Definition: Private-key Encryption Scheme: 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: Kerchoffs' Principle: 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: Sufficient Key-space Principle: 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: Bayes Theorem: $\Pr[A|B] = \frac{\Pr[B|A]\cdot\Pr[A]}{\Pr[B]}$.

Definition: Perfect Secrecy: 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: $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$: (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: Perfect Indistinguishability: 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: One-time Pad: 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: Shannon's Theorem: 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 (t, ε)-secure if any adversary running for time at most t succeeds in breaking the scheme with probability at most ε.

Definition: PPT (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: $\mathtt{negl}$ (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: Secure: A scheme where any PPT adversary succeeds in breaking the scheme with at most negligible probability.

Definition: Probabilistic: 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: Secure: 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: Fixed-length encryption scheme: 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: $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$ (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: EAV-secure (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: $\Big|\Pr[\mathtt{out}_{\mathcal{A}}(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)) = 1] - [\mathtt{out}_{\mathcal{A}}(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)) = 1]\Big| \leq \mathtt{negl}(n)$.

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: PRG (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 expansion factor of G.

Construction: Stream Cipher: 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: $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$: 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: Multiple-EAV-Secure: 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: $\mathsf{PrivK}^{\mathsf{CPA}}_{\mathcal{A},\Pi}(n)$: $k \leftarrow \mathsf{Gen}(1^n)$, 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: CPA-secure (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: $\mathsf{PrivK}^{\mathsf{LR-CPA}}_{\mathcal{A},\Pi}(n)$: Same as $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$, but for CPA-security. Extends to Multiple-CPA-security.

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

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

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

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

Definition: PRF (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: Permutation: 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 efficient if $F_k(x)$ and $F_k^{-1}(x)$ are computable with a polynomial-time algorithm.

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

Definition: Synchronized: 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: Unsynchronized: Stream cipher mode which is stateless, taking a new IV each time.

Definition: ECB Mode (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: CBC Mode (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 not secure.

Definition: OFB Mode (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: CTR Mode (CounTeR): Pick an $\mathtt{ctr} = IV$, then $c_0 = \mathtt{ctr}$, $c_1 = F_k(\mathtt{ctr} + 1) \oplus m_1$, $\ldots$, $c_\ell = F_k(\mathtt{ctr} + \ell) \oplus m_\ell$. 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 $F_k(\mathtt{ctr} + \ell)$ is used as the new IV, remains secure.

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

Definition: $\underline{\mathsf{PrivK}^{\mathsf{CCA}}_{\mathcal{A},\Pi}(n)}$: 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: CCA-Secure (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: Non-Malleability: 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: MAC (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: Canonical Verification: 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: $\underline{\mathsf{Mac\text{-}forge}_{\mathcal{A},\Pi}(n)}$: $k \leftarrow \mathsf{Gen}(1^n)$. $\mathcal{A}$ 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: Secure MAC (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: Strong MAC: 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: CBC-MAC: 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 $\hat{t} := F_{k_2}(t)$.