Unforgeable</u>: A private-key encryption scheme Π such that for all PPT adversaries $\mathcal{A}$, $\Pr[\mathsf{Enc\text{-}Forge}_{\mathcal{A},\Pi}(n) = 1] \leq \mathtt{negl}(n)$.

Definition: <u>Authenticated</u>: A private-key encryption scheme that is CCA-secure and unforgeable.

Construction: <u>Encrypt-and-authenticate</u>: Given plaintext m, sender transmits $\langle c, t \rangle$, where $c \leftarrow \mathsf{Enc}_{k_E}(m)$ and $t \leftarrow \mathsf{Mac}_{k_M}(m)$. The receiver behaves as expected, obtaining m from $\mathsf{Dec}_{k_E}(c)$, and running $\mathsf{Vrfy}_{k_M}(m, t)$. It is likely the case here that t leaks information about the message (often, MACs are deterministic, breaking CPA-security), and so this is <u>not</u> an authenticated encryption scheme.

Construction: <u>Authenticate-then-encrypt</u>: Given plaintext m, sender transmits c, where $t \leftarrow \mathsf{Mac}_{k_M}(m)$ and $c \leftarrow \mathsf{Enc}_{k_E}(m||t)$. The receiver behaves as expected, decrypting $m||t$ from c, then checking $\mathsf{Vrfy}_{k_M}(m, t)$. If, for example, a CBC-mode-with-padding scheme is used, the decrypt algorithm will return a "bad padding" error, while if the padding passes, Vrfy will return an "authentication failure". This difference can leak information and allow for various attacks on the scheme, so this is <u>not</u> an authenticated encryption scheme.

Construction: <u>Encrypt-then-authenticate</u>: Given plaintext m, sender transmits $\langle c, t \rangle$, where $c \leftarrow \mathsf{Enc}_{k_E}(m)$ and $t \leftarrow \mathsf{Mac}_{k_M}(m)$. The receiver behaves as expected, checking $\mathsf{Vrfy}_{k_M}(c, t)$, then decrypting m as $\mathsf{Dec}_{k_E}(c)$. Of the three listed, this is the only one that <u>is</u> an authenticated encryption scheme (Assuming that Enc is CPA-secure, Mac is strongly secure, and k_E and k_M are chosen independently uniformly at random.)

There are 3 major types of network attacker attacks.
In a <u>reordering attack</u>, an attacker swaps the order of messages sent across a network, making c_2 arrive before c_1.
In a <u>replay attack</u>, an attacker resends messages later.
In a <u>reflection attack</u>, an attacker sends messages from a sender back to them at a later time, which the other person never sent.
The first two attacks can be prevented when A and B (the two people communicating across the network) keep counters, $\mathsf{ctr}_{A,B}$ and $\mathsf{ctr}_{B,A}$, of how many messages have been sent/received in each direction.
A reflection attack can either be prevented by having a reflection bit b to say who the sender is, or by having a different key-set for messages going different directions.
In the $\mathsf{Mac\text{-}forge}^{\text{1-time}}_{\mathcal{A},\Pi}$ experiment, adversary $\mathcal{A}$ outputs m', is given a tag $t' \leftarrow \mathsf{Mac}_k(m')$, then can calculate and think, then output (m, t), $m \neq m'$, which are verified as usual to determine success.

Definition: <u>ε-secure</u> (also one-time ε-secure): A MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ such that for all (even unbounded) adversaries $\mathcal{A}$, $\Pr[\mathsf{Mac\text{-}forge}^{\text{1-time}}_{\mathcal{A},\Pi} = 1] \leq \varepsilon$.

Definition: <u>Strongly universal</u>: A function $h : \mathcal{K} \times \mathcal{K} \to \mathcal{T}$ such that for all distinct $m, m' \in \mathcal{M}$, and all $t, t' \in \mathcal{T}$, it holds that $\Pr[h_k(m) = t \wedge h_k(m1) = t'] = \frac{1}{|\mathcal{T}|^2}$, where the probability is taken over uniform choice of $k \in \mathcal{K}$.

Construction: Let $h : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be a strongly universal function. Define a MAC as follows: Gen: uniform $k \in \mathcal{K}$. Mac: given k, m, output tag $t := h_k(m)$. Vrfy: On input k, m, t, output 1 iff $t \stackrel{?}{=} h_k(m)$.

Theorem: If h is a strongly universal function, then the above construction is a $\frac{1}{|\mathcal{T}|}$-secure MAC for messages in $\mathcal{M}$.

Theorem: For any prime p, the function h defined as $h_{a,b}(m) = [a \cdot m + b \mod p]$, where $\mathcal{M} = \mathbb{Z}_p$, and $\mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$, so $(a, b) \in \mathcal{K}$, $m \in \mathcal{M}$, is strongly universal.

Definition: Hash function: A function with output length ℓ is a pair of PPT algorithms (Gen, H) such that $\mathsf{Gen}(1^n)$ outputs a key s, and H takes s and a string $x \in \{0,1\}^*$, and outputs a string $H^s(x) \in \{0,1\}^n$, assuming n is implicit in s.
Definition: Compression function (fixed-length hash function for inputs of length ℓ'): a hash function where H^s is only defined for inputs $x \in \{0,1\}^{\ell'(n)}$, and $\ell'(n) > \ell(n)$.
Definition: $\mathsf{Hash\text{-}Coll}_{\mathcal{A},\Pi}(n)$: $s \leftarrow \mathsf{Gen}(1^n)$. Adversary $\mathcal{A}$is given s and outputs x, x'. (If Π is fixed-length, then $x, x' \in \{0,1\}^{\ell'(n)}$.) The output is 1 (success) iff $x \neq x'$ but $H^s(x) = H^s(x')$.
Definition: Collision resistant: A has function $\Pi = (\mathsf{Gen}, H)$ such that for all PPT adversaries $\mathcal{A}$, $\Pr[\mathsf{Hash\text{-}Coll}_{\mathcal{A},\Pi}(n) = 1] \leq \mathtt{negl}(n)$.
Definition: Second-preimage resistance (target-collision resistance): A hash function such that given s and x, an adversary cannot find x' such that $x' \neq x$ and $H^s(x) \neq H^s(x')$.
Definition: Preimage resistance: A hash function such that given s and y, an adversary cannot find x such that $H^s(x) = y$.
Construction: Merkle-Damgård: Let (Gen, h) be a fixed-length hash function for inputs of length $2n$ and with output length n. Construct (Gen, H) as follows: $\mathsf{Gen} = \mathsf{Gen}$, H: given s and $x \in \{0,1\}^*$ of length $L < 2^n$, let $B = \left\lceil \frac{L}{n} \right\rceil$, pad x so its length is a multiple of n. Consider the padded result as n-bit blocks $x_1, \ldots, x_B$. Set $x_{B+1} = L$. Set $z_0 = 0^n$, as the IV. For $i = 1, \ldots, B+1$, let $z_i = h^s(z_{i-1}||x_i)$. Output z_{B+1}.
Theorem: If (Gen, h) is collision resistant, then so is (Gen, H).
Construction: Hash-and-MAC: Let $\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$ be a MAC for length $\ell(n)$, let $\Pi_H(\mathsf{Gen}_H, H)$ be a hash function, with output length $\ell(n)$. Construct MAC $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$ as follows: Gen': Takes 1^n, choses uniform $k \in \{0,1\}^n$, $s \leftarrow \mathsf{Gen}_H(1^n)$, outputs key $k' = \langle k, s \rangle$. Mac': Given $\langle k, s \rangle$, $m \in \{0,1\}^*$, output $t \leftarrow \mathsf{Mac}_k(H^s(m))$. Vrfy': Given $\langle k, s \rangle$, $m \in \{0,1\}^*$, tag t, output 1 iff $\mathsf{Vrfy}_k(H^s(m), t) = 1$.
Theorem: If Π is a secure MAC and Π_H is collision resistant, the above construction is a secure MAC for arbitrary-length messages.
Construction: HMAC: Let (Gen_H, H) be a Merkle-Damgård-generated hash function on (Gen_H, h) taking inputs of length $n + n'$. Let $\mathsf{opad}$and ipadbe fixed constants of length n'. Define a MAC as follows: Gen: Given 1^n, $s \leftarrow \mathsf{Gen}_H(1^n)$, uniform random $k \in \{0,1\}^{n'}$. Output key $\langle s, k \rangle$. Mac: Given $\langle s, k \rangle$ and $m \in \{0,1\}^*$, output $t := H^s((k \oplus \mathsf{opad})||H^s((k \oplus \mathsf{ipad})||m))$. Vrfy: Given $\langle s, k \rangle$, $m \in \{0,1\}^*$, tag t, output 1 iff t recomputes correctly.
Definition: Weakly collision resistant: A Hash function (Gen_H, H) defined as a Merkle-Damgård transform, except with $k = IV$ being uniformly chosen from $\{0,1\}^n$, such that every PPT adversary $\mathcal{A}$ has at most negligible success finding a collision (without knowing k.).
Theorem: Let $k_{out} = h^s(IV||(k \oplus \mathsf{opad}))$, $\hat{y}$ be the length-padded y, including anything before it, $\widetilde{\mathsf{Mac}}_k(y) = h^s(k||\hat{y})$, and $G^s(k) = h^s(IV||(k \oplus \mathsf{opad}))||h^s(IV||(k \oplus \mathsf{ipad})) = k_{out}||k_{in}$. If G^s is a PRG for any s, $\widetilde{\mathsf{Mac}}_k(y)$ is a secure fixed-length mac for messages of length n, and (Gen_H, H) is weakly collision resistant, then HMAC is a secure MAC for arbitrary-length messages.
Definition: Birthday problem/attack: Out of n distinct "days", if $\sqrt{n}$ "people" are chosen, there is a 50% chance that two of them will share a birthday. This places a lower bound of $2\log(T)$ bits on the size of a hash function, where T is the time we want to run the collision-attack in.
Construction: Birthday Attack: (small space). Start with random valid input x_0, then repeatedly compute $x_i = H(x_{i-1})$ and $x_{2i} = H(H(x_{2(i-1)}))$. If they are ever equal, collision has occurred in $x_0, \ldots, x_{2(i-1)}$. Calculate each x_j, x_{j+i}, and we will find a collision. This runs in $\Theta(2\ell/2)$ time, where ℓ is the length of the output.
Definition: Random Oracle Model: Model in which the oracle O chooses its function H at random when instantiated, and so probabilities are also taken over the choice of the function H. It is then used in whichever function needed. The values of H are considered to be computed the first time they are requested.
Construction: If $\ell_{out} > \ell_{in}$, a random oracle can be used as a pseudorandom generator. If $\ell_{out} < \ell_{in}$, a random oracle is collision resistant. If $F_k(x) = H(k||x)$, where $|k| = |x| = n$, $\ell_{out} = n$, $\ell_{in} = 2n$, then F is a pseudorandom function, where H is the pseudorandom oracle.
In general, a proof of security in the random oracle model is significantly better than no proof at all. There have been no successful real-world attacks on schemes proven secure in the random-oracle model, when the random oracle was instantiated properly. Most cryptographic hash functions should not be used "off the shelf" to instantiate a random oracle model.
Definition: Virus Fingerprinting: Process by which a virus scanner stores hashes of known viruses, and compares hashes of email attachments and newly downloaded programs to these known viruses.
Definition: Deduplication: Process of comparing hashes of new files to hashes of already stored files to eliminate the storing of duplicates. Especially used in the context of cloud storage among multiple users.
Definition: P2P file-sharing: Processing of storing hashes of available files, allowing for easy requests, etc.
Definition: Merkle-Damgård Tree: A tree constructed from 2^t by placing a file at each leaf of a t-level tree, then computing the hash of each pair of files, then each pair of hashes, and so on, until a single root hash is computed. This hash is then stored. Often denoted $\mathcal{MT}_t$.
Theorem: Let (Gen_H, H) be collision resistant. Then $(\mathsf{Gen}_H, \mathcal{MT}_t)$ is also collision resistant for any fixed t.
Construction: Password Hashing: On a computer, the hash of a password, hpw, will be stored, and when the user enters their password, $H(pass) \stackrel{?}{=} hpw$, if so, authenticated. To prevent dictionary attacks, sometimes a salt is used to calculate $H(s, pass)$.
Definition: min-entropy: A probability distribution $\mathcal{X}$ has m bits of min-entropy if for every fixed value x it holds that $\Pr_{X \leftarrow \mathcal{X}}[X = x] \leq 2^{-m}$. That is, even the most likely outcome occurs with probability at most 2^{-m}.
Definition: LFSR: Linear Feedback Shift Register. Very efficient to implement in hardware. Consists of an array of n registers, $s_{n-1}, \ldots, s_0$ with n feedback coefficients, $c_{n-1}, \ldots, c_0$. The size of the array is called the degree of the LFSR. On each clock tick, s_0 is the output bit, all bits are shifted right by 1 register, and s_{n-1} is set to the XOR of some subset of the other registers, defined as those where $c_i = 1$.
These are insecure because the initial state, feedback coefficients, and all future bits, can be determined from watching at most $2n$ consecutive bits of output. We can improve the security by adding non-linear combinations to compute s_{n-1}, and it is possible to define such functions with good statistical properties. Trivium is one such stream cipher.
Definition: RC4: a similar algorithm that also involves bit swaps, is used in many security situations, but is known to have vulnerabilities.
Definition: Avalanche Effect: In any block cipher, a "small change" to the input must affect every bit of the output.
Definition: Feistel Network: A network which operates in a series of rounds. In each round, a keyed round function is applied. This function need not be invertible. In a balanced Feistel Network, the ith round of function $\hat{f}_i$ takes as input a sub-key x_i and an $\ell/2$-bit string and outputs an $\ell/2$-bit string.
Construction: Feistel Network: For round i of a Feistel network, divide the input into two halves, L_{i-1} and R_{i-1}, each of length $\ell/2$, where ℓ is the block length of the cipher. The output (L_i, R_i) is defined as $L_i := R_{i-1}, R_1 = L_{i-1} \oplus f_i(R_{i-1})$. In an r-round Feistel network, the ℓ-bit input becomes (L_0, R_0), and the output is the ℓ-bit value (L_r, R_r).
Theorem: Let F be a keyed function defined by a Feistel Network. Then regardless of the round functions $\{\hat{f}_i\}$, and the number of rounds, F_k is an efficiently invertible permutation for all k.
Definition: DES: Data Encryption Standard. Originally a 16-round Feistel Network with a block length of 64 bits and a key length of 56 bits. Vulnerable to brute-force attacks, but the strengthened triple-DES is widely used today.
Construction: DES $\hat{f}$. $\hat{f}(k_i, R)$, with $k_i \in \{0,1\}^{28}$, $R \in \{0,1\}^{32}$, R is expanded to 48-bit R' by duplicating the first half of the bits, $R' \oplus k_i$ is computed, split and passed through an S-box which takes 6-bit inputs to 4-bit outputs, and these 4-bit outputs are mixed to produce the 32-bit output. DES uses 16 rounds. The S-boxes were carefully designed to be 4-to-1 functions, and changing any 1 bit of the input changes at least 2 bits of the output. The mixing was designed that the output from any S-box affects the input to six of the S-boxes in the next round. Therefore, the mangle function exhibits a strong avalanche effect, which will, after 8 rounds, affect all 64 bits of output. Since DES uses 16 rounds total, similar inputs yields independent-looking outputs. After 30 years, the best known practical attack on DES is still an exhaustive search through its key space. The 56-bit key length is such that such an attack *is* feasible.
Definition: Triple Encryption: Using 3 keys, $F'_{k_1,k_2,k_3}(x) := F_{k_3}(F^{-1}_{k_2}(F_{k_1}(x)))$. Using only 2, $F'_{k_1,k_2}(x) := F_{k_1}(F^{-1}_{k_2}(F_{k_1}(x)))$. Triple-DES is constructed using DES with either of these variants.
Definition: AES: Advanced Encryption Standard. Has a state array, which is a 4x4 array. Initially the input of the cipher. Then, consists of 4 stages. Stage 1: Add Round Key: A 128-bit sub-key is derived from the master key, and interpreted as a 4x4 array of bytes. The state array is XORed with the sub-key. Stage 2: Sub Bytes: The state array is mixed at the byte level according to a fixed lookup table S. Stage 3: Shift Rows: The bytes in each row are shifted to the left by 0, 1, 2, and 3 places respectively, from the top. Stage 4: Mix Columns: An invertible transformation is applied to each column. It has the property that if the inputs differ in $b > 0$ bytes, the outputs differ in at least $5 - b$ bytes. In the final round, Mix Columns is replaced with AddRoundKey. To date, there have been no practical attacks significantly better than an exhaustive key-search, so AES is an excellent choice for any cryptographic scheme that requires a (strong) pseudorandom permutation.
Definition: Ideal Cipher Model: A strengthening of the random-oracle model in which all parties have access to an oracle for a random keyed permutation $F: \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^\ell$ and F^{-1}.
Construction: Davies-Meyer: Let F be a block cipher with n-bit key length and ℓ-bit block length. The compression function is $h: \{0,1\}^{n+\ell} \to \{0,1\}^\ell$ by $h(k,x) := F_k(x) \oplus x$.
Theorem: If F is an ideal cipher, then Davies-Meyer yields a collision-resistant compression function. From the HW, $F_k(x) \oplus x \oplus k$ also does, but $F_k(x)$ and $F_k(x) \oplus k$ do not.
Care must be taken when instantiating Davies-Meyer with a particular block cipher, for example, DES causes issues.
MD5 is bad and should not be used, SHA-0 is flawed, SHA-1 has known slight flaws that make it easier to theoretically crack, but has not produced any collisions. It is not recommended for use, SHA-2 seems to be secure, and can be used. SHA-3 is rather powerful, and unusual, and considered very, very secure.
Definition: $\mathsf{Invert}_{\mathcal{A},\Pi}\mathcal{A}f$: Uniform $x \in \{0,1\}^n$, $y := f(x)$. $\mathcal{A}$ is given 1^n and y, outputs x'. Outcome 1 iff $f(x') = y$.
Definition: One-way: A function $f: \{0,1\}^* \to \{0,1\}^*$ such that there is a polynomial-time algo M_f that computes f, and for every PPT adversary $\mathcal{A}$, $\Pr[\mathsf{Invert}_{\mathcal{A},\Pi}\mathcal{A}f(n) = 1] \leq \mathtt{negl}(n)$.
Definition: Hard-core predicate: A function $\mathtt{hc}: \{0,1\}^* \to \{0,1\}$ for f such that $\mathtt{hc}$ can be computer in polynomial time, and for all PPT adversaries $\mathcal{A}$, $\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(1^n, f(x)) = \mathtt{hc}(x)] \leq \frac{1}{2} + \mathtt{negl}(n)$.
Theorem: Goldreich-Levin: Assume one-way functions (resp., permutations) exist. Then $\exists$ a one-way function (resp., permutation) g and a hard-core predicate $\mathtt{hc}$ of g.
Theorem: Let f be a one-way permutation and let $\mathtt{hc}$ be a hard-core predicate of f. Then $G(s) := f(s)||\mathtt{hc}(s)$ is a PRG with expansion factor $\ell(n) = n + 1$.
Theorem: If $\exists$ a PRG with expansion factor $\ell(n) = n + 1$, then for any polynomial $\mathtt{poly}$, $\exists$ a PRG with expansion factor $\mathtt{poly}(n)$.
Theorem: If $\exists$ a PRG with expansion factor $\ell(n) = 2n$, then there exists a PRF.
Theorem: If $\exists$ a PRF, then $\exists$ a strong PRP.
Theorem: Assuming the existence of one-way permutations, $\exists$ PRGs with any polynomial expansion factor, PRFs, and strong PRPs.
Theorem: Assuming the existence of one-way permutations, $\exists$ CCA-secure private-key encryption schemes and secure message authentication codes.
Theorem: Let f be a one-way function and define $g(x, r) := (f(x), r)$, where $|x| = |r|$. Define $\mathtt{gl}(x, r) := \oplus_{i=1}^n x_i \cdot r_i$, where $x = x_1 \cdots x_n$, and $r = \cdots r_n$. Then $\mathtt{gl}$ is a hard-core predicate of g.
Theorem: Let f and $\mathtt{gl}$ be as above. If $\exists$ a PPT $\mathcal{A}$ such that $\mathcal{A}(f(x), r) = \mathtt{gl}(x, r)$ $\forall n$ and $\forall x, r \in \{0,1\}^n$, then $\exists$ PPT $\mathcal{A}'$ such that $\mathcal{A}'(1^n, f(x)) = x$ $\forall n$ and $\forall x \in \{0,1\}^n$.
Theorem: Let f and $\mathtt{gl}$ be as above. If $\exists$ a PPT $\mathcal{A}$ and polynomial $p(\cdot)$ such that $\Pr_{x, r \leftarrow \{0,1\}^n}[\mathcal{A}(f(x), r) = \mathtt{gl}(x, r)] \geq \frac{3}{4} + \frac{1}{p(n)}$ for infinitely many values of n, then $\exists$ a PPT $\mathcal{A}$ such that $\Pr_{x \leftarrow \{0,1\}^n}\left[\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))\right] \geq \frac{1}{4p(n)}$.
Theorem: Let f be a one-way permutation with hard-core predicate $\mathtt{hc}$. Then algorithm $G(s) := f(s)||hc(s)$ is a PRG with expansion factor $\ell(n) = n + 1$.
Theorem: If $\exists$ a PRG G with expansion factor $n+1$, for any polynomial $\mathtt{poly}$ $\exists$ a PRG $\hat{G}$ with expansion factor $\mathtt{poly}(n)$.
Construction: Let G be a PRG with expansion factor $\ell(n) = 2n$, and define G_0, G_1 as $G(k) = G_0(k)||G_1(k)$, where $|G_0(k)| = |G_1(k)| = |k|$. For $x \in \{0,1\}^n$, define $F_k: \{0,1\}^n \to \{0,1\}^n$ as $F_k(x_1 x_2 \ldots x_n) = G_{x_n}(\ldots(G_{x_2}(G_{x_1}(x))))$.
Theorem: The above construction is a PRF.
Theorem: If one-way functions exist, then so do PRGs, PRFs, and strong PRPs.
Theorem: If one-way functions exist, then so do CCA-secure private-key encryption schemes and secure message authentication codes.
One-way functions are sufficient for all private-key cryptography.
Theorem: If a PRG exists, then so does a one-way function.
Theorem: If there exists an EAV-secure private-key encryption scheme that encrypts messages twice as long as its key, then a one-way function exists.
Theorem: MACs also imply that one-way functions exist.
Theorem: Let $a \in \mathbb{N}$ and let $b \in \mathbb{N}^+$. Then $\exists$ unique $q, r \in \mathbb{N}$ such that $a = qb + r$, and $0 \leq r < b$. These can be computed in polynomial time.
Definition: GCD: Greatest Common Divisor of $a, b \in \mathbb{N}$. Largest $c \in \mathbb{N}$ such that $c \mid a$ and $c \mid b$. Notation: $c = \gcd(a, b)$.
Theorem: Let $a, b \in \mathbb{N}^+$. $\exists$ X, Y such that $Xa + Yb = \gcd(a, b)$. $\gcd(a, b)$ is the smallest positive integer that can be expressed this way.
Theorem: If $c \mid ab$ and $\gcd(a, c) = 1$, then $c \mid b$. If p prime and $p \mid ab$, then either $p \mid a$ or $p \mid b$.
Theorem: If $a \mid N$, $b \mid N$, and $\gcd(a, b) = 1$, then $ab \mid N$.
Definition: mod: $a \equiv b \mod N$ iff $N \mid a - b$. This also means that $[a \mod N] = [b \mod N]$, reduction mod N.
Theorem: If $a = a' \mod N$ and $b = b' \mod N$, then $(a + b) = (a' + b') \mod N$ and $ab = a'b' \mod N$.
Definition: Invertible: Given $b, n \in \mathbb{N}$, b, $\exists c$ such that $bc = 1 \mod N$.
Theorem: Let $b, N \in \mathbb{N}$, $b \geq 1$, $N > 1$. Then b is invertible $\mod N$ iff $\gcd(b, N) = 1$.
Definition: Group: A set $\mathbb{G}$ with a binary operation $\circ$ such that: 1) $\forall g, h \in \mathbb{G}$, $g \circ h \in \mathbb{G}$. 2) $\exists e \in \mathbb{G}$ such that $\forall g \in \mathbb{G}$, $e \circ g = g = g \circ e$. e is called the identity. 3) $\forall g \in \mathbb{G}$, $\exists h \in \mathbb{G}$ such that $g \circ h = e = h \circ g$. h is called the inverse of g. 4) $\forall g_1, g_2, g_3 \in \mathbb{G}$, $(g_1 \circ g_2) \circ g_3 = g_1(g_2 \circ g_3)$.
Definition: Abelian: A group $\mathbb{G}$ with the additional property that $\forall g, h \in \mathbb{G}$, $g \circ h = h \circ g$.
Theorem: Let $\mathbb{G}$ be a group and $a, b, c \in \mathbb{G}$. If $ac = bc$, then $a = b$. If $ac = c$, then $a = e$.
Definition: Order: The number $m = |\mathbb{G}|$ that is the number of elements in a group, if it is finite.
Theorem: Let $\mathbb{G}$ be a finite group with $m = |\mathbb{G}|$. Then $\forall g \in \mathbb{G}$, $g^m = 1$.
Theorem: Let $\mathbb{G}$ be a finite group with $m = |\mathbb{G}| > 1$. Then $\forall g \in \mathbb{G}$, and any $x \in \mathbb{N}$, $g^x = g[x \mod m]$.
Theorem: Let $\mathbb{G}$ be a finite group with $m = |G| > 1$. Let $e > 0$ be an integer, and define the function $f_e: \mathbb{G} \to \mathbb{G}$ by $f_e(g) = g^e$. If $\gcd(e, m) = 1$, then f_e is a permutation (i.e. a bijection). Also, if $d = e^{-1} \mod m$, then f_d is the inverse of f_e.
Definition: $\mathbb{Z}_N$: The additive abelian group, $\mathbb{Z} \mod N$, elements $\{0, 1, \ldots, N-1\}$.
Definition: $\mathbb{Z}_N^*$: The multiplicative abelian group, $\mathbb{Z} \mod N$. Consists of the elements g in $\{1, \ldots, N-1\}$ such that $\gcd(g, N) = 1$. There are all of the elements which are invertible $\mod N$.
Definition: ϕ: The Euler phi function: $\phi(N) = |\mathbb{Z}_N^*|$, the number of positive integers $< N$ which are relatively prime to N.
Theorem: Let $N = \Pi_i p_i^{e_i}$, where the $\{p_i\}$ are distinct primes, and $e_i \geq 1$ (take the prime factorization of N). Then $\phi(N) = \Pi_i p_i^{e_i - 1}(p_i - 1)$. In particular, if p prime, $\phi(p) = p - 1$, and if $N = pq$, p, q prime, then $\phi(N) = (p-1)(q-1)$.
Theorem: Take arbitrary $N > 1$, $a \in \mathbb{Z}_N^*$. Then $a^{\phi(N)} = 1 \mod N$. If $N = p$ is prime, and $a \in \{1, \ldots, p-1\}$, then $a^{p-1} = 1 \mod p$.
Theorem: Fix $N > 1$. For integer $e > 0$, define $f_e: \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ by $f_e(x) = [x^e \mod N]$. If e is relatively prime to ϕN, then f_e is a permutation. Let $d = e^{-1} \mod N$. Then f_d is the inverse of f_e.
Definition: Isomorphism: Let $\mathbb{G}, \mathbb{H}$ be groups with operations $\circ_{\mathbb{G}}$ and $\circ_{\mathbb{H}}$. A function $f: \mathbb{G} \to \mathbb{H}$ is an isomorphism if it is a bijection and $\forall g_1, g_2 \in \mathbb{G}$, $f(g_1 \circ_{\mathbb{G}} g_2) = f(g_1) \circ_{\mathbb{H}} f(g_2)$. If f exists, we say the groups are isomorphic, and $\mathbb{G} \simeq \mathbb{H}$. $|\mathbb{G}| = |\mathbb{H}|$.
Theorem: Chinese Remainder: CRT: Let $N = pq$, where $p, q > 1$ are relatively prime. Then $\mathbb{Z}_N \simeq \mathbb{Z}_p \times \mathbb{Z}_q$ and $\mathbb{Z}_N^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*$. Moreover, let f be the function mapping $x \in \mathbb{Z}_N$ to pairs (x_p, x_q), $x_p \in \mathbb{Z}_p$, $x_q \in \mathbb{Z}_q$, with $f(x) := ([x \mod p], [x \mod q])$, then f is an isomorphism from $\mathbb{Z}_N$ to $\mathbb{Z}_p \times \mathbb{Z}_q$, and the restriction of f to $\mathbb{Z}_N^*$ is an isomorphism from $\mathbb{Z}_N^*$ to $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$.