Masters Dissertation

A Catalogue for Provably Secure Symmetric Modes and Primitives

Anisha Dutta

Roll No CRS1901
M.Tech. Cryptology and Security
Indian Statistical Institute, Kolkata

A Catalogue for Provably Secure Symmetric Modes and Primitives

Dissertation Submitted in Partial Fulfillment of the requirements for the degree of

Master of Technology in Cryptology and Security

Submitted by

Anisha Dutta

Roll No CRS1901
MTech Cryptology and Security
Indian Statistical Institute, Kolkata

Supervised by

Prof. Ir. Bart Preneel
Electrical Engineering Department
KU Leuven, Leuven, Belgium

and
Prof. Bimal Kumar Roy
Applied Statistical Unit
Indian Statistical Institute, Kolkata, India

Daily Supervisors

Dr. Elena Andreeva
Asst. Prof., Security and Privacy
TU Wien, Vienna, Austria

and
Amit Singh Bhati
Doctoral Researcher
KU Leuven, Leuven, Belgium

July 9, 2021

Dedicated to

The Afterglow of the Big Bang

CERTIFICATE

This is to certify that the dissertation entitled **"A Catalogue for Provably Secure Symmetric Modes and Primitives"** submitted by **Anisha Dutta** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology** in **Cryptology and Security** is a bonafide record of work carried out by him under our supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in our opinion, has reached the standard needed for submission.

Ir. Bart Preneel
Professor,
Electrical Engineering Department,
Katholieke Universiteit Leuven, Belgium.

Bimal Kumar Roy
Professor,
Applied Statistics Unit,
Indian Statistical Institute, Kolkata, India.

Acknowledgments

I would like to show my highest gratitude to my advisors, *Prof. Ir. Bart Preneel*, Professor, Electrical Engineering Department, Katholieke Universiteit Leuven, Belgium and *Prof. Bimal Kumar Roy*, Cryptology and Security Research Unit, Indian Statistical Institute, Kolkata, India for their guidance, continuous support and keeping faith in my capability. Prof. Bimal Kumar Roy has made it possible for me to take part in such a good research work and has helped me in every manner starting from the days at Indian Statistical Institute till date.

I would also like to thank *Dr. Elena Andreeva*, Asst. Prof., Security and Privacy, TU Wien, Vienna, Austria and *Amit Singh Bhati*, Doctoral Researcher, KU Leuven, Belgium for their continuous assistance from the beginning of this project till date. Both of them helped me a lot understanding every single aspect of the project, and in spite of their busy schedule, they were always there whenever I needed.

My deepest thanks to *Dr. Nilanjan Dutta*, Assistant Professor, TCG CREST, Kolkata and *Dr. Avik Chakraborti*, Assistant Professor, University of Exeter, UK for their valuable suggestions and discussions throughout my work. No matter how overcrowded their schedule was, they always helped me with every confusion that I had. I would like to show my utmost gratitude to my teacher *Prof. Avishek Adhikari*, Presidency University, Kolkata, India, for guiding and supporting me morally and academically at every point of time. Getting a chance to study at ISI Kolkata was not possible for me without his constant guidance.

Due to the global pandemic situation, I worked from ISI Kolkata hostel and my home during the entire project. Any words of gratitude will be less to my parents Mr. Amal Kumar Dutta Mrs. Shampa Dutta, my dearest sister Ananta Dutta and my friend Sayantan Chakraborty for their everlasting support, both in academic and moral means, and desire to make me feel happy always. I feel lucky to have you all in my life.

Anisha Dutta
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Modern-day Symmetric-key Cryptology is enriched with numerous contributions including symmetric-key primitives to modes of operation. The approach to design and develop provably secure designs have accelerated the growth of this subject. A lot of encryption, authentication and authenticated encryption modes are available publicly that are provably secure and gives good results in terms of efficiency. But there is not much resource that studies all the modes and conclude about their performance at the same time. This work is intended to study and compare all these modes of operation, both in terms of security (confidentiality and integrity) and efficiency (implementation area and throughput). We took care of different security notions and design rationales of compared schemes and generalised them as much as possible.

Keywords : Symmetric Cryptology, Modes of Operations, Provable Security, Encryption, Authentication, Authenticated Encryption, Web-Implementation.

Contents

List of Tables

List of Figures

List of Algorithms

1 Introduction

1.1 Introductory Notes

Cryptography is commonly called as the science of secrets. In the distant past, it was quite simple a process where the sender used to scramble the messages and sent it so even if some adversarial body gets hold of the message, they could not understand the message. At that moment, the baby version of cryptography focused on message confidentiality (which is treated to be the encryption) — conversion of the whole messages from a readable form into an unreadable one and back again at the opposite end, without leaking any message to the eavesdroppers. This so called encryption attempted to ensure secrecy in communications between spies, military leaders, and diplomats. In this modern era of computation, the subject cryptography itself has become a keystone of system and network security, encompassing all the ways of secrecy, verification, authentication and communication.

Symmetric key cryptography offers simple encryption schemes that comes with implementation-efficient mathematical structures. Some building blocks of symmetric encryption are Block-Ciphers, Stream-Ciphers and Hash functions. By only hiding the encryption key, these simple protocols offer high security. Alongside, these primitives are easy to implement as they do not require rigorous computations as prerequisite. In order to achieve security, these schemes generally do not require large integers or finite field arithmetic to be implemented.

But everything comes at a price. In search for security, efficiency gets lost. Block ciphers, forkciphers and tweakable block-ciphers are designed to encrypt short, fixed length messages. Thus, to encrypt large messages, the most basic idea that we can think of is to fragment the messages. But, using same key and same cipher for encrypting long messages (fragmented in parts) results in increasing the advantage of adversary. That is, the adversary can keep track of a code book and can get advantage. Again, choosing new keys every time kills the simplicity of implementation. This is why we need to assemble the ciphers in some standard way, so that choosing only one key will suffice. Here modes of operation comes into play.

Different encryption methods have been used throughout history to prevent non-authorized people from understanding or forging messages. Confidentiality of encryption is the property which guarantees that anyone without the proper knowledge of key cannot recognize the original data, whereas data integrity ensures accuracy and reliability of the data to an authorized user. Establishing confidentiality and integrity of communication is certainly the most important goals of cryptography. The notion of authenticated encryption combines these two security goals in a single symmetric-key, cryptographic primitive. A lot of effort has been invested in authenticated encryption till date. The recent competitions for Authentication modes, Block Ciphers, Authenticated Encryption Modes has boosted the research activity in this area even more. As a result, the area of Modes of Operation boasts numerous results, both theoretically and practically oriented. Alongside, every year, a lot of mode of operation gets published in various cryptographic conference and journal.

Our work explore the current landscape of results on these Modes of operation. The scope of Modes of operation is very large. Despite the modes being standardized and well-known, there is no significant work that summarizes all the schemes, and conclude about their comparative analysis. In our survey, we tried to cover up almost all published modes, and the comparative analysis explores pros and cons of the modes in a systematic way.

1.2 Project Summary

1.2.1 Project Objective

Various modes of operations have been standardised by multiple cryptographic competitions and journals throughout last few decades. Since the research volume is increasing day by day, we are getting new ideas, new schemes and new directions in this field to enrich the subject. As a result, we currently have a huge numbers of various modes of operations, which we can use according to our need in specific research work. Some of them are highly efficient, some are vulnerable with some loopholes but works well in other situations and some are advised to use in certain scenarios for better efficiency and security measures. Despite the modes being standardized and well-known, their goal and properties vary. This actually bothers a researcher to choose any mode swiftly as they have to go through a good number of well-entrenched schemes to effectively pick the suitable one.

The primary objective of this project is to have a systematic study on all these modes of operations and to compare them in a systematic manner. The final output of this project will be in form of a data platform, which will help one to easily compare between different choices of modes of operation with respect to desired criteria. This will save a lot of manual effort and time of people and organizations worldwide as they will be capable of choosing their required mode of operation without going through each mode appearing in the major worldwide competitions' and conferences submission lists. Also since this already contains a list of huge number of different modes, this work significantly widens the range of choices of an actual designer of cryptographic system.

1.2.2 Role of Standards Bodies

Since the volume of the research in different areas of Electronic Engineering and associated disciplines started to flourish at their best speed, the last century saw a major development into forming different professional associations to promote innovation and industrial competitiveness. These bodies do hold a lot of credit for the current research atmosphere for these subjects. To name a few, National Institute of Standards and Technology (NIST, formed as National Bureau of Standards in 1901, became NIST in 1988), Institute of Electrical and Electronics Engineers (IEEE, formed in 1963), ISO/IEC JTC 1 (Formed in 1987), etc. These organizations conducts several competitions to develop, maintain and promote standards in the fields of Cryptology and Cyber Security.

Cryptographic competitions, conferences and journals that are recommended or standardised by NIST are widely viewed as the safest way to select cryptographic algorithms. The reason is pretty clear as the selected modes in these fields are collectively, quite good(both in terms of security and efficiency), and NIST, in particular, has shown commendable leadership in their work on standardizing or recommending these modes of operation. There is obvious reasons that standards often take years to complete because sculpting a proper and nearly perfect standard is hard. These symmetric cryptography standards are designed for specific requirements, well-informed by the science that the cryptographic community has been working to create and that this is precisely the direction we want our research to be continued.

Several regional standards deal with cryptography in general and some of the algorithm specified in them can be considered to be lightweight. Lightweight cryptography has been one of the intense topics in symmetric cryptography in the recent years. A huge number of lightweight algorithms have been published, standardized and/or used in commercial products. In the USA, cryptographic standards are handled mainly by NIST, which famously standardized the AES after an open competition. NIST LWC is arguably the biggest competition for the Light-Weight Cryptographic protocols. The Japanese Cryptography Research and Evaluation Committees (CRYPTREC) also maintains the "e-Government Recommended Ciphers List" which contains algorithms whose usage should be preferred. The Telecommunications Technology Association of Korea (TTA) provides the relevant standards in South Korea. In Europe, the NESSIE project selected several block ciphers and eSTREAM competition is popular to find good stream ciphers. NIST notably organized a Hash function competition to create a new hash standard, SHA-3. This was not meant to replace SHA-2, as no significant attack on SHA-2 has been demonstrated, but for the need for an alternative, dissimilar cryptographic hash function. This competition attracted 64 hash-function submissions from 200 cryptographers around the world, and then through a tremendous volume of security evaluations and performance evaluations, eventually NIST chose Keccak as SHA-3. The AES competition was also organized by the NIST. The goal of the Advanced Encryption Standard (AES) competition was to specify "an unclassified, publicly disclosed encryption algorithm capable of protecting sensitive government information well into the next century". Each AES submission was required to be a block cipher supporting a block length of 128 bits and key lengths of 128, 192, and 256 bits with specific properties regarding Security, Cost and Hardware and software suitability. PHC (Password Hashing Competition) ran from 2013 to 2015 as an open competition — the same kind of process as NIST's AES and SHA-3 competitions. CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) is one of the most popular competition for designing new AEAD Modes.

Apart from the Cryptographic competitions organised by the standard bodies, the conferences also contribute a lot in the research. CRYPTO, the International Cryptology Conference, is an academic conference on all aspects of cryptography and cryptanalysis. The first CRYPTO, held in 1981, was the first major conference on cryptology and was all the more important because relations between government, industry and academia were rather tense. Encryption was considered a very sensitive subject and the coming together of delegates from different countries was unheard-of at the time. The initiative for the formation of the International Association for Cryptologic Research (IACR) came during CRYPTO '82, and CRYPTO '83 was the first IACR sponsored conference. The IACR organizes and sponsors annual flagship conferences - Crypto, Eurocrypt, Asiacrypt and some area conferences in specific sub-areas of cryptography and symposiums like Fast Software Encryption (FSE), Public Key Cryptography (PKC), Cryptographic Hardware and Embedded Systems (CHES), Theory of Cryptography (TCC), Real World Crypto Symposium (RWC). Cryptology Research Society of India (CSRI) organises Indocrypt, an international Conference on Cryptology annually in India.

Cryptographic competitions are not a foolproof though. DES, the output of the cryptographic competition at NBS(1976), had faced major issues like linear and differential cryptanalytic attacks. Even the famous AES cipher have undergone timing attacks for its primary versions[4]. As a third example, SHA-3 was forced to aim for a 2^{512} level of preimage security, and as a result, performance complaints and slowing down deployment were reported. Due to the restriction of time-upper bound and availability of too many schemes might result into overlooking some important schemes and not including them in this thesis. A critique is always welcome bring out these shortcomings.

1.2.3 Our Contribution

Symmetric encryption, being an older method of encryption and being much faster and more efficient than asymmetric encryption, takes a toll on networks due to performance issues with data size and heavy CPU usage. For symmetric key-cryptology, since the key for both encryption and decryption are same, thus it takes no effort to exchange keys. Moreover, due to the better performance and faster speed of symmetric encryption compared to asymmetric, symmetric cryptography is typically used for bulk encryption / encrypting large amounts of data. Hence, cryptographers always had a special eye on the research areas of Symmetric Cryptography. Considering the growth in this field since the last few decades, we can notice that for security aspects like confidentiality and integrity, there are various symmetric cryptographic schemes which provide a certain level of security making various efficiency or functionality trade-offs. These schemes are published and standardised by various cryptographic competitions. This project targets to somewhat tries to bind all the aspects of these schemes under one roof. We can formally distribute the aims of the project into these categories.

1. **Studying the security and functionalities of Provable Symmetric Schemes.**

   This project heavily focuses on categorising the various Provably Secure Symmetric Modes with respect to the security and efficiency parameters. Hence, we shall be describing the parameters in details so that one can learn these parameters well enough before going through the detailed evaluation and comparison between the symmetric modes. Also, if some additional parameters or security definitions are required to study for some particular sections, we have kept them there for better understanding.

2. **Producing exhaustive classification of Symmetric Modes of Operation.**

   We have gone through all available standard modes of operations from various well-known cryptographic competitions of recent years. We have studied the Symmetric Modes from the likes of CAESAR, NIST-LWC, etc. competitions and various conferences and enlisted the modes with their respective properties including security type, security bound, rate, state size, key size, whether or not it has some underlying building blocks, different variants, etc. As a result, one can go through the listed properties of given mode and easily judge whether the mode can be used for her desired scenario.

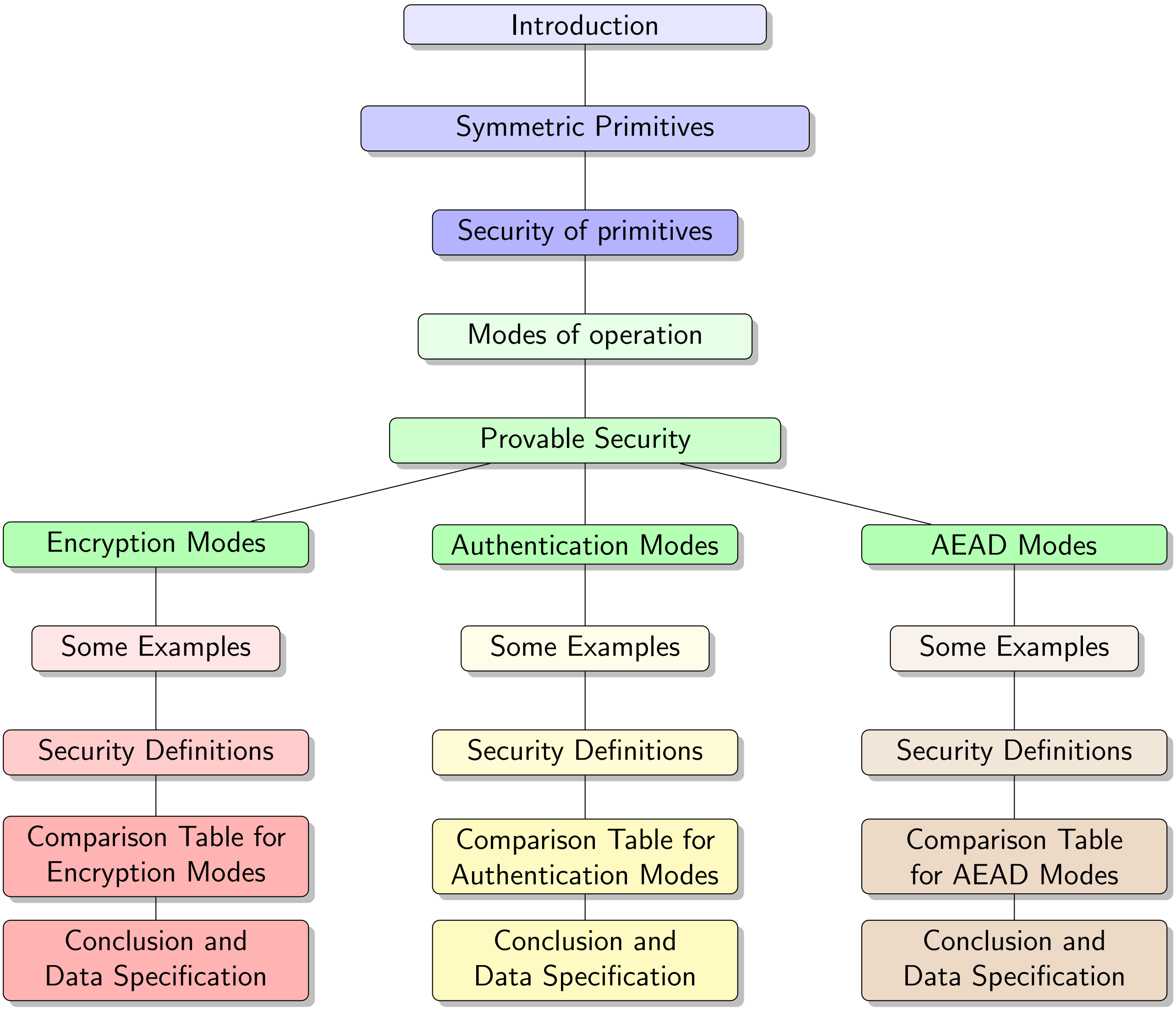
3. **Discussing exhaustive comparison between different Modes of Operations.**

   After listing down properties in tabular format, we compare them through those tables, which helps one to further choose which mode to be considered for her required criteria, be it to be having less overhead or which are resistant against misuse of nonce/initialization vector or which one guarantees security for finite and/or variable length inputs etc. This kind of work with such a large volume of provable symmetric modes, to the best of our knowledge, is a first attempt in this domain.

4. **Combining the classified data in suitable public online domain.**

   We have published the whole data through a publicly available web-application that will mention all functionalities, requirements efficiency parameters and security bounds of the standard modes. This, we believe, shall be one of a kind library in the domain of provable symmetric modes of operation.

1.3 Thesis Outline



The above chart briefly describes flow and the contents of this project report. The contents in blue, green, red, yellow and brown colors are to be studied from Sections 2,3,4,5 and 6, resp. We advise the reader to complete sections 2 and 3 before moving into the later part. Last three sections are for discussing Encryption, Authentication and AEAD Modes, resp.

2 Symmetric-key Primitives

Symmetric Key algorithms are cryptographic algorithms that use the same cryptographic keys for both the encryption of plaintext and the decryption of ciphertext. The keys may be identical, or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. The requirement that both parties have access to the secret key is one of the main drawbacks of symmetric-key encryption, in comparison to public-key encryption (also known as asymmetric-key encryption).

A secure symmetric encryption scheme $\mathcal{SE} = (\mathbf{KeyGen}, \mathbf{Enc}, \mathbf{Dec})$ consists of three algorithms, described below.

- $\mathbf{KeyGen}(1^n)$: The key-generation algorithm generates a key K of fixed length n in a probabilistic manner according to the protocol.
- $\mathbf{Enc}(K, M)$: The encryption algorithm **Enc** is probabilistic in nature and is keyed by the key K generated from **KeyGen**. By probabilistic, we mean, the output for this function with a given message $M \in \mathcal{M}$ is uniformly and identically distributed over the space of ciphertexts.
- $\mathbf{Dec}(K, C)$: The decryption algorithm Dec is a deterministic algorithm for any choice of $K \in \mathcal{K}, C \in \mathcal{C}$.

Correctness: The scheme $\mathcal{SE}$ is correct if for any choice of $M \in \mathcal{M}, K \in \mathcal{K}$, we have $\mathbf{Dec}(K, \mathbf{Enc}(K, M)) = M$.

Unlike public key cryptosystems, symmetric key cryptographic primitives offer us secure and efficient schemes that comes without rigorous mathematical constructions. We will discuss various important symmetric key primitives and their properties.

2.1 Primitives

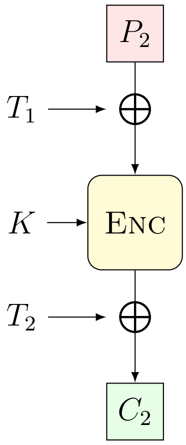
- **Block Cipher**

  Block cipher is a deterministic algorithm operating on fixed-length bits, called blocks. A secure block cipher is suitable for the encryption of only a single block of data at a time, using a fixed key. Block ciphers are building blocks in other cryptographic protocols, such as universal hash functions and pseudorandom number generators.

  A block cipher consists of three paired algorithms, $KeyGen$ for key generation, $\mathcal{E}$ for encryption, and the other for decryption, $\mathcal{D}$. Both $\mathcal{E}$ and $\mathcal{D}$ accept two inputs : an input block of size n bits and a key of size k bits; and both yield an n-bit output block. The decryption algorithm $\mathcal{D}$ is defined to be the inverse function of encryption, *i.e.* $\mathcal{D} = \mathcal{E}^{-1}$. More formally, a block cipher is specified by an encryption function $E_K(P) := E(K, P) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, which takes as input a key K, of bit length k (called the key size), and a bit string P, of length n (called the block size), and returns a string C of n bits. $P \in \{0,1\}^n$ is called the plaintext, and $C \in \{0,1\}^n$ is termed the ciphertext. For each K, the function $E_K(P)$ is required to be an invertible mapping on $\{0,1\}^n$. The inverse for $\mathcal{E}$ is defined as a function $E_K^{-1}(C) := D_K(C) = D(K, C) : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, taking a key K and a ciphertext C to return a plaintext value P, such that $\forall K : D_K(E_K(P)) = P$. For example, a block cipher encryption algorithm might take a 128-bit block of plaintext as input, and output a corresponding 128-bit block of ciphertext. The exact transformation is controlled using a second input – the secret key. Decryption is similar : the decryption algorithm takes, in this example, a 128-bit block of ciphertext together with the secret key, and yields the original 128-bit block of plain text.

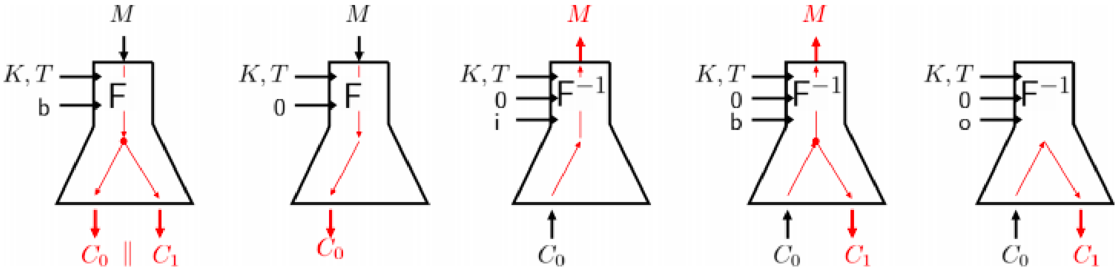
- **Tweakable Block Cipher**

  Tweakable Block ciphers(TBC) are similar to the block ciphers, enhanced with tweak T. Changing a key, (when suspected to be compromised) can be expensive as it can change the internal mechanism. Whereas, modifying tweak can be done in a cheap manner. The tweaks are added before and after applying the underlying block cipher. Thus, a TBC $\tilde{\mathcal{E}}$ can be considered as the tuple of functions $(KeyGen, \widetilde{Enc}, \widetilde{Dec})$ where $Keygen$ is probabilistic function that outputs key K and tweak T_1, T_2. $\widetilde{Enc}_{T_1,T_2} : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is the $\widetilde{Enc}$ function constrained with tweaks T_1 and T_2, is defined as $\widetilde{Enc}_{T_1,T_2}(K, M) = Enc_K(T_1 \oplus M) \oplus T_2$. Here $\mathcal{E} = (Enc, Dec)$ is the underlying block cipher. $\widetilde{Dec}_{T_1,T_2} : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is defined as $\widetilde{Dec}_{T_1,T_2}(K, C) = Dec_K(T_2 \oplus C) \oplus T_1$. The graphical representation of TBC is as follows



- **ForkCipher**

  A forkcipher is a pair of deterministic algorithms, the encryption algorithm:
  $F : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \times \{0,1,b\} \rightarrow \{0,1\}^n \bigcup \{0,1\}^n \times \{0,1\}^n$
  and the decryption algorithm
  $F^{-1} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \times \{0,1\} \times \{i,o,b\} : \{0,1\}^n \bigcup \{0,1\}^n \times \{0,1\}^n$
  The functions F, F^{-1} are defined as follows: $F(K,T,M,0) = C_1$, $F(K,T,M,b) = C_2$, $F(K,T,M,1) = C_1||C_2$, $F_{-1}(K,T,C_1,0,i) = M$, $F_{-1}(K,T,C_2,1,i) = M$, $F_{-1}(K,T,C_1,0,b) = M||C_2$, $F_{-1}(K,T,C_2,1,b) = M||C_1$, $F_{-1}(K,T,C_1,0,o) = C_2$, $F_{-1}(K,T,C_2,1,o) = C_1$. Pictorially,



- **Stream Cipher**

  A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream). A stream cipher is a pseudorandom key-stream generator. The encryption using stream ciphers is XOR-based.

  The pseudorandom keystream is typically generated serially from a random seed value using digital shift registers. The seed value s serves as the cryptographic key for decrypting the ciphertext stream. Stream ciphers represent a different approach to symmetric encryption from block ciphers. Block ciphers operate on large blocks of digits with a fixed, unvarying transformation. This distinction is not always clear-cut: in some modes of operation, a block cipher primitive is used in such a way that it acts effectively as a stream cipher. Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity. However, stream ciphers can be susceptible to serious security problems if used incorrectly (see stream cipher attacks); in particular, the same starting state (seed) must never be used twice.

- **Cryptographic Hash Function**

  A hash function is a function H that, given a message M of an arbitrary length, returns a value $H(M) = h$ of a fixed length n. They have many applications in computer security such that Message Authentication Codes (MAC), Digital Signatures and User Authentication. Hash functions are efficiently computable. A hash function H is called a cryptographic hash function if the following assumptions hold for H:

  Collision resistance. Finding two messages M and $M' \neq M$ such that $H(M) = H(M')$ should be 'hard'. It shall cost $\Omega(2^{\frac{n}{2}})$ by the birthday paradox.

  Pre-Image resistance. From a hash h, finding a message M so that $H(M) = h$ should cost $\mathcal{O}(2^n)$ by exhaustive search.

  Second Pre-Image resistance. Given a message M and its hash $H(M)$, finding another message M' such that $H(M) = H(M')$ should cost $\mathcal{O}(2^n)$ by exhaustive search.

- **Compression Functions**

  Compression functions are a variant of hash functions, restricted over fixed-sized domains. A comression function $CF : 0, 1^m \rightarrow 0, 1^n$ is a function that takes a message M of length $m > n$ as input and returns a value $CF(M) = h$ of fixed length n. Compression functions are one-way. These can be composed to construct hash functions.

There are many provably secure and standardised primitives that we can use as building blocks for various cryptographic purposes. Examples of popular block cipher algorithms including AES, GOST, KASUMI, PRESENT, SEA, TEA, KTANTAN, LED, TWINE etc., most of which are lightweight in terms of state size. Whereas, A5/1, ChaCha, Trivium, Grain, SNOW-3G, MICKEY are secure stream cipher candidates. Encrypting a message does not guarantee that it will be authentic. Hence, often a message authentication code is added to a ciphertext to ensure that changes to the ciphertext over a public channel will be noted by the receiver. Message authentication codes can be constructed from block cipher or hash functions. We can also build hash functions from block ciphers.

2.2 Security of Primitives

1. **Pseudo-Random Function (PRF)**

   A function $f : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^m$ is called a (t, ϵ, q) *Pseudo Random function* if the followings hold
   • Given a key $K \in \{0,1\}^s$ and an input $X \in \{0,1\}^n$, there is an efficient algorithm to compute $F_K(X) = f(X, K)$.
   • For any polynomial time adversary $\mathscr{A}$, $\mid Pr_{K\in\{0,1\}^s}[\mathscr{A}^{F_K}(x) = y] - Pr_{K\in\{0,1\}^s}[\mathscr{A}^{f}(x) = y] \mid < \epsilon$, for any $y \in \{0,1\}^m$ and $f : \{0,1\}^n \to \{0,1\}^m \in \mathscr{F}$. ($\mathscr{F}$ is the family of all n-input m-output Boolean functions).

2. A PRF F is called **pseudorandom permutation(PRP)** if it is a permutation on $\{0,1\}^s$, and satisfies the properties of a PRF.

3. **PRP-CPA Security of an Encryption Scheme**

   Let $\mathcal{E} = (Enc, Dec)$ be an encryption scheme, $\mathcal{PRP}$ be the set of all pseudorandom permutations. $Enc : \mathcal{P} \times \mathcal{K} \times \mathcal{M} \to \mathscr{C}$ depends upon parameter set *Params*. Consider the following game for both cases of **Exp**.

Adversary$\mathcal{A}$		**Challenger** $\mathcal{C}$	
		choose $K \xleftarrow{\$} \mathcal{K}$	
		choose $F \xleftarrow{\$} \mathcal{PRP}$	
For $i = 1, 2, \ldots, n$			
$m_i \leftarrow \mathcal{M}$			
$Params_i \leftarrow \mathcal{P}$	$\xrightarrow{(Params_i, m_i)}$		
		choose $b \xleftarrow{\$} \{0,1\}$	
		For $i = 1, 2, \ldots, n$	
		if $b = 0$	if $b = 1$
		$\mathbf{Real}_{\mathcal{E}_K^{Params_i}}(m_i)$	$\mathbf{PRP}_{\mathcal{E}_K^{Params_i}}(m_i)$
		$c_i \leftarrow \mathcal{E}_K.Enc(Params_i, m_i)$	$c_i \leftarrow F(Params_i, m_i)$
	$\xleftarrow{c_i}$		
$b' \leftarrow \mathbf{Exp}^{CPA-j}(\mathcal{A})$			
output b'			

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with $Input = (m_1, c_1, Params_1, m_2, c_2, Params_2, ..., m_n, c_n, Params_n)$:

Case 1:	**Case 2:**
$\mathbf{Exp}^{PRP-CPA-1}(\mathcal{A})$	$\mathbf{Exp}^{PRP-CPA-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$g \xleftarrow{\$} \mathcal{PRP}$
$b' \leftarrow \mathcal{A}^{\mathcal{E}_K}(Input)$	$b' \leftarrow \mathcal{A}^{g}(Input)$

The PRP-CPA-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{prp-cpa}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{prp-cpa-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{prp-cpa-0}(\mathcal{A}) = 1]|$$

.

4. **PRP-CCA Advantage of an Encryption Scheme**

Let $\mathcal{E} = (Enc, Dec)$ be an encryption scheme, $\mathcal{PRP}$ be the set of all pseudorandom functions. $Enc : \mathcal{P}_1 \times \mathcal{K} \times \mathcal{M} \to \mathscr{C}$ depends upon parameter set $Params$. Consider the following game for both cases of **Exp**.

Decide Encryption scheme $\mathcal{E}$

Adversary$\mathcal{A}$

Challenger $\mathcal{C}$

choose $K \xleftarrow{\$} \mathcal{K}$

choose $F \xleftarrow{\$} \mathcal{PRP}$

For $i = 1, 2, \ldots, n$
$m_i \leftarrow \mathcal{M}$
$Params_i \leftarrow \mathcal{P}$
For $j = 1, 2, \ldots, n'$
${c_j}' \leftarrow \mathscr{C}$
${Params_j}' \leftarrow \mathcal{P}$

$\xrightarrow[(Params'_j, c'_j)]{(Params_i, m_i)}$

choose $b \xleftarrow{\$} \{0, 1\}$

For $i = 1, 2, \ldots, n$ **For** $j = 1, 2, ..., n'$

if $b = 0$	if $b = 1$
$\mathbf{Real}_{\mathcal{E}_K^{Params_i}}(m_i)$	$\mathbf{PRP}_{\mathcal{E}_K^{Params_i}}(m_i)$
$c_i \leftarrow \mathcal{E}_K.Enc(Params_i, m_i)$	$c_i \leftarrow F(Params_i, m_i)$
$m'_j \leftarrow \mathcal{E}_K.Dec(Params'_j, c'_j)$	$m'_j \leftarrow F(Params'_j, c'_j)$

$\xleftarrow{c_i, m'_j}$

$b' \leftarrow \mathbf{Exp}^{CPA-j}(\mathcal{A})$
output b'

$\mathcal{A}$ wins the game if $b' = b$

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with $Input = (m_1, c_1, Params_1, m_2, c_2, Params_2, ...,$
$m_n, c_n, Params_n; c'_1, m'_1, Params'_1, ..., c'_{n'}, m'_{n'}, Params'_{n'})$:

$$\begin{array}{c|c} \textbf{Case 1:} & \textbf{Case 2:} \\ \mathbf{Exp}^{PRP-CCA-1}(\mathcal{A}) & \mathbf{Exp}^{PRP-CCA-0}(\mathcal{A}) \\ K \xleftarrow{\$} \mathcal{K} & g \xleftarrow{\$} \mathcal{PRP} \\ b' \leftarrow \mathcal{A}^{\mathcal{E}_K}(Input) & b' \leftarrow \mathcal{A}^{g}(Input) \end{array}$$

The PRP-CCA-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{prp-cca}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{prp-cca-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{prp-cca-0}(\mathcal{A}) = 1]|$$

.

5. **LOR-CPA security:** The left-or-right CPA security game for an encryption scheme $\mathcal{E}$ is as follows:

Adversary$\mathcal{A}$ **Challenger** $\mathcal{C}$

choose $K \xleftarrow{\$} \mathcal{K}$

For $i = 1, 2, \ldots, n$
$m_i, m'_i \leftarrow \mathcal{M}$
$params_i, params'_i \leftarrow \mathcal{P}$

$\xrightarrow[(params_i, m_i)]{(params'_i, m'_i)}$

choose $b \xleftarrow{\$} \{0, 1\}$
For $i = 1, 2, \ldots, n$

$$\begin{array}{c|c} \text{if } b = 0 & \text{if } b = 1 \\ \textbf{Left}_{\mathcal{E}_K}^{params_i, params'_i}(m_i, m'_i) & \textbf{Right}_{\mathcal{E}_K}^{params_i, params'_i}(m_i, m'_i) \\ c_i \leftarrow \mathcal{E}_K.Enc(params_i, m_i) & c_i \leftarrow \mathcal{E}_K.Enc(params'_i, m'_i) \end{array}$$

$\xleftarrow{c_i}$

$b' \leftarrow \mathbf{Exp}^{LOR}(\mathcal{A})$
output b'

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with $Input = (m_1, params_1, m_2, params_2, ..., m_n, params_n; m'_1, params'_1, m'_2, params'_2, ..., m'_n, params'_n; c_1, c_2, ..., c_n)$:

$$\begin{array}{c|c} \textbf{Case 1:} & \textbf{Case 2:} \\ \mathbf{Exp}^{LOR-1}(\mathcal{A}) & \mathbf{Exp}^{LOR-0}(\mathcal{A}) \\ K \xleftarrow{\$} \mathcal{K} & K \xleftarrow{\$} \mathcal{K} \\ b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,1)}(Input) & b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,0)}(Input) \end{array}$$

The LOR-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{lor-cpa}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-0}(\mathcal{A}) = 1]|$$

.

6. **LOR-CCA security:** The left-or-right CCA security game for an encryption scheme $\mathcal{E}$ is as follows:

Adversary$\mathcal{A}$

Challenger $\mathcal{C}$

choose $K \xleftarrow{\$} \mathcal{K}$

For $i = 1, 2, \ldots, n$

$m_i, m'_i \leftarrow \mathcal{M}$

$params_i, params'_i \leftarrow \mathcal{P}$

For $j = 1, 2, \ldots, n'$

$c_j, c'_j \leftarrow \mathcal{M}$

$params_j, params'_j \leftarrow \mathcal{P}$

$$\xrightarrow[(params_i, m_i)]{(params'_i, m'_i)}$$

$$\xrightarrow[(params_j, c_j)]{(params'_j, c'_j)}$$

choose $b \xleftarrow{\$} \{0, 1\}$

For $i = 1, 2, \ldots, n$

if $b = 0$	if $b = 1$
Left$_{\mathcal{E}_K^{params_i, params'_i}}(m_i, m'_i)$	**Right**$_{\mathcal{E}_K^{params_i, params'_i}}(m_i, m'_i)$
$c_i \leftarrow \mathcal{E}_K.Enc(params_i, m_i)$	$c_i \leftarrow \mathcal{E}_K.Enc(params'_i, m'_i)$
$m_j \leftarrow \mathcal{E}_K^{-1}.Enc(params_j, c_j)$	$m_j \leftarrow \mathcal{E}_K^{-1}.Enc(params'_j, c'_j)$

$$\xleftarrow[m^b{}_J]{c^b{}_i}$$

$b' \leftarrow \mathbf{Exp}^{LOR}(\mathcal{A})$

output b'

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CCA-1}$ and $\mathbf{Exp}^{CPA-0}$ with $Input = (m_1, params_1, m_2, params_2, ..., m_n, params_n; m'_1, params'_1, m'_2, params'_2, ..., m'_n, params'_n; c^b{}_1, c^b{}_2, ..., c^b{}_n; c_1, params_1, ..., c_{n'}, params_{n'}; c'_1, params'_1, c'_{n'}, params'_{n'}); m^b_1, ...m^b_n$:

Case 1:	**Case 2:**
$\mathbf{Exp}^{LOR-1}(\mathcal{A})$	$\mathbf{Exp}^{LOR-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,1)}(Input)$	$b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,0)}(Input)$

The LOR-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{lor-cca}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-0}(\mathcal{A}) = 1]|$$

.

7. **Tweakable PRP or TPRP security**

This security notion is defined for tweakable block ciphers.Suppose $\mathcal{E}^{T,\beta} = (F, F^{-1})$ be a Tweakable block cipher encryption scheme, $\mathcal{PRP}$ be the set of all random permutation, T be the tweak for $\mathcal{E}$ selected prior to starting the game. Thus the parameters for encryption will be $params = T, T \in \mathcal{T}$. The TPRP security of $\mathcal{E}$ is defined as using the PRP-CCA game with real and ideal games is as follows.

TPRP-Real$_F$:	**TPRP-Ideal**$_F$:
Get M or C from $\mathcal{A}$	Get M or C from $\mathcal{A}$
$K \longleftarrow \{0,1\}^k$	for $T \in \mathcal{T}$, $\pi_T \longleftarrow_{\$} \mathcal{PRP}$
Oracle $Enc(T, K, M)$	**Oracle** $Enc(T, \beta, K, M)$
return $c \longleftarrow F(T, K, M)$	**return** $c \longleftarrow \pi_T(K, M)$
Oracle $Dec(T, K, M)$	**Oracle** $Dec(T, K, M)$
return $m \longleftarrow F^{-1}(T, K, C)$	if $\exists\ m\ \ni\ \pi_T(K, M) = C$
	then **return** m
	else **return** $\perp$

Then the TPRP advantage of the adversary $\mathcal{A}$ for the scheme F will be

$$Adv_{\mathcal{E}}^{TPRP}(\mathcal{A}) = |Pr[Exp_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

8. **Pseudorandom tweakable forked permutation Security (PRTFP)**

Suppose $\mathcal{E}^{T,\beta} = (F, F^{-1})$ be a Fork Cipher encryption scheme, $\mathcal{PRP}$ be the set of all random permutation, T be the tweak for $\mathcal{E}$ selected prior to starting the game. Thus the parameters for encryption will be $params = (T, \beta), T \in \mathcal{T}, \beta \in \{0, 1, b, i, o\}$. The PRTFP security of $\mathcal{E}$ is defined as using the PRP-CCA game with real and ideal games is as follows.

PRTFP-Real$_F$:	**PRTFP-Ideal**$_F$:
Get M or C from $\mathcal{A}$	Get M or C from $\mathcal{A}$
$K \longleftarrow \{0,1\}^k$	for $T \in \mathcal{T}$, $\pi_{T,A,N} \longleftarrow_{\$} \mathcal{PRP}$
Oracle $Enc(T, \beta, K, M)$	**Oracle** $Enc(T, \beta, K, M)$
return $c \longleftarrow F(T, \beta, K, M)$	**return** $c \longleftarrow \pi_{T,A,N}(K, M)$
Oracle $Dec(T, \beta, K, M)$	**Oracle** $Dec(T, \beta, K, M)$
return $m \longleftarrow F^{-1}(T, \beta, K, C)$	if $\exists\ m\ \ni\ \pi_{T,A,N}(K, M) = C$
	then **return** m
	else **return** $\perp$

Then the PRTFP advantage of the adversary $\mathcal{A}$ for the scheme F will be

$$Adv_{\mathcal{E}}^{PRTFP}(\mathcal{A}) = |Pr[Exp_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

9. **LPRF or leakage resistant PRF Security**

Let $\mathcal{F} : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}^t$ be a function family. Consider the following LPRF Game.

```
Game LPRF

Procedure Initialize

$b \xleftarrow{\$} \{0,1\}, K \xleftarrow{\$} \mathcal{K}$
return

Procedure LF(X,L)

$y \leftarrow \mathcal{F}(K, X)$
$\Lambda \leftarrow L(K, X)$

Procedure F(X)

if $b = 0$ then
    if $f(x) = \perp$ then
        $f(x) \xleftarrow{\$} \{0,1\}^t$
    end if
    Return $f(x)$
else
    return $\mathcal{F}(K, X)$
end if

Procedure Finalize(b')

return $(b' = b)$
```

For any adversary $\mathcal{A}$, $Adv_{\mathcal{F}}^{lprf}(\mathcal{A}, \mathcal{L}_F) = |Pr[LPRF^{\mathcal{A}} \implies \textbf{true}] - \frac{1}{2}|$

10. **Birthday Bound Security**

A birthday attack is a type of cryptographic attack that exploits the mathematics behind the hardness of birthday problem in probability theory. This attack can be used to abuse communication between two or more parties. The attack depends on the higher likelihood of collisions found between random attack attempts and a fixed degree of permutations.
Suppose an adversary wants to exploit a cryptosystem $\mathcal{E} = (E, E^{-1})$ that encrypts n-bit messages into n-bit ciphertexts. Consider the IND-CPA game as follows:

```
Game $IND-CPA^{\mathcal{E}_K}(\mathcal{A})$:
$\mathcal{A}$ queries with 2 messages $m_0, m_1$ to $\mathcal{C}$
$\mathcal{C}$ chooses $b \xleftarrow{\$} \{0,1\}$
$\mathcal{C}$ sends $C_b = E_K(m_b)$ to $\mathcal{A}$
$\mathcal{A}$ returns $b'$
$\mathcal{A}$ wins if $b = b'$
```

Before returning b', adversary can query to the encryption oracle of E_K only, with messages other than m_1, m_0. Then the adversary succeeds with probability at most $\frac{1}{2}$. For evaluating the ciphertext correctly from a given plaintext and encryption algorithm E, an adversary needs the key K. But without having the knowledge of K, the adversary has no other way than guessing – which he can do correctly with probability only $\frac{1}{2^n}$.

11. **Security beyond birthday bound**

As discussed in the definition of birthday bound security, the attacker has to make at least $\mathcal{O}(2^{n/2})$ many queries to attack the cryptosystem. Now, if for a construction, one can show that even after making $\mathcal{O}(2^{n/2})$ many queries, no adversary can obtain enough data to attack the cryptosystem, then the system is called beyond birthday bound secure.

It is clear that after $\mathcal{O}(2^n)$ many queries any attacker will have all information about the cryptosystem. To show that a cryptosystem is beyond-birthday secure, the objective will be to show that any adversary have to query $\mathcal{O}(2^{rn})$ times to obtain enough knowledge to issue a successful attack, where $1 < r < \frac{1}{2}$. In this case, the beyond-birthday-bound of the cryptosystem will be $\mathcal{O}(2^{rn})$.

3 Need of Extending Symmetric Primitives

The main limitations of the above-mentioned primitives are:

- The primitives are designed to encrypt small messages. For example, block ciphers cannot process more than a certain amount (n bit) memory at a time.
- Suppose we want to encrypt some large message of 1024 bits using a 128 bit primitive. The simplest extension would be to fragment the message in $\frac{1024}{128} = 8$ blocks, and use the same primitive for 8 times for each of the 128 bit blocks. The main problem here is that, if any block is repeated, then adversary will have some potential advantage that can harm the cryptosystem. This happens because in this case, we are using same key and primitive every time.
- Here we can think of using new keys every time. But that will not be efficient at all, as we need to use a key as long as the message to be encrypted.

3.1 Modes of Operation and their Advantages

To avoid the limitations of symmetric primitives, modes of operation come into play. They use same primitive in such a manner that, reusing the same key would not lead to any attack on the cryptosystem.

A mode of operation often use parameters like Nonce, Associated data, Initialization vector etc. in order to enhance security. A nonce is used in various schemes to encrypt messages, and it is ideally unique for each message. Though there are some schemes which can withstand attacks when a malicious user repeat nonce for adversarial purpose(AEZ, ESTATE). Associated data is used in some authenticated encryption modes(Authenticated encryption with associated data or AEAD). These can be as long as the message to be encrypted, and most of the AEAD modes process these before processing the messages(GIFT-COFB, Romulus, Remus). A Symmetric scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms, described below.

- **Randomized Key Generation Algorithm $\mathcal{K}$**
  Returns a string K. We let $Keys(\mathcal{SE})$ denote the set of all strings that have non-zero probability of being output by K. The members of this set are called keys. We write $K \leftarrow K$ for the operation of executing K and letting K denote the key returned.
- **Encryption algorithm $\mathcal{E}$**
  Might be randomized or stateful, takes a key $K \in Keys(\mathcal{SE})$ and a plaintext $M \in \{0,1\}^*$ to return a ciphertext $C \in \{0,1\}^* \cup \{\perp\}$. We write $C \leftarrow \mathcal{E}_K(M)$ (evaluating C may or may not depend upon the message/cipher blocks encrypted so far) for the operation of executing $\mathcal{E}$ on K and M and letting C denote the ciphertext returned.
- **Deterministic decryption algorithm $\mathcal{D}$**
  It takes a key $K \in Keys(\mathcal{SE})$ and a ciphertext $C \in \{0,1\}^*$ to return some $M \in \{0,1\}^* \cup \{\perp\}$. We write $M \leftarrow D_K(C)$ for the operation of executing $\mathcal{D}$ on K and C and letting M denote the message returned.

The Symmetric scheme $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is said to provide correct decryption if for any key $K \in Keys(\mathcal{SE})$, any sequence of messages $M_1, M_2, \ldots, M_q \in \{0,1\}^*$, and any sequence of ciphertexts $C_1 \leftarrow \mathcal{E}_K(M_1), C_2 \leftarrow \mathcal{E}_K(M_2), \ldots, C_q \leftarrow \mathcal{E}_K(M_q)$ that may arise in encrypting

$M_1, \ldots, M_q$, it is the case that $\mathcal{D}_K(C_i) = M_i$ for each $C_i \neq \perp$.

Modes of operation in view of symmetric key cryptology can be classified into 3 major topics according to their functionality: (I) Encryption modes, (II) Authentication modes, (III) Authenticated Encryption modes. The security of these modes of operations depend upon the security of underlying primitives. The security bound/security goals are established using provable security and reduction arguments.

3.2 Provable Security

When a block cipher is used in a given mode of operation, the resulting algorithm should ideally be about as secure as the block cipher itself. ECB (discussed above) emphatically lacks this property: regardless of how secure the underlying block cipher is, ECB mode can easily be attacked. On the other hand, CBC encryption mode with random IV can be proven to be secure under the assumption that the underlying block cipher is likewise(PRP-CPA) secure. Note, however, that making statements like this requires formal mathematical definitions for what it means for an encryption algorithm or a block cipher to "be secure". Hence, while designing new symmetric modes, specially with some underlying ciphers, we must ensure the newly created mode is secure, with the assumption of the security of the underlying cipher and/or hash function. This general approach to cryptography – proving higher-level algorithms are secure under explicitly stated assumptions regarding their components – is known as provable security.

Provable security refers to any type or level of computer security that can be proved. A system is said to have *provable security* if its security requirements can be stated formally in an adversarial model, as opposed to heuristically, with clear assumptions that the adversary has access to the system as well as enough computational resources. The proof of security, called a *reduction*, is that these security requirements are met provided the assumptions about the adversary's access to the system are satisfied and some clearly stated assumptions about the hardness of certain computational tasks hold. For proving that a given cryptosystem is secure under the assumption that the underlying cipher is secure, there are 2 main techniques:

1. **Game-based Technique:** Most of the modes of operations are proven to be secure using this method. Using this method, we can obtain the security bounds also. The main motivation for using this method is *if a method* **Q** *can be obtained from another method* **P**, *then* "**P** *is insecure* $\implies$ **Q** *is insecure"*. That is, we use techniques of reduction for this method. Various distinguisher based and simulator based security definitions use this technique. For *distinguisher* based games, the adversary tries to distinguish output from the original algorithm and a random bit-string. On the other hand, for the *simulator* based games, the adversary tries to replicate the design of the original algorithm.

2. **H-coefficient Technique:** The "H-coefficient technique" was introduced in 1990 and 1991 by Patarin. Since then, it has been used many times to prove various results on pseudo-random functions and pseudo-random permutations. Recently, it has also been used on key-alternating ciphers (Even-Mansour). This technique proves security by designing distinguisher for a cryptographic algorithm and a pseudorandom functions.[10]

4 Encryption Modes

The block ciphers are schemes for encryption or decryption where a block of plaintext is treated as a single block and is used to obtain a block of ciphertext with the same size. Today, AES (Advanced Encryption Standard) is one of the most used algorithms for block encryption. It has been standardized by the NIST (National Institute of Standards and Technology) in 2001, in order to replace DES and 3DES which were used for encryption in that period. We can use some algorithms for padding block when the plaintext is not a full block, like PKCS5 or PKCS7, it also can defend against PA attack, if we use ECB or CBC mode.

Using block ciphers, tweakable block ciphers and other primitives, various encryption modes can be obtained. Construction of some of the modes are explained in 4.1. The different security claims are described in 4.2.

4.1 Some Examples of Encryption Modes

When encrypting multiple blocks of data using a block cipher, there are various encryption modes that may be employed, each having particular advantages and disadvantages. We will look at some of the encryption modes with symmetric block ciphers here.

1. **CBC : Cipher Block Chaining Mode**

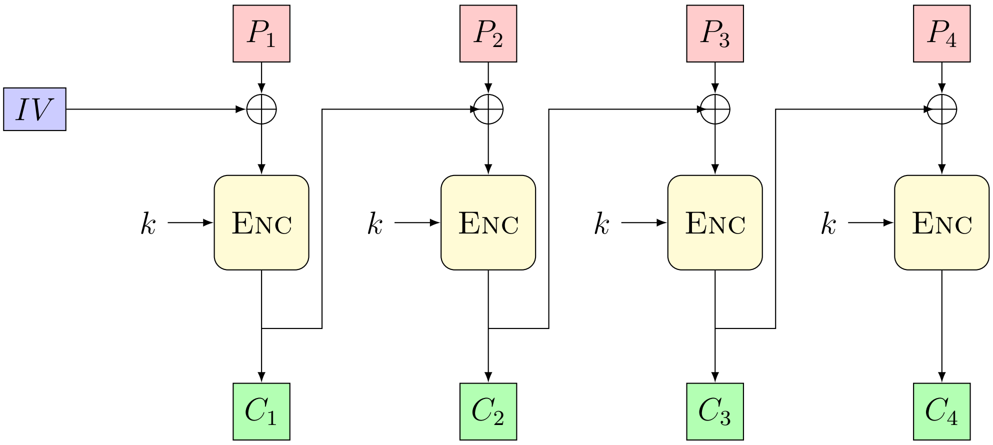
   Cipher Block Chaining (CBC) mode adds a feedback mechanism to the encryption scheme, where the plaintext is exclusively-ORed (XORed) with the previous ciphertext block prior to encryption so that two identical plaintext blocks will encrypt differently. While CBC protects against many brute-force, deletion, and insertion attacks, a single bit error in the ciphertext yields an entire block error in the decrypted plaintext block and a bit error in the next decrypted plaintext block.

   This is an IV-based encryption scheme, the mode is secure for a random IV. Confidentiality is not achieved if the IV is merely a nonce(not random), nor if IV is repeated, as the standard incorrectly suggests to do. No chosen-ciphertext attack (CCA) security is guaranteed. Confidentiality is forfeit in the presence of a correct-padding oracle for many padding methods. Can be used as a building block for CBC-MAC algorithms. No important advantages over CTR mode. This is also used to obtain various MAC tags, which of them are also discussed in our project report.

Algorithm 1 CBC Mode : $CBC_{K,IV}(P)$

```
// $K \in \mathcal{K},\ IV \in \{0,1\}^n,\ P \in (\{0,1\}^n)^+$
$P_1P_2 \dots P_m \leftarrow P$ where $|P_i| = n$
$C_0 \leftarrow IV$
for $i \leftarrow 1$ to $m$ do
    $C_i \leftarrow E_K(P_i \oplus C_{i-1})$
end for
$C \leftarrow C_iC_2 \dots C_m$
return $C$
```

As described in the given algorithm $CBC_{K,IV}(P)$ for CBC Mode, the plain text is divided into blocks. In this mode, IV (Initialization Vector) is used, which can be a random block of text. IV is used to make the ciphertext of each block unique. The first block of plain text and IV is combined using the XOR operation and then encrypted the resultant message using the key and form the first block of ciphertext. The first block of ciphertext is used as IV for the second block of plain text. The same procedure will be followed for all blocks of plain text. At the receiver side, the ciphertext is divided into blocks. The first block ciphertext is decrypted using the same key, which is used for encryption. The decrypted result will be XOR with the IV and form the first block of plain text. The second block of ciphertext is also decrypted using the same key, and the result of the decryption will be XOR with the first block of ciphertext and form the second block of plain text. The same procedure is used for all the blocks. CBC Mode ensures that if the block of plain text is repeated in the original message, it will produce a different ciphertext for corresponding blocks. Note that the key which is used in CBC mode is the same; only the IV is different, which is initialized at a starting point. We hereby present the pictorial description of the CBC Mode.



2. **CTR : Counter Mode**

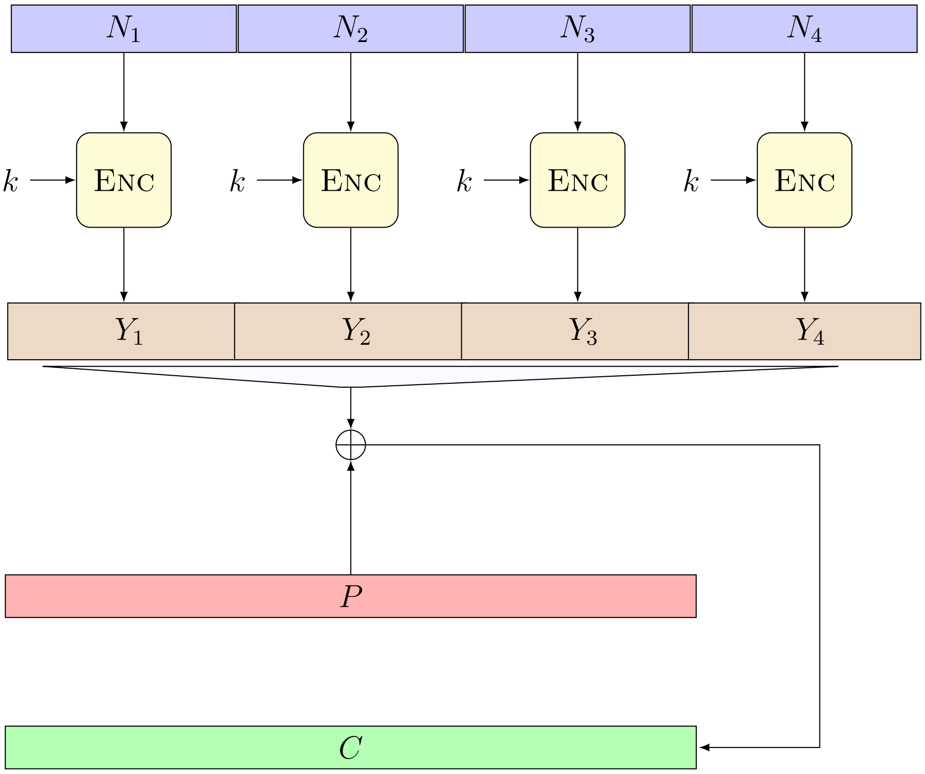
Counter (CTR) mode is a relatively modern addition to block ciphers. Like CFB and OFB, CTR mode operates on the blocks as in a stream cipher; like ECB, CTR mode operates on the blocks independently. Unlike ECB, however, CTR uses different key inputs to different blocks so that two identical blocks of plaintext will not result in the same ciphertext. Finally, each block of ciphertext has specific location within the encrypted message. CTR mode, then, allows blocks to be processed in parallel — thus offering performance advantages when parallel processing and multiple processors are available — but is not susceptible to ECB's brute-force, deletion, and insertion attacks.

An IV-based encryption scheme, the mode achieves indistinguishability from even a nonce IV. As a secure nonce-based scheme, the mode can also be used as a probabilistic encryption scheme, with a random IV. Complete failure of privacy if a nonce(IV) gets reused on encryption or decryption. The parallelizability of the mode often makes it faster, in some settings much faster, than other confidentiality modes. An important building block for authenticated-encryption schemes. Overall, usually the best and most modern way to achieve privacy-only encryption. Used to implement GCM AES.

Algorithm 2 CTR Mode : $CTR_{K,N}(P)$

```
// $K \in \mathcal{K},\ N \in (\{0,1\}^n)^+,\ P \in \{0,1\}^*,\ |N| = \lceil |P|/n \rceil$
$m \leftarrow \lceil |P|/n \rceil$
$(N_1, N_2, \ldots, N_m) \leftarrow N$
for $i \leftarrow 1$ to $m$ do
    $Y_i \leftarrow E_K(N_i)$
end for
$Y \leftarrow MSB_{|P|}\ (Y_iY_2 \ldots Y_m)$
$C \leftarrow P \oplus Y$
return $C$
```

As the name is counter, it uses the sequence of numbers as an input for the algorithm. When the block is encrypted, to fill the next register next counter value is used. Note: the counter value will be incremented by 1. For encryption, the first counter is encrypted using a key, and then the plain text is XOR with the encrypted result to form the ciphertext. The counter will be incremented by 1 for the next stage, and the same procedure will be followed for all blocks. For decryption, the same sequence will be used. Here to convert ciphertext into plain text, each ciphertext is XOR with the encrypted counter. For the next stage, the counter will be incremented by the same will be repeated for all Ciphertext blocks.



3. **ECB : Electronic Codebook Mode**

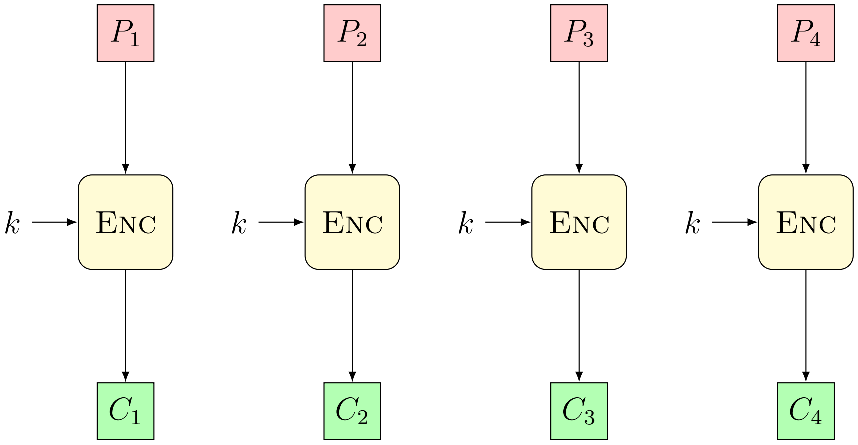
Electronic Codebook (ECB) mode is the simplest, most obvious application: the secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. ECB is susceptible to a variety of brute-force attacks (because of the fact that the same plaintext block will always encrypt to the same ciphertext), as well as deletion and insertion attacks. In addition, a single bit error in the transmission of the ciphertext results in an error in the entire block of decrypted plaintext.

An extension of any block-cipher, the mode enciphers messages that are a multiple of n bits by separately enciphering each n-bit piece. The security properties are weak, the method leaking equality of blocks across both block positions and time. ECB is not regarded as a "general-purpose" operation mode. Generally used to encrypt salary figures of an office, and offers searchability over encrypted data.

Algorithm 3 ECB Mode : $ECB_K(P)$

```
// $K \in \mathcal{K},\ P \in (\{0,1\}^n)^+$
$P_1 P_2 \ldots P_m \leftarrow P$ where $|P_i| = n$
for $i \leftarrow 1$ to $m$ do
    $C_i \leftarrow E_K(P_i)$
end for
$C \leftarrow C_1 C_2 \ldots C_m$
return $C$
```

It is very apparent from the algorithm that this is one of the simplest modes of operation. In this mode, the plain text is divided into a block where each block is 64 bits. Then each block is encrypted separately. The same key is used for the encryption of all blocks. Each block is encrypted using the key and makes the block of ciphertext. At the receiver side, the data is divided into a block, each of 64 bits. The same key which is used for encryption is used for decryption. It takes the 64-bit ciphertext and, by using the key convert the ciphertext into plain text. As the same key is used for all blocks' encryption, if the block of plain text is repeated in the original message, then the ciphertext's corresponding block will also repeat. As the same key used for tor all block, to avoid the repetition of block ECB mode is used for an only small message where the repetition of the plain text block is less.



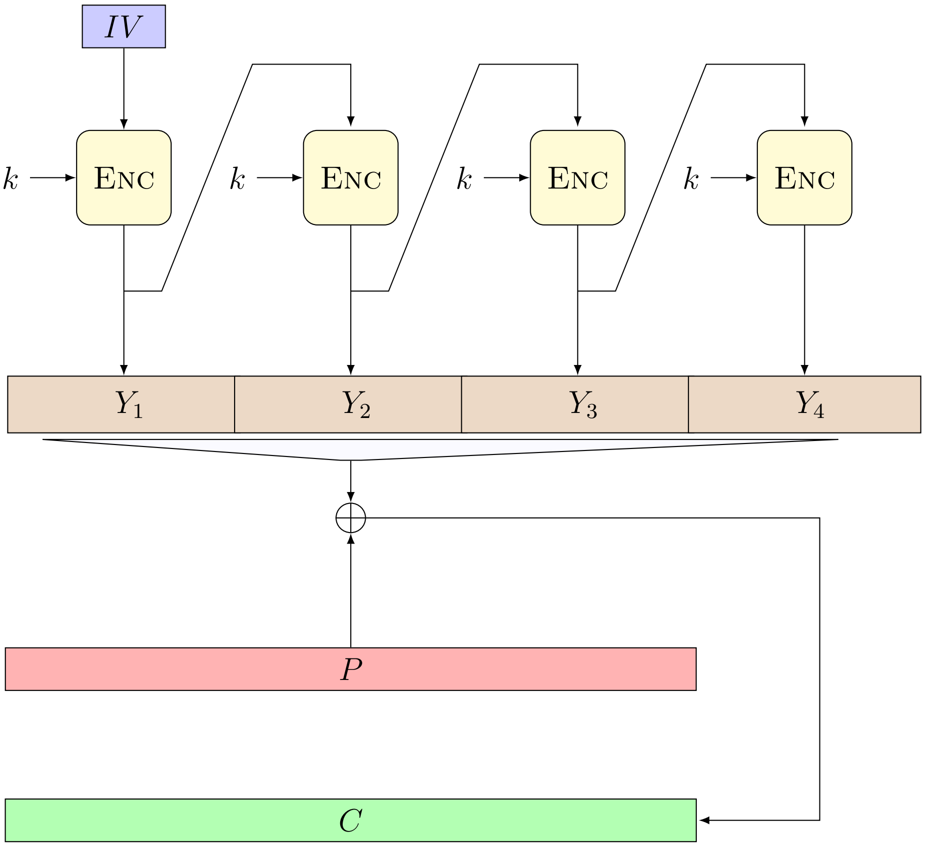
4. **OFB : Output Feedback Mode**

Output Feedback (OFB) mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that generates the keystream independently of both the plaintext and ciphertext bit-stream. In OFB, a single bit error in ciphertext yields a single bit error in the decrypted plaintext.

An IV-based encryption scheme, the mode is secure, with the assumption that a random IV. Confidentiality is not achieved if the IV is a nonce, although a fixed sequence of IVs (eg, a counter) does work fine. Ciphertexts are highly malleable. No CCA security. Natively encrypts strings of any bit length (no padding needed). Can be used to encrypt variable length messages, as the XORing key can be of any length multiple of the blockcipher.

Algorithm 4 OFB Mode : $OFB_{K,\ IV}(P)$

```
// $K \in \mathcal{K},\ IV \in \{0,1\}^n,\ P \in \{0,1\}^*$
$m \leftarrow \lceil |P|/n \rceil$
$Y_0 \leftarrow IV$
**for** $i \leftarrow 1$ **to** $m$ **dos**
    $Y_i \leftarrow E_K(Y_{i-1})$
**end for**
$Y \leftarrow MSB_{|P|}\ (Y_i Y_2 \ldots Y_m)$
$C \leftarrow P \oplus Y$
**return** $C$
```



OFB Mode stands for output feedback Mode. OFB mode is similar to CDB mode; the only difference is in CFB, the ciphertext is used for the next stage of the encryption process, whereas in OFB, the output of the IV encryption is used for the next stage of the encryption process. The IV is encrypted using the key and form encrypted IV. Plain text and leftmost 8 bits of encrypted IV are combined using XOR and produce the ciphertext. For the next stage, the ciphertext, which is the form in the previous stage, is used as an IV for the next iteration. The same procedure is followed for all blocks.

5. **CFB : Cipher Feedback Mode**

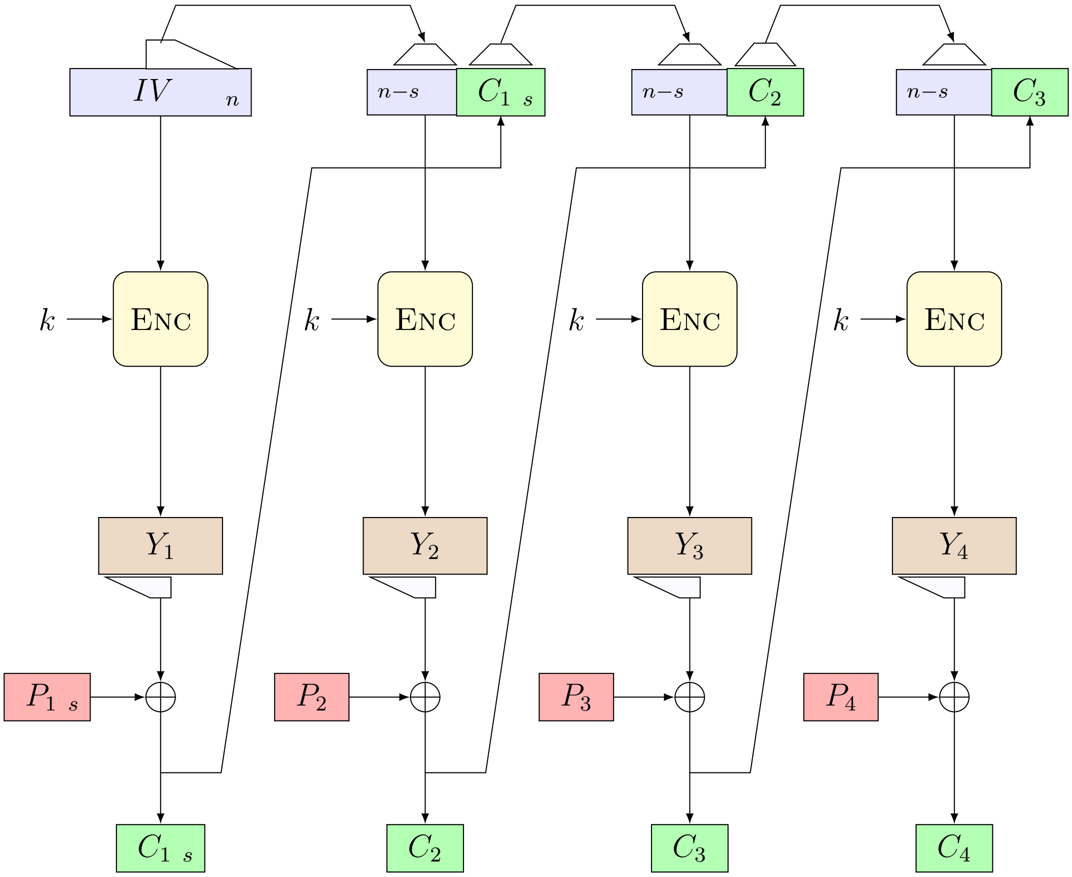
Cipher Feedback (CFB) mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using one-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded. CFB mode generates a keystream based upon the previous ciphertext (the initial key comes from an Initialization Vector [IV]). In this mode, a single bit error in the ciphertext affects both this block and the following one.

An IV-based encryption scheme, the mode is secure assuming a random IV. Confidentiality is not achieved if the IV is predictable. Ciphertexts are malleable, thus, no CCA-security. Scheme depends on a parameter s, $1 \leq s \leq n$, typically $s = 1$ or $s = 8$. Less efficient compared to other modes, for needing one block-cipher call to process only s bits . The mode achieves an interesting "self-synchronization" property; insertion or deletion of any number of s-bit characters into the ciphertext only temporarily disrupts correct decryption.

Algorithm 5 CFB Mode : $CFB_{K,\ IV}(P)$

```
// $K \in \mathcal{K},\ IV \in \{0,1\}^n,\ P \in (\{0,1\}^s)^+$
$P_1 P_2 \ldots P_m \leftarrow P$ where $|P_i| = s$
$X_1 \leftarrow IV$
for $i \leftarrow 1$ to $m$ do
    $Y_i \leftarrow E_K(X_i)$
    $C_i \leftarrow P_i \oplus MSB_s\ (Y_i)$
    $X_{i+1} \leftarrow LSB_{n-s}\ (X_i)\ ||\ C_i$
end for
$C \leftarrow C_1 C_2 \ldots C_m$
return $C$
```

In this mode, the data is encrypted in the form of units where each unit is of 8 bits. Like cipher block chaining mode, IV is initialized. The IV is kept in the shift register. It is encrypted using the key and form the ciphertext. Now the leftmost j bits of the encrypted IV is XOR with the plain text's first j bits. This process will form the first part of the ciphertext, and this ciphertext will be transmitted to the receiver. Now the bits of IV is shifted left by j bit. Therefore the rightmost j position of the shift register now has unpredictable data. These rightmost j positions are now filed with the ciphertext. The process will be repeated for all plain text units.



4.2 Security Definitions for Encryption Modes

1. **Semantic-CPA Security (Sem-CPA):**

   We first define the basic structure of a semantic security game. For defining security of a mode of operation, the queries can be done with large messages and choice for initialization vector IV. The basic structure of the game is similar to IND-CPA game defined for symmetric-key primitives. The game can be stated in 2 different ways, viz., *Left-or-Right (LOR)* and *Real-or-Random (ROR)*. For both the games, each time, both the queried plaintexts will have same length. For LOR security, a message cannot be queried more than once at the same component(*i.e.*, adversary cannot query with same (IV, M) or (IV', M')) more than once. Consider $\mathcal{SE} = (\mathcal{E}, \mathcal{D})$ be the mode of operation.

Left or Right Security: The Sem-CPA game for LOR security is as follows:

Adversary$\mathcal{A}$		**Challenger** $\mathcal{C}$	
		choose $K \xleftarrow{\$} \mathcal{K}$	
For $i = 1, 2, \ldots, n$ $M_i, M_i' \leftarrow \mathcal{M}$ $IV_i, IV_i' \leftarrow \mathcal{IV}$			
	$\xrightarrow[(IV_i,M_i)]{(IV_i',M_i')}$		
		choose $b \xleftarrow{\$} \{0,1\}$ **For** $i = 1, 2, \ldots, n$	
		if $b = 0$ **Left**$_{\mathcal{E}_K^{IV_i,IV_i'}}(M_i, M_i')$ $C_i \leftarrow \mathcal{E}_K.Enc(IV_i, M_i)$	if $b = 1$ **Right**$_{\mathcal{E}_K^{IV_i,IV_i'}}(M_i, M_i')$ $C_i \leftarrow \mathcal{E}_K.Enc(IV_i', M_i')$
	$\xleftarrow{C_i}$		
$b' \leftarrow \mathbf{Exp}^{LOR}(\mathcal{A})$ output b'			

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with $Input = (M_1, IV_1, M_2, IV_2, ..., M_n, IV_n; M_1', IV_1', M_2', IV_2', ..., M_n', IV_n'; C_1, C_2, ..., C_n)$ where $M_i = M_{1i}M_{2i}\ldots M_{ri}, M_i' = M_{1i}'M_{2i}'\ldots M_{ri}'$ and $\mathcal{A}$ is a stateful adversary depending upon a choice of bit 0 or 1:

Case 1:	**Case 2:**
$\mathbf{Exp}^{LOR-1}(\mathcal{A})$	$\mathbf{Exp}^{LOR-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,1)}(Input)$	$b' \leftarrow \mathcal{A}_{\mathcal{E}}^{LR(.,.,0)}(Input)$

The LOR-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{lor-sem-cpa}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{LOR-0}(\mathcal{A}) = 1]|$$

.

Real or Random Security: The Sem-CPA game for ROR security is almost similar to the LOR game. Here the adversary queries with a single message every time:

Decide Encryption mode $\mathcal{E}$

$$\begin{array}{lcc}
\textbf{Adversary}\mathcal{A} & & \textbf{Challenger } \mathcal{C} \\
& & \text{choose } K \xleftarrow{\$} \mathcal{K} \\
& & \text{choose } F \xleftarrow{\$} \mathcal{PRP} \\
\textbf{For } i = 1, 2, \dots, n & & \\
\quad M_i \leftarrow \mathcal{M} & & \\
\quad IV_i \leftarrow \mathcal{IV} & & \\
& \xrightarrow{(IV_i, M_i)} & \\
& & \text{choose } b \xleftarrow{\$} \{0, 1\} \\
& & \textbf{For } i = 1, 2, \dots, n \\
& & \begin{array}{c|c}
\text{if } b = 0 & \text{if } b = 1 \\
\textbf{Real}_{\mathcal{E}_K^{IV_i}}(M_i) & \textbf{PRP}_{\mathcal{E}_K^{IV_i}}(M_i) \\
C_i \leftarrow \mathcal{E}_K.Enc(IV_i, M_i) & C_i \leftarrow F(IV_i, M_i)
\end{array} \\
& \xleftarrow{C_i} & \\
b' \leftarrow \textbf{Exp}^{ROR}(\mathcal{A}) & & \\
\quad \text{output } b' & &
\end{array}$$

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\textbf{Exp}^{CPA-1}$ and $\textbf{Exp}^{CPA-0}$ with $Input = (M_1, IV_1, M_2, IV_2, ..., M_n, IV_n; C_1, C_2, ..., C_n)$ where $M_i = M_{1i}M_{2i}\dots M_{ri}$:

Case 1:	**Case 2:**
$\textbf{Exp}^{ROR-1}(\mathcal{A})$	$\textbf{Exp}^{ROR-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}_{\mathcal{E}}(Input)$	$b' \leftarrow \mathcal{A}_F(Input)$

The ROR-advantage of $\mathcal{A}$ is defined as:

$$Adv^{ror-sem-cpa}(\mathcal{A}) = |Pr[\textbf{Exp}^{ROR-1}(\mathcal{A}) = 1] - Pr[\textbf{Exp}^{ROR-0}(\mathcal{A}) = 1]|$$

.

2. **Semantic CCA Security:** We can define similar Left-or-Right and Real-or-Random security in view of sem-CCA by redefining the games. This time, we have to consider that the adversary has the privilege of querying with ciphertexts blocks.

   For both the games, each time, both the queried plaintexts will have same length. For LOR security, a message cannot be queried more than once at the same component(*i.e.*, adversary cannot query with same (IV, M) or (IV', M')) more than once.

3. **TSPRP Security:**The TSPRP game for a tweakable HCTR mode $\tilde{H}$ is as follows:

Adversary$\mathcal{A}$		**Challenger** $\mathcal{C}$	
		choose $K \xleftarrow{\$} \mathcal{K}$	
		choose $F \xleftarrow{\$} \mathcal{PRP}$	
For $i = 1, 2, \dots, n$			
$M_i \leftarrow \mathcal{M}$			
$IV_i \leftarrow \mathcal{IV}$			
For $j = 1, 2, \dots, n'$			
$C_j \leftarrow \mathcal{C}$			
$IV'_j \leftarrow \mathcal{IV}$			
	$\xrightarrow[(IV'_j, C'_j)]{(IV_i, M_i)}$		
		choose $b \xleftarrow{\$} \{0, 1\}$	
		For $i = 1, 2, \dots, n$	
		if $b = 0$	if $b = 1$
		$\mathbf{Real}_{\mathcal{E}_K^{IV_i}}(M_i)$	$\mathbf{PRP}_{\mathcal{E}_K^{IV_i}}(M_i)$
		$C_i \leftarrow \tilde{H}_K.Enc(IV_i, M_i)$	$C_i \leftarrow F(IV_i, M_i)$
		$M'_j \leftarrow \tilde{H}_K^{-1}.Enc(IV'_j, C'_j)$	$M'_j \leftarrow F(IV'_j, C'_j)$
	$\xleftarrow[M'_j]{C_i}$		
$b' \leftarrow \mathbf{Exp}^{ROR}(\mathcal{A})$			
output b'			

$\mathcal{A}$ wins the game if $b' = b$.

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with
$Input = (M_1, IV_1, M_2, IV_2, ..., M_n, IV_n; C_1, C_2, ..., C_n$
$C'_1, IV'_1 \dots C'_j, IV'_j; M'_1, \dots M'_j)$ where $M_i = M_{1i}M_{2i} \dots M_{ri}, C'_J = C'_{1j}C'_{2j}...C'_{rj}$:

Case 1:	**Case 2:**
$\mathbf{Exp}^{TSPRP-1}(\mathcal{A})$	$\mathbf{Exp}^{TSPRP-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$K \xleftarrow{\$} \mathcal{K}$
$b' \leftarrow \mathcal{A}_{\tilde{H}}(Input)$	$b' \leftarrow \mathcal{A}_F(Input)$

The TSPRP-advantage of $\mathcal{A}$ is defined as:

$$Adv_{\mathcal{F}}^{lor-sem-cpa}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{F}}^{TSPRP-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{F}}^{TSPRP-0}(\mathcal{A}) = 1]|$$

.

4. **Insecurity of Fixed IV**

Suppose the adversary $\mathcal{A}$ is querying the challenger $\mathcal{C}$ with messages consisting of length $2n$(the blockcipher encrypts one block of length n at a time). Suppose the fixed IV being used is IV_0. Now, $\mathcal{A}$ queries with two messages $m_0 = m_{00}||m_{01}$ and

$m_1 = m_{10}||m_{11}$(where length of each m_{ij} is n), such that $m_{00} \neq m_{10}$. $\mathcal{A}$ queries the encryption oracle with $m'_0 = m_{00}||m_{01} \oplus 1$ and $m'_1 = m_{10}||m_{11} \oplus 1$. Clearly these queries are valid. Suppose the oracle answers $c'_0 = c_{00}||c_{01}$ and $c'_1 = c_{10}||c_{11}$ respectively. Now, $\mathcal{C}$ chooses b=0 or 1 respectively and gives $c = Enc_{IV_0,K}(m_b) = c_0||c_1$ to $\mathcal{A}$. If $c_0 = c_{00}$ then $\mathcal{A}$ returns 0, and otherwise, i.e., if $c_0 = c_{10}$, $\mathcal{A}$ returns 1. Thus the case of fixed IV is not secure for any mode of operation.

5. **Nonce Respecting and Misusing Adversary**

   In nonce respecting case, IV is cannot be reused for two distinct queries. Still, the adversary is allowed to manipulate the IV to get a desired one. Whereas in case of nonce misusing adversarial model, the adversary is free to choose any nonce. Thus the adversary can repeat a nonce.

4.3 Implementation Area and Throughput: Definitions

Thought these parameters are defined in this section, these will remain relevent for authentication and authenticated encryption modes also.

1. **State Size** The total size of variables that are being used from beginning to end of the mode of operation, along with the block-cipher size. For example, for CTR mode, an encryption mode that we thoroughly discussed in the next section, if size of $CTR = t$, size of block-cipher= n, and size of key K = k, then state size will be $n + k$, as for CTR mode, we only need to remember the block cipher and the key to encrypt every block.

2. **Rate** Suppose r-bit block can be encrypted using n-bit block-cipher mode X. Then rate of X=$\frac{r}{n}$. All the modes including ECB, CBC, CTR etc processes 128-bit data chunks with 128 bit block-ciphers. That is, they all calls the block cipher only once for encrypting each chunk. So each one will have a rate =1.

3. **Parallelizable** Suppose a message $M := M[1]M[2]\dots M[m]$ is being encrypted. If the i^{th} block's encryption is independent of any block(s) encrypted so far, then the mode is parallelizable. This means, that the mode can be used to encrypt arbitrarily large data chunks. This property enhances the efficiency of the mode. That is, if a mode is parallelizable, its speed increases.

4. **Inverse Free** A mode X is inverse free if at the time of decryption of X, the block-cipher decryption(or inversion operation) is not necessary. For example, for OFB mode, we can obtain decryption of $C := C[1]C[2]\dots C[m]$ as $M := M[1]M[2]\dots M[m]$ where $M[i] = C[i] \oplus E_k(O[i-1])$. Thus, D_k is not required here. OFB is inverse free mode.

5. **Online** A mode is called online if it is not necessary to process the whole data prior to processing the i^{th} block of data. That is, encrypting the message block $M[i]$ does not require processing of $M[i+k], M[i+k+1], ..., M[n]$ for fixed k.

4.4 Comparison Table for Encryption Modes

The second column explains assumption on the underlying primitive's security. The third column is the security type of the mode. The column **Bound** gives the security bound of the mode. Here, all the Security bounds apply for Nonce-Respecting case unless otherwise specified. Clearly, IV serves as the nonce here.

Mode	Assumption	Security	IV type	Bound	State Size	Rate	Prop.	Key Size
CBC	PRP-CPA	Sem-CPA(ROR)	Random IV	$\frac{\sigma^2}{2^n}$	$\lvert E_K\rvert + \lvert K\rvert$	1	O	$\lvert K\rvert$
CTR	PRP-CPA	Sem-CPA(ROR)	Random IV	$\frac{\sigma^2}{2^{n+1}}$	$\lvert E_K\rvert + \lvert K\rvert$	1	O,I,P	$\lvert K\rvert$
OFB	PRP-CPA	Sem-CPA(ROR)	random IV	$\frac{\sigma^2}{2^n}$	$\lvert E_K\rvert + \lvert K\rvert$	1	O,I	$\lvert K\rvert$
CFB	PRP-CPA	Sem-CPA(ROR)	random IV	$\frac{q(q-1)}{2^{n+1}}$	$\lvert E_K\rvert + \lvert K\rvert$	1	O,I	$\lvert K\rvert$
ECB	any	Insecure	not used	NA	$\lvert E_K\rvert + \lvert K\rvert$	1	O,P	$\lvert K\rvert$
Tweakable HCTR	TPRP	TSPRP	random	$2(\frac{\sigma}{2^n} + \frac{q\sigma}{2^{m+n}})$ $+\frac{q^2}{2^{m+n}}$	$\lvert \tilde{E}_K\rvert + \lvert K\rvert +$ $\lvert IV\rvert + \lvert H_1\rvert + \lvert H_2\rvert$	1	P	$\lvert K\rvert + \lvert K_h\rvert$
HCTR	PRP-CCA	Sem-CCA(ROR)	Nonce	$\frac{4.5\sigma^2}{2^n}$	$\lvert \tilde{E}_K\rvert + \lvert K\rvert$ $+\lvert N\rvert + \lvert H\rvert$	1	P	$\lvert K\rvert$
XEX	PRP-CCA	Sem-CCA-ROR	Random	$\frac{4.5q^2}{2^n}$	$\lvert E_K\rvert + \lvert K_1\rvert + \lvert I\rvert$	1	O,P	$\lvert K_1\rvert + \lvert K_2\rvert$
LRW	PRP-CCA	Sem-CCA-ROR	Nonce	$\frac{7.5q^2}{2^n}$	$\lvert E_K\rvert + \lvert K_1\rvert + \lvert K_2\rvert$	1	O,P	$\lvert K_1\rvert + \lvert K_2\rvert$
CMC	PRP-CPA	Sem-CPA-ROR	Random	$\frac{7\sigma^2}{2^n}$	$\lvert M\rvert + \lvert E_K\rvert + \lvert K\rvert$	$\frac{1}{2}$		$\lvert K\rvert$
EME	PRP-CPA	Sem-CPA-ROR	Random	$\frac{7\sigma^2}{2^n}$	$\lvert E_K\rvert + \lvert K\rvert + \lvert SP\rvert +$ $\lvert SC\rvert + \lvert M\rvert + \lvert L\rvert$	$\frac{1}{2}$	P	$\lvert K\rvert$

Table 1: Comparison Table for Encryption Modes.

K, K_i	Keys Used in the Schemes	k	key Size
n	Size of Underlying Block Cipher	I	Inverse-free
q	Total No of Queries	O	Online
σ	No of Blocks Queried	P	Parallelizable

4.5 Suitable Data Specifications

In this section, we shall discuss performance of some important Encryption modes based on specifications of data to be encrypted. We shall conclude which type of data best fits for the modes.

CBC	Small state size. Optimal rate.
CTR	Small state size, optimal rate. Online, parallel, inverse free.
Tweakable HCTR	Beyond birthday bound secure.
HCTR	Beyond birthday bound secure. Small key size.
LRW	Online. Beyond birthday bound secure. Optimal rate.
EME	Parallelizable. Small key size needed.

5 Authentication Modes

A message authentication code (MAC), sometimes known as a tag, is a short piece of information used to authenticate a message – in other words, to confirm that the message came from the stated sender (its authenticity) and has not been forged. The MAC value protects a message's data integrity, as well as its authenticity, by allowing verifier (who also possess the secret key) to detect any changes to the message content. Informally, a message authentication code system consists of three algorithms - (I) A key generation algorithm selects a key from the key space uniformly and randomly. (II) A signing algorithm efficiently returns a tag given the key and the message. (III) A verification algorithm efficiently verifies the authenticity of the message given the key and the tag. That is, return accepted when the message and tag are not tampered with or forged, and otherwise return $\perp$ or rejected.

A secure MAC must refrain adversarial attempts to forge tags, for arbitrary, selected, or all messages. It should be computationally impossible to compute a valid tag of the given message without knowledge of the key, even if for the worst case, we assume the adversary knows the tag of any message but the one in question. We are primarily interested in the standard complexity theoretic model for MACs, in fact many attacks implicitly assume the real-or-ideal model for stating their results. We shall first describe some typical examples of MAC modes and then go into detailed comparison between around thirty MAC Modes.

5.1 Some Examples of Authentication Modes

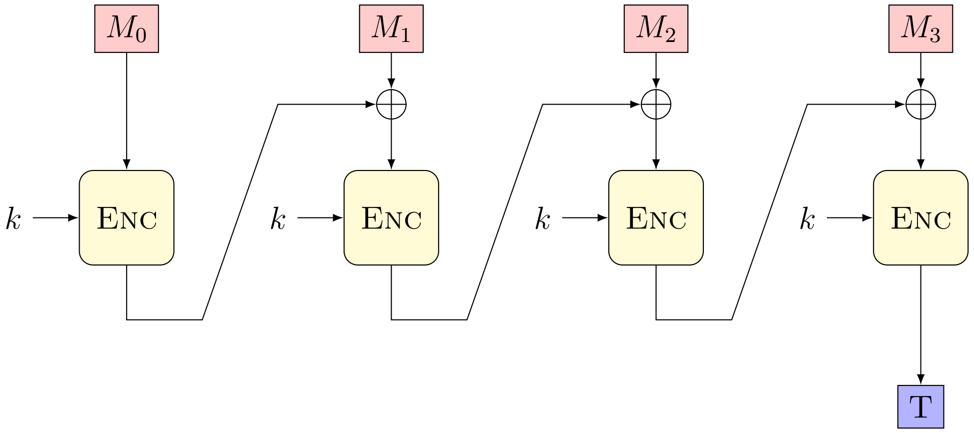
1. **Raw CBC-MAC**

   Document ISO/IEC 9797-1:1999 is said to define some six different MAC algorithms, all CBC-MAC variants, referred to in the specification as MAC Algorithms 1–6. Each scheme takes as input a key and a string of essentially arbitrary length and produces a tag T. The modes are parameterized not only by the underlying blockcipher $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ and tag length τ , where $1 \leq \tau \leq n$, but on one or two further parameters like the padding method and, in some cases, implicitly, the key-separation method, too. These different MAC algorithms are based on the so-called Raw CBC-MAC Algorithm. The algorithm is described in the following.

Algorithm 6 Raw CBC-MAC : $CBCMAC_K(M)$

```
if $|M|$ is not a positive multiple of $n$ then
    return INVALID
end if
$M_1 M_2 \ldots M_m \leftarrow M$
$C_0 \leftarrow 0^n$
for $i \leftarrow 1$ to $m$ do
    $C_i \leftarrow E_K(M_i \oplus C_{i-1})$
end for
return $C_m$
```

As we can see from the discussed algorithm, the idea is almost similar to the CBC Encryption Mode. The only difference is that at every iteration there, we get a corresponding cipher text block from each plaintext block, whether here we are getting the tag at the end of the process. A pictorial representation of the above algorithm is given in the following.



The algorithm is classical, underlying the techniques described not only in ISO 9797-1 but also ANSI X9.9, ANSI X9.19, FIPS 81, FIPS 113, ISO 8731-1, ISO 9807, and ISO 9797. The raw CBC-MAC is to be considered as a derivative of the CBC encryption scheme, the message M that we wish to MAC is CBC-encrypted, but only the final block of ciphertext is returned as the MAC. While useful and classical, the raw CBC-MAC has some major restrictions also.

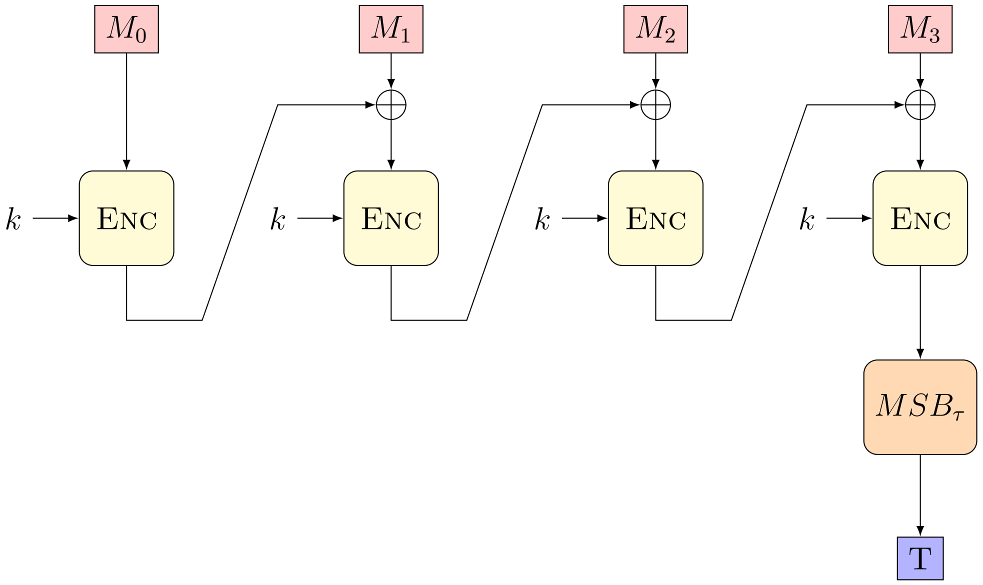
In the next sections, we discuss four more MACs, all of them based on the CBC-MAC. Some are provably secure as VIL PRFs, some as FIL PRFs, and some have no provable security. Some of the schemes admit damaging attacks. Some of the modes are dated. Key-separation is inadequately attended to for the modes that have it. It would also be fine to adopt none of these modes, in favor of CMAC. Some of the ISO 9797-1 MACs are widely standardized and used, especially in banking.

2. **CBC-MAC Algorithm 1**

This first algorithm, the basic CBC-MAC aka CBC-MAC Algorithm 1 is the raw CBC-MAC except for the inclusion of padding at the beginning and truncation at the end. The detailed algorithm and the pictorial representation of the CBC-MAC Algorithm is hereby given. $CBCMAC_K(M)$ is the Raw CBC-MAC Algorithm described in the previous section.

Algorithm 7 MAC Algorithm 1 : $ALG1_K\ (M)$

```
$M \leftarrow Pad(M)$
$Tag \leftarrow CBCMAC_K\ (M)$
$T \leftarrow MSB_\tau\ (Tag)$
return $T$
```



The method seems to begin with ANSI X9.9, where Encryption model is DES and $\tau = 32$. Assuming that the Encryption model is a PRP, the VIL security of CBC-MAC Algorithm 1 depends on the choice of Pad and τ. The provable security results for MAC Algorithm 1 (FIL-security with padding method-1, or VIL-security with padding-method 3) go only as far as the birthday bound. It is unfortunate that CBC-MAC Algorithm 1's basic security properties depend in a complicated way on the values of Pad and τ , even assuming that the blockcipher for Encryption is a good PRP. One cannot make simple statements, like 'CBC-MAC Algorithm 1 is a good PRF assuming the underlying blockcipher is a good PRP' since the statement is not always true.

3. CBC-MAC Algorithm 2

The CBC-MAC Algorithm 2 are basically two schemes, or rather considered to be two variants. In the more basic variant, ALG2A, the MAC key has $2k$ bits; keys are of the form $K \mid\mid K'$ where $K, K' \in \{0,1\}^k$ are keys for the underlying blockcipher. The key K is used to encrypt the raw CBC-MAC prior to truncation. Mode ALG2B is identical except that some key derivation function, which we call Sep, defines K and K' from given key J. Here is the algorithm.

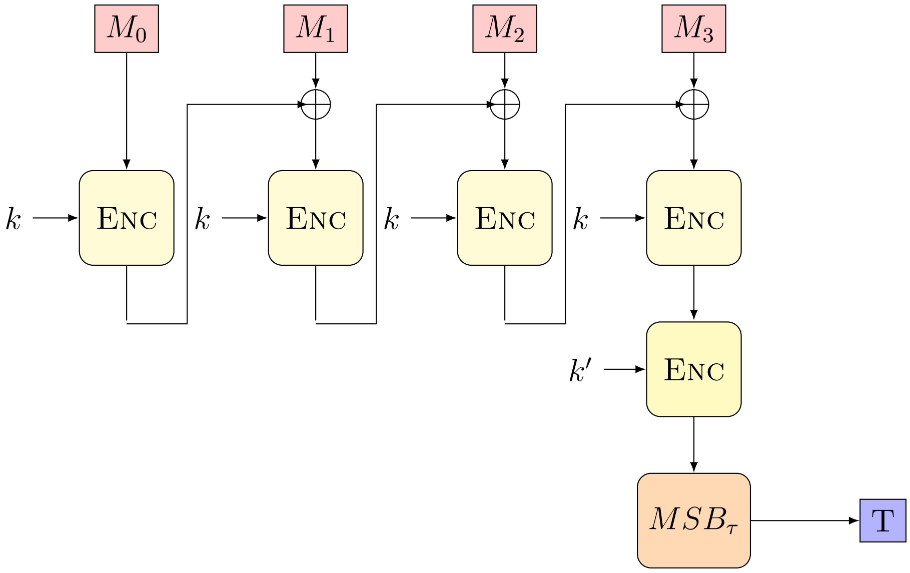
Algorithm 8 MAC Algorithm 2

MAC Algorithm 2A : ALG2A$_{K,K'}$ (M)

```
$M \leftarrow Pad(M)$
$TAG \leftarrow CBCMAC_K\ (M)$
$Tag \leftarrow E_{K'}\ (TAG)$
$T \leftarrow MSB_\tau\ (Tag)$
return $T$
```

MAC Algorithm 2B : ALG2B$_J$ (M)

```
$K||K' \leftarrow Sep(J)$
return $ALG2A_{K,K'}\ (M)$
```



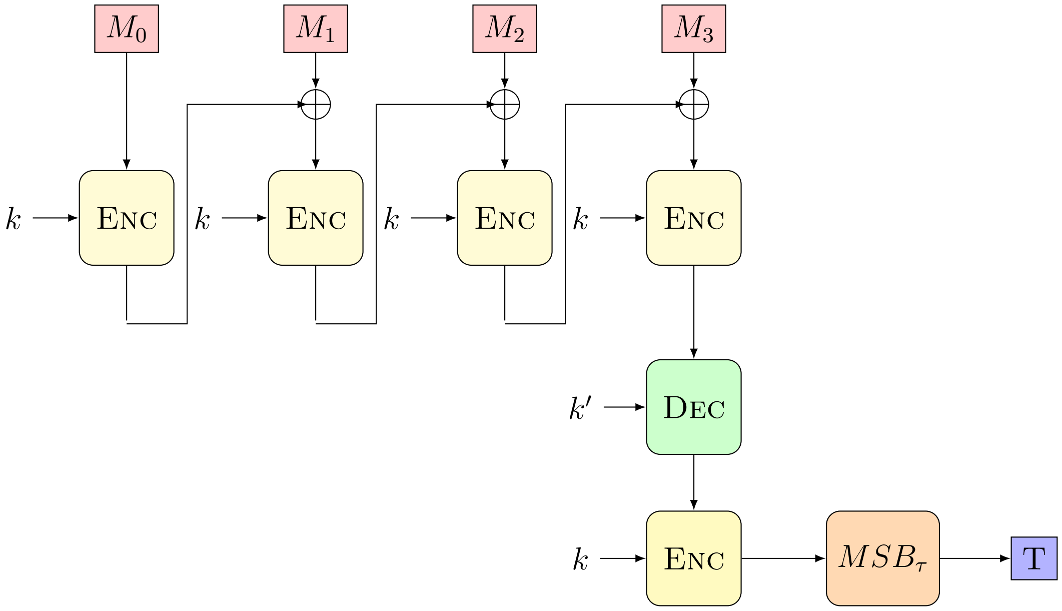
As for ALG2B, a fundamental question to ask is what property the key-separation algorithm *Sep* needs to have. The ISO standard says nothing in this direction, and what it does say goes against achieving provable-security guarantees, and is even self-contradictory. The spec begins by saying that the value of K' may be derived from K in such a way that the two are different. This would suggest that in the algorithm ALG2B, it should say $K' \leftarrow Sep(K)$, with K, not some (invented) J, as the key. This is though another way to do the key-separation is to derive both K and K' from a common master key. A better approach for key separation would have been to use a more standard and provable security friendly key-separation technique, like $K = E_J(C1)$ and $K' = E_J(C2)$ for distinct n-bit constants $C1$ and $C2$. Use of such a technique would allow one to claim provable security for ALG2B under the corresponding conditions on ALG2A.

4. **CBC-MAC Algorithm 3**

This method dates to ANSI X9.19 (1986), a standard for retail banking; consequently, the blockcipher mode has been called the retail MAC. It is like CBC-MAC Algorithm 2 but, rather than seal the last block by enciphering the raw CBC-MAC, the designers instead switch to triple encryption for sealing it. The Key-separation algorithm is no longer a permitted option; the underlying key must be 2k bits. Let us first go through the algorithm and the pictorial representation if the CBC-MAC Algorithm 3.

Algorithm 9 MAC Algorithm 3 : $ALG3_{K,K'}\ (M)$

```
$TAG \leftarrow CBCMAC_K\ (M)$
$Tag \leftarrow E_K\ (D_{K'}\ (TAG))$
$T \leftarrow MSB_\tau\ (Tag)$
**return** $T$
```



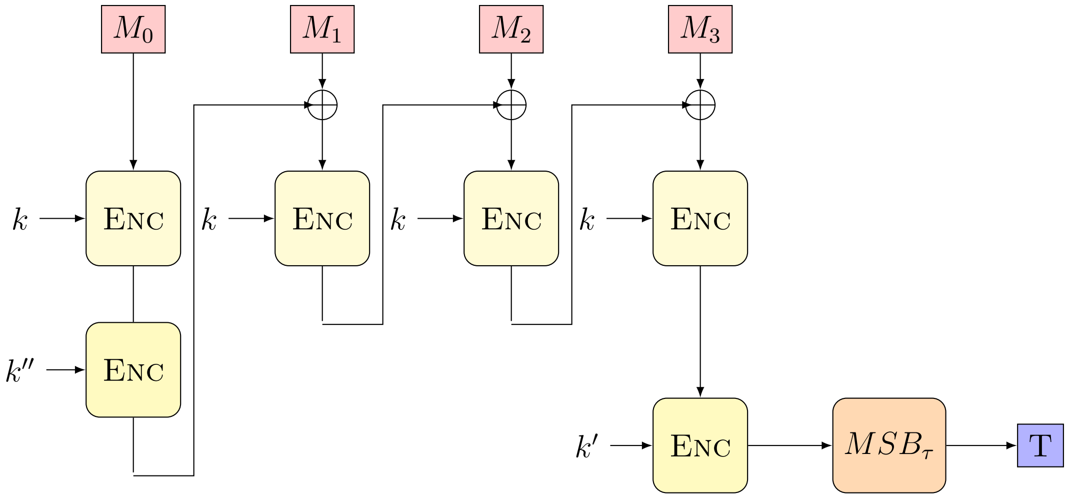
Assuming the underlying blockcipher is a secure PRP, CBC-MAC Algorithm 3 is VIL-secure, again to the birthday bound, assuming type-2 padding. Also, CBC-MAC Algorithm 3 with type-1 padding inherits FIL-security from the raw CBC-MAC : the application of $E_K \circ D_K$ to the CBC-MAC does no harm, in the information-theoretic setting of the analysis, since the composed permutation will be independent of the encryption method. Similarly, it inherits provable-security to the birthday bound with type-3 padding. Neither result is so-called interesting, but these padding schemes make little sense for the inventions of CBC-MAC Algorithms 2 or 3. One common question may arise regarding the reason of CBC-MAC Algorithm 3's inclusion, given its similarity to MAC Algorithm 2 and its needing one extra blockcipher call. One answer is the apparently improved resistance to exhaustive key search. CBC-MAC Algorithm 3 seems better with respect to exhaustive key search than MAC Algorithm 2 just as triple-encryption appears better in this regard than double-encryption.

5. **CBC-MAC Algorithm 4**

CBC-MAC Algorithm 4 was designed by Knudsen-Preneel, keeping DES as the underlying blockcipher. Hence, the algorithm goes by the name MacDES. Here is the algorithm.

Algorithm 10 MAC Algorithm 4 : $ALG4_{K,K'}\ (M)$

```
$K'' \leftarrow Sep\ (K||K')$
$M \leftarrow Pad\ (M)$
if $|M| < 2n$ then
    return INVALID
end if
$C_1 \leftarrow E_{K''}\ (E_K\ (M_1))$
for $i \leftarrow 2$ to $m$ do
    $C_i \leftarrow E_K(M_i \oplus C_{i-1})$
end for
$Tag \leftarrow E_{K'}\ (C_m)$
$T \leftarrow MSB_\tau\ (Tag)$
return $T$
```



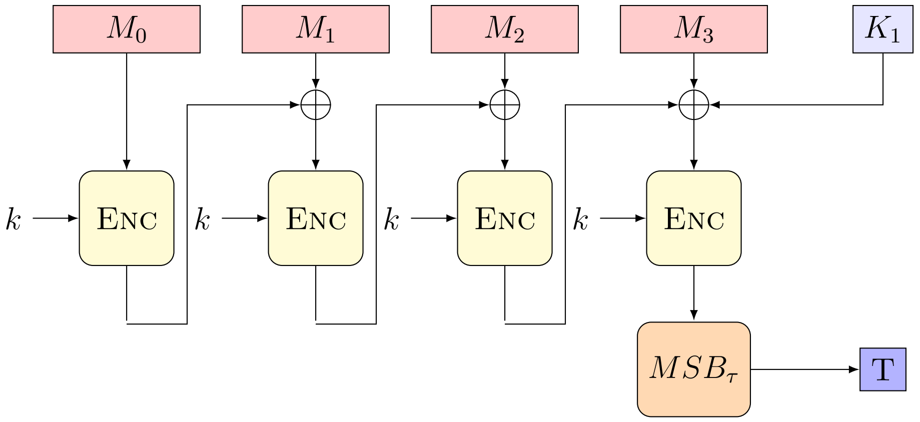
One can argue the efficiency of one of MAC Algorithms 1–3 on triple-DES instead of single-DES, but MAC Algorithm 4 is more efficient as it aims to buy enhanced key length with two blockcipher calls more than the raw CBC-MAC. Though it seems like the scheme needs $3k$ length key, but actually, k'' is obtained from k and k'. So originally, this scheme requires $2k$ length key only.

6. **CMAC mode**

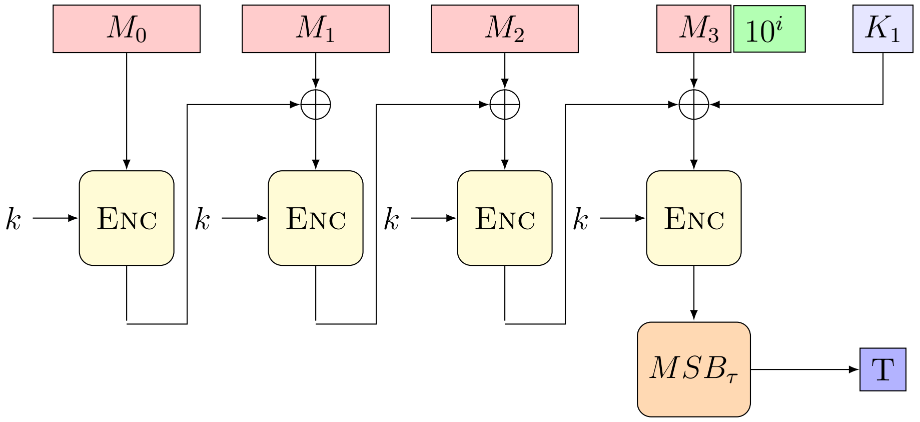
This chapter looks at the CMAC mode (Cipher-based MAC) of NIST Recommendation SP 800-38B. CMAC is a simple and clean variant of the CBC-MAC. Used with a strong 128-bit blockcipher, the scheme enjoys good provable-security properties—reasonable bounds under standard assumptions—with no significant identified flaws. Recently, its provable security bounds have improved, furthering our understanding of this mode.

Algorithm 11 CMAC Algorithm : $CMAC_K\ (M)$

```
$K1 \leftarrow$ dbl $(E_K(0^n))$
$K2 \leftarrow$ dbl $(K_1)$
if $|M| \in \{n, 2n, 3n, \dots\}$ then
    $K' \leftarrow K1$
    $P \leftarrow M$
else
    $K' \leftarrow K2$
    $P \leftarrow M10^i$ where $i = n - 1 - (|M| \mod n)$
end if
$M_1 M_2 \dots M_m \leftarrow M$ where $|M_x| = n\ \forall\ 1 \leq x \leq m$
for $i \leftarrow 1$ to $m$ do
    $C_i \leftarrow E_K(M_i \oplus C_{i-1})$
end for
$T \leftarrow MSB_\tau\ (C_m)$
return $T$
```



CMAC with Full Final Block (i.e. n divides $|M|$)



CMAC with Partial Final Block (where we need the Padding)

CMAC is also a MAC based Authentication Mode, provably secure up to the birthday bound as a (VIL) PRF, assuming the underlying blockcipher is a good PRP. This has essentially minimal overhead for a CBCMAC-based scheme. Inherently serial nature is a problem in some application domains, and use with a 64-bit blockcipher would necessitate occasional re-keying. This is cleaner than the ISO 9797-1 collection of MACs. This mode takes additional key K_1 for XORing with last block, alongside obtaining the MAC according to CBC-MAC 2 with padding type 2. The values $K1$ and $K2$ are derived from K by doubling the key once or twice, where to double means to multiply by the point in $GF(2^n)$ whose representation is $u = 0^{n-1}10$.

7. HMAC Mode

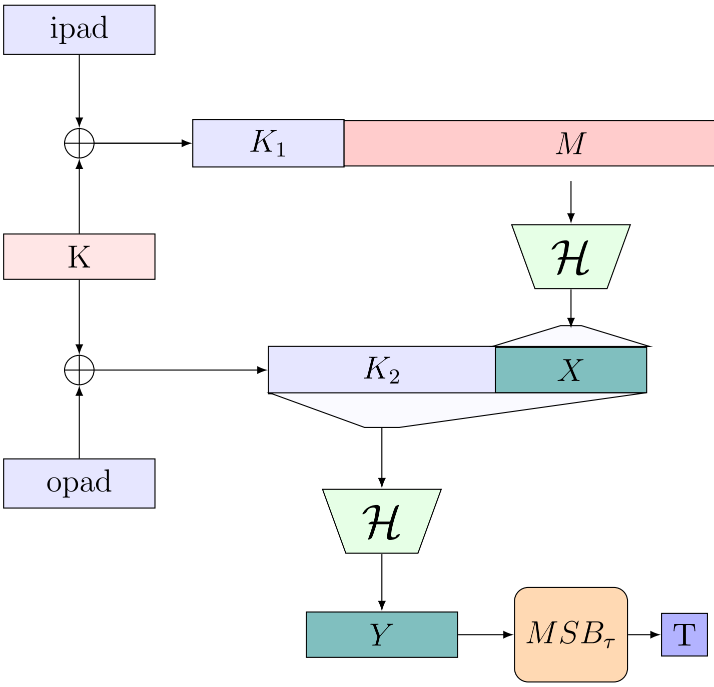
HMAC, aka Keyed-Hash or Hash-based MAC is a specific type of MAC involving a cryptographic hash function and a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authenticity of a message. HMAC can provide message authentication using a shared secret instead of using digital signatures with asymmetric cryptography. It trades off the need for a complex public key infrastructure by delegating the key exchange to the communicating parties, who are responsible for establishing and using a trusted channel to agree on the key prior to communication. Any cryptographic hash function, such as SHA-2 or SHA-3, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-X, where X is the hash function used.

Algorithm 12 HMAC Algorithms

```
HMAC0 Algorithm : $\mathrm{HMAC0}_K\ (M)$
  $K1 \leftarrow K \oplus \mathrm{ipad}$
  $K2 \leftarrow K \oplus \mathrm{opad}$
  $X \leftarrow H(K1 \mid\mid M)$
  $Y \leftarrow H(K2 \mid\mid X)$
  $T \leftarrow \mathrm{MSB}_\tau(Y)$
  return $T$

HMAC1 Algorithm : $\mathrm{HMAC1}_K\ (M)$
  $K \leftarrow K \mid\mid 0^{b-c}$
  $T \leftarrow \mathrm{HMAC0}_K(M)$
  return $T$

HMAC2 Algorithm : $\mathrm{HMAC2}_K\ (M)$
  if $|K| > b$ then
      $K \leftarrow H(K)$
  end if
  $K \leftarrow K \mid\mid 0^{b-|K|}$
  $T \leftarrow \mathrm{HMAC0}_K(M)$
  return $T$
```



The above picture represent HMAC0 algorithm. HMAC is MAC based on a cryptographic hash function rather than a blockcipher (although most cryptographic hash functions are themselves based on blockciphers). The mechanism enjoys strong provable-security bounds, albeit not from preferred assumptions. Multiple closely-related variants in the literature complicate gaining an understanding of what is known. No damaging attacks have ever been suggested. Widely standardized and used. HMAC is used not only as a MAC but also as a PRF, assuming underlying compression function H is a PRF.

Algorithm 13 HMAC Algorithms (Alternative Constructions)

```
**NMAC Algorithm** : $\mathrm{NMAC}_{L1\ ||\ L2}\ (M)$
$X \leftarrow h^*_{L1}\ (M\ ||\ \mathrm{pad}(b + |M|))$
$Y \leftarrow h_{L2}\ (X\ ||\ \mathrm{pad}(b + c))$
return $Y$

**KDF0 Algorithm** : $\mathrm{KDF0}\ (K)$
$L1 \leftarrow h_{IV}(K \oplus \mathrm{ipad})$
$L2 \leftarrow h_{IV}(K \oplus \mathrm{opad})$
return $L1\ ||\ L2$

**HMAC0 Algorithm** : $\mathrm{HMAC0}_K\ (M)$
$L1\ ||\ L2 \leftarrow \mathrm{KDF0}\ (\mathrm{K})$
$Y \leftarrow \mathrm{NMAC}_{L1\ ||\ L2}\ (M)$
$T \leftarrow \mathrm{MSB}_\tau(Y)$
return $T$

**KDF1 Algorithm** : $\mathrm{KDF1}\ (K)$
$K \leftarrow K\ ||\ 0^{b-c}$
$L1\ ||\ L2 \leftarrow \mathrm{KDF0}\ (\mathrm{K})$
return $L1\ ||\ L2$

**HMAC1 Algorithm** : $\mathrm{HMAC1}_K\ (M)$
$L1\ ||\ L2 \leftarrow \mathrm{KDF1}\ (\mathrm{K})$
$Y \leftarrow \mathrm{NMAC}_{L1\ ||\ L2}\ (M)$
$T \leftarrow \mathrm{MSB}_\tau(Y)$
return $T$

**KDF2 Algorithm** : $\mathrm{KDF2}\ (K)$
if $|K| > b$ then
    $K \leftarrow H(K)$
end if
$K \leftarrow K\ ||\ 0^{b-|K|}$
$L1\ ||\ L2 \leftarrow \mathrm{KDF0}\ (\mathrm{K})$
return $L1\ ||\ L2$

**HMAC2 Algorithm** : $\mathrm{HMAC2}_K\ (M)$
$L1\ ||\ L2 \leftarrow \mathrm{KDF2}\ (\mathrm{K})$
$Y \leftarrow \mathrm{NMAC}_{L1\ ||\ L2}\ (M)$
$T \leftarrow \mathrm{MSB}_\tau(Y)$
return $T$
```

5.2 Security Definitions for Authentication Modes

1. **Padding Types**

Type	**Padding Technique**
Pad_1	0^* -Appending Padding Technique
Pad_2	10^* -Appending Padding Technique $M \rightarrow M\|\|0^i$ where i is the least non-negative no $\ni\ \|M\|+i$ is a positive multiple of n
Pad_3	Length-Appending Padding Technique $M \rightarrow L\|\|M\|\|0^i$ where i is the least non negative no $\ni\ \|M\|+i$ is a positive multiple of n and L is Binary Encoding of $\|M\|$ into a field of n bits
Pad_4	$M \rightarrow M\|\|10^{(-\|M\|-2) \mod m}1$
Pad_5	$M \rightarrow M\|\|10^{(-\|M\|-t-2) \mod m}1\|\| < M >_t$ $< M >_t$ is the t-bit representation of M
Pad_6	$M \rightarrow M\|\|10^{-\|M\|-65 \mod l}\|\| < \lceil \frac{\|M\|+65}{l} \rceil >_{64}$
Pad_7	$M \rightarrow M\|\|10^{383+(-M \mod m)}\|\| < M >_{128}$
Pad_8	$M \rightarrow Pad_2(M)\|\|0^{m+(-\|Pad_2(M)\| \mod m)}$
Pad_9	$(M,d) \rightarrow Pad_2(M)\|\| < d >_n$

Table 2: Different Padding Techniques

2. **Advantage of an Everywhere Second Pre-Image Finding Adversary**

   Let $p, n \in \mathbf{N}$ with $p \geq n$ and let $F : Z_2^p \rightarrow Z_2^n$ be a compressing function using primitive $P \in Prims$. Let $\lambda \leq p$. The advantage of an everywhere second preimage finding adversary $\mathcal{A}$ is defined as:

$$Adv_F^{esec[\lambda]}(\mathcal{A}) = max_{z' \in \mathbf{Z}_2^\lambda} Pr\Big(\mathcal{P} \xleftarrow{\$} Prims, z \longleftarrow \mathcal{A}^{\mathcal{P}}(z')\big|F(z) = y\Big)$$

3. **Advantage of an Everywhere Pre-Image Finding Adversary**

   Let $p, n \in \mathbf{N}$ with $p \geq n$ and let $F : Z_2^p \rightarrow Z_2^n$ be a compressing function using primitive $P \in Prims$. Let $\lambda \leq p$. The advantage of an everywhere preimage finding adversary $\mathcal{A}$ is defined as

$$Adv_F^{epre}(\mathcal{A}) = max_{y \in \mathbf{Z}_2^n} Pr\Big(\mathcal{P} \xleftarrow{\$} Prims, z \longleftarrow \mathcal{A}^{\mathcal{P}}(y)\big|z \neq z' \wedge F(z) = F(z')\Big)$$

4. **VOLPRF security**

Let $F : \mathcal{K} \times \mathcal{X} \times \mathbb{N} \rightarrow \mathcal{Y}^+$ be a keyed function whose output length is determined by $\mathcal{X}$. That is, for arbitrary length input (X, d), it outputs from $\mathcal{Y}^d$. Consider the Sem-CPA-ROR-advantage game for encryption modes. We rephrase the Real and Ideal games as follows, and determine the VOLPRF Advantage from the CPA security definition.

```
$\mathbf{Real}_{\mathcal{E}_K^{X_i,d_i}}(m_i)$

$K \xleftarrow{\$} \mathcal{K}$

oracle $Enc(X_i, d_i, K, m_i)$
return $c_i \leftarrow \mathcal{E}_K.Enc(X_i, d_i, m_i)$
```

```
$\mathbf{PRP}_{\tilde{\mathcal{E}}_K^{X_i,d_i}}(m_i)$

$F \xleftarrow{\$} \mathcal{PRP}$

oracle $Enc(X_i, d_i, K, m_i)$
if $\exists\ m_i\ \ni\ F(X_i, d_i, K, m_i) = c_i$
    return $m_i$
else return $m_i \xleftarrow{\$} \mathcal{Y}^{d_i}$
```

The VOLPRF-Advantage of $\mathcal{A}$ is defined as following.

$$Adv_{\mathcal{E}}^{VOLPRF}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{E}}^{prp-cpa-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{E}}^{prp-cpa-0}(\mathcal{A}) = 1]|$$

.

5. **UFCMA-advantage**

Let $MAC : \mathcal{K} \times \{0,1\}^* \longrightarrow \{0,1\}^*$ be a message authentication code and let $\mathcal{A}$ be an adversary. We consider the following experiment.

$\mathbf{Exp}_{MAC}^{uf-cma}(\mathcal{A})$:

$K \xleftarrow{\$} \mathcal{K}$
Run $\mathcal{A}^{MAC_K(.)VF_K(.,.)}$ where $VF_K(M,T) = \begin{cases} 1 \text{ if } MAC_K(M) = T \\ 0 \text{ otherwise} \end{cases}$

If $\mathcal{A}$ made a verification query (M, T) s.t.
• Oracle VF returned 1 and
• $\mathcal{A}$ did not make tag generation query MAC for M prior to querying (M, T) to VF
Then return 1, else return 0. The uf-cma-advantage of $\mathcal{A}$ is defined as:

$$Adv_{MAC}^{uf-cma}(\mathcal{A}) = Pr[Exp_{MAC}^{uf-cma}(\mathcal{A}) \implies 1]$$

.

5.3 Comparison Table for Authentication Modes

In this section, we first produce a comparison table between the six MAC Modes we described in the Section 5.1. This table compares the MAC nodes in term of security type and message padding types. The parameters we mainly focus for the comparisons are PRP Advantage, Padding types, Key Size, Security Types, number of calls of Underlying Primitives and whether the modes are online and/or Parallelizable. These modes can be well categorised through this table. This is to note that with padding type 1, the CBC-MAC modes are not secure against variable input length attacks. Also, CBC-MAC 1 mode require a key of smaller size compared to all the other modes [9].

Moreover, in the next page, we also look into twenty three more MAC modes, which we have collected through various publications and competitions worldwide and compared them with the same set of parameters. The indices of the parameters are given in the below of this page for better understanding.

Mode	Security bound	Padding	Key Size	Security type	No of calls	Proper-ties
CBC-MAC 1	$\frac{\sigma q}{2^n}$ [UFCMA]	1	k	BB,FIL	$\lceil \mu/n \rceil$	O
		2	k	BB,VIL	$\lceil (\mu+1)/n \rceil$	
		3	k	BB,VIL	$\lceil \mu/n \rceil + 1$	
CBC-MAC 2	$\frac{q^2}{2^n}$ [UFCMA]	1	2k	BB,FIL	$\lceil \mu/n \rceil + 1$	O
		2(EMAC)	2k	BB,VIL	$\lceil (\mu+1)/n \rceil + 1$	
		3	2k	BB,VIL	$\lceil \mu/n \rceil + 2$	
CBC-MAC 3	$\frac{q^2}{2^n}$ [UFCMA]	1	2k	BB,FIL	$\lceil \mu/n \rceil + 2$	O
		2	2k	BB,VIL	$\lceil (\mu+1)/n \rceil + 2$	
		3	2k	BB,VIL	$\lceil \mu/n \rceil + 3$	
CBC-MAC 4		1	2k	BB,FIL	$\lceil \mu/n \rceil + 2$	O
		2	2k	BB,VIL	$\lceil (\mu+1)/n \rceil + 2$	
		3	2k	BB,VIL	$\lceil \mu/n \rceil + 3$	
CMAC	$\frac{4\sigma^2}{2^n}$ [UFCMA]	2	$2k$	BBB,VIL	$\lceil (\mu+1)/n \rceil$	O
HMAC	$\frac{q(q-1)}{2^{c+1}}$ [UFCMA]	NA	$3k$	Secure PRF	NA	

Table 3: Comparison Table 1 for MAC modes.

K, K_i	Keys Used in the Schemes	k	key Size
n	Size of Underlying Block Cipher	μ	Message Length
q	Total No of Message Queries		(Message can be broken into $\lceil \frac{\mu}{n} \rceil$
BB	Birthday Bound Security		blocks according to CBC-MAC1)
BBB	Beyond Birthday Bound Security	O	Online
FIL	Secured for Fixed Input Length	P	Parallelizable
VIL	Secured for Variable Input Length	σ	No of Blocks Queried
l	Maximum Length of Adversarial Query	σ'	Total Output Block Size
TBC	Tweakable Block Cipher		

Mode	Security bound	Padding	Key size	Security	No of calls	Properties
PMAC	$\frac{5lq^2}{2^n-2l}$[UFCMA]	any	k	BBB,VIL	$\lceil \mu/n \rceil$	P
LightMAC	$\left(1+\frac{2}{2^{n/2}-1}+\frac{1}{(2^{n/2}-1)^2}\right)\left(\frac{q_{IC}^2}{2^n}+\frac{q_v}{2^t}\right)$	2	$\lvert K_1 \rvert + \lvert K_2 \rvert$	Secure MAC	$\lceil \frac{\mu}{n} \rceil$	O,P
KECCAK	$\frac{q^2}{2^n}$	4	1600b	Collision resistant	$\lceil \frac{\mu}{1088} \rceil$ or $\lceil \frac{\mu}{512} \rceil$	O
BLAKE	$\frac{q^2}{2^n}$	5	384b or 768b(IV+s)	Collision Resistant	$\lceil \frac{\mu}{512} \rceil$ or $\lceil \frac{\mu}{1024} \rceil$	O
Grϕstl	$(\frac{\lambda+65}{m}+2)\frac{q(q-1)}{2^l}+\frac{2q}{2^n}$	6	512b or 1024b	$esec[\lambda]$	$\lceil \frac{\mu}{512} \rceil$ or $\lceil \frac{\mu}{1024} \rceil$	O
JH	$\frac{4q^2}{2^{l-m}}+\frac{2q}{2^n}$	7	1024b	$epre$	$\lceil \frac{\mu}{512} \rceil$	O
Skein	$\frac{2q}{2^l}+\frac{2q}{2^n}$	8	512+ (128.$\lceil \mu/512 \rceil$)b(IV+tweak)	$esec[\lambda]$	$\lceil \frac{\mu}{512} \rceil$	O
nEHtM	$(\frac{12\mu^2 q_m^2}{2^{2n}}+\frac{(4q+m+2q_v)\mu}{2^n})$ $\times\frac{72q_m^3}{2^{2n}}+\frac{q_m+4q_v}{2^n}$	NA	same as blockcipher	MAC-unforgeability	2	P
Two-Track-MAC		7	160b	BB, secure MAC		
WHIRLPOOL	$\mathcal{O}(\frac{\sigma^2}{2^n})$[collision resistant]	7	512b	secure hash	$\lceil \frac{\mu}{512} \rceil$	O
PMAC$_{Plus}$	$\frac{27l^3q^3}{2^{2n}}+\frac{3lq}{2^n}$	2	80/128/192/256b	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O,P
PMACX	$\frac{2^{2d}q^2}{2(2^n-q)^2}+\frac{2^d q^3}{3.2^{2n}(2^n-q)}+\frac{2^d q^2}{2^n(2^n-q)}$	2	same as TBC	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O,P
PMAC with Parity	$\frac{q^2l^2}{2^{2n}}+\frac{q^2}{2^n}$	2	same as TBC	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O,P
Sum of CBCs	$\frac{12q^3l^4}{2^{2n}}$	any	same as blockcipher	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O
OMAC	$\frac{5q^2l}{2^n}+\frac{8q^2l^4}{2^{2n}}$	2	k	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O
Cascade						
TMAC	$\frac{(3m^2+1)q^2}{2^n}$	2	$\lvert K_1 \rvert + \lvert K_2 \rvert$	BBB, VIL, secure PRF	$\lceil \frac{\mu}{n} \rceil$	O
3kf9	$\frac{q}{2^{n-2}}(l+1)+\frac{2q^3l^3+q^3l^2+2q^3l+2q^3}{2^{2n-1}}$	2	$\lvert K_1 \rvert + \lvert K_2 \rvert + \lvert K_3 \rvert$	BBB, secure PRF	$\lceil \frac{\mu}{n} \rceil + 2$	O
PEPS	$\frac{q-1}{2^{n/2}}+\frac{1}{2^l}+$ $(q(q-1)+1).Adv_G^{PRF}$	any	same as G	secure PRF	$\lceil \frac{\mu}{n} \rceil$	O
ZMAC	$\frac{2.5\sigma^2}{2^{n+min(n,t)}}+4(\frac{q}{2^n})^{3/2}$	9	same as TBC	secure PRF	$\lceil \frac{\mu}{2n} \rceil$	O,P
ZMAC+	$\frac{(\sigma')^2}{2^n}(\frac{4m+2}{2^{n+min(n,t)}}+\frac{4}{2^{min(n,t)}})$ $+\frac{2\sigma'}{2^n}+\frac{2(q-1)\sigma+q^2+4(q-1)\sigma'}{2^{min(n,t)}}$	9	same as TBC	VOLPRF security	$\lceil \frac{\mu}{2n} \rceil$	O,P
DoveMAC	$\frac{4\sigma}{2^n}+\frac{q^2m^2}{2^{2n}}+\frac{2q^2+4qm^2+4q^2m^2}{2^{n+min(n,t)}}$	2	same as TBC	secure PRF	$\lceil \frac{\mu}{t+n} \rceil$	O
EWCDM	$\frac{q^{3/2}+q_v}{2^n}$[nonce respecting], $\frac{(q+q_v)^2}{2^n}$		$\lvert K_h \rvert + \lvert K \rvert + \lvert K' \rvert$	secure MAC		
DWCDM+	$\frac{2q_m}{2^n}+\frac{3q_m}{2^{2n/3}}+\frac{2nq_m}{2^{2n/3}}(1+q_v\epsilon)$		$\lvert K \rvert$	secure MAC		

Table 4: Comparison Table 2 for MAC Modes

5.4 Suitable Data Specifications

In this section, we shall discuss performance of Authentication modes based on specifications of data to be encrypted. That is, we shall conclude which type of data best fits for the modes.

CBC-MAC $1-4$	all are secure against variable input length adversary with padding type $\neq 1$
LightMAC	Easy to implement. Small state size. Good security bound.
KECCAK	SHA3 winner. Can perform hashing, generate pseudorandom number, used in AE modes.
TMAC	small key size. Good security bound.
$3kf9$	Beyond birthday bound secure. Online.
$ZMAC_+$	Secure against variable input length adversary. Online.
DoveMAC	Can be used in AE modes. Beyond birthday bound secure.

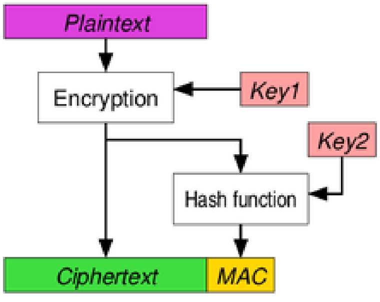
6 Authenticated Encryption Modes

Authenticated encryption (AE) is form of encryption which simultaneously assure the confidentiality and authenticity of data. Authenticated encryption with associated data (AEAD) is a variant of AE that allows a recipient to check the integrity of both the encrypted and un-encrypted information in a message. AEAD binds associated data (AD) to the ciphertext and to the context where it is supposed to appear so that attempts to cut and paste a valid ciphertext into a different context are detected and rejected. Now, with the Encryption modes and Authentication modes discussed so far, we can think of combining any one from the list of Encryption mode and any one from authentication modes to achieve confidentiality and integrity together. There are 3 types of generic compositions we can think of.

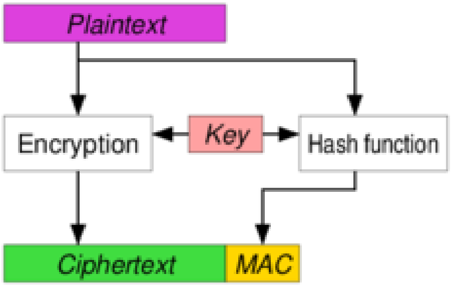
1. **Encrypt-then-MAC (EtM)** : The plaintext is first encrypted, then a MAC is produced based on the resulting ciphertext. The ciphertext and its MAC are sent together. This is the only method which can reach the highest definition of security in AE, but this can only be achieved when the MAC used is strongly unforgeable. Eg. Used in IPsec.

2. **Encrypt-and-MAC (E&M)** : A MAC is produced based on the plaintext, and the plaintext is encrypted without the MAC. The plaintext's MAC and the ciphertext are sent together. Eg. Used in SSH.

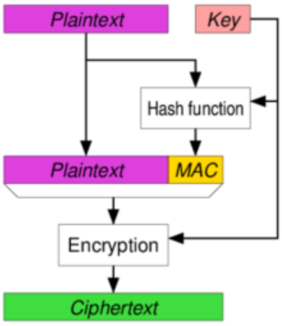
3. **MAC-then-Encrypt (MtE)** : A MAC is produced based on the plaintext, then the plaintext and MAC are together encrypted to produce a ciphertext based on both. The ciphertext (containing an encrypted MAC) is sent. Eg, Used in SSL/TLS.



Encrypt-then-MAC



Encrypt-and-MAC



MAC-then-Encrypt

Limitations: This can be clearly understood that *Encrypt-and-MAC* is not secure at all, since adversary can forge the ciphertext part(colored green) to attack the system. For *Encrypt-then-MAC* mechanism, we need two different keys which makes the system inefficient. Also, *MAC-then-Encrypt* returns error while padding of data is done. Due to these limitations of the three mechanisms, people started looking for Authenticated Encryption modes.

A nonce based authenticated encryption with associated data consists of the encryption algorithm $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C}$, tag generation algorithm $Tag : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{T}$, and the authenticated decryption algorithm $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \bigcup \{\perp\}$ where $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}, \mathcal{T}$ are respectively the key-space, nonce-space, associated data space, plaintext space, ciphertext space and tag space.

6.1 Some Examples of AEAD modes

We currently have a good number of AEAD Modes available, which are based on either Block Cipher, Stream Cipher or Sponge Constructions. In this section, we briefly discuss about four AEAD Modes - CCM, GCM, GIFT-COFB and Ascon Modes. CCM and GCM Modes were the early exposure into this domain. GIFT-COFB and Ascon are the finalists in the NIST LWC and CAESAR competitions, respectively.

1. **CCM Mode**

   As the name suggests, CCM mode combines the well known CBC-MAC with the well known counter mode of encryption. The amalgamation is in the style of MAC-then-Encrypt, but important changes make it differ from generic composition *i.e.* CBC-MAC is first computed on the message to obtain a tag; the message and the tag are then encrypted using counter mode. CCM was submitted to NIST. The initial motivation was to replace the OCB mechanism that was in the draft IEEE 802.11i standard [87] by a patent-unencumbered scheme. While CCM was designed with that particular purpose in mind, it has, by virtue of its widespread standardization, become a general-purpose scheme. Overall, CCM has many things in its favor. It enjoys a good provable-security guarantee. It has adequate performance for most applications. It is simple to implement—simpler than GCM. Hence, CCM is widely used currently.

Algorithm 14 CCM Encryption : $CCM_K^{N,A}(P)$

ECB$_K$() : ECB is an Encryption Mode, described in Section 4.1
CBCMAC$_K$() : Raw CBC-MAC is an Authentication mode, given in Section 5.1

```
$m \leftarrow \lceil |P|/128 \rceil$
$N_0 N_1 \ldots N_m \leftarrow \mathrm{Count}(N, m)$
$Y_0 Y_1 \ldots Y_m \leftarrow \mathrm{ECB}_K\ (N_0 N_1 \ldots N_m)$
$C \leftarrow P \oplus Y_0 Y_1 \ldots Y_m$
$B_0 B_1 \ldots B_r \leftarrow \mathrm{Format}(N, A, P)$
$Tag \leftarrow \mathrm{CBCMAC}_K\ (B_0 B_1 \ldots B_r)$
$T \leftarrow \mathrm{MSB}_\tau\ (Tag) \oplus Y_0$
return $C \mid\mid T$
```

The CCM algorithm has been described formally in the above. As described, the CCM mode is parameterized by a blockcipher $E : K \times \{0,1\}^n \rightarrow \{0,1\}^n$ with an $n = 128$ bit blocksize, a tag length $\tau \in [32..128]$, a formatting function *Format*, and a counter-generation function *Count*. The functions Format and Count determine the message space, nonce space, associated-data space, and restrictions on the tag length. The main insight is that the same encryption key can be used for both, provided that the counter values used in the encryption do not collide with the (pre-)initialization vector used in the authentication. A proof of security exists for this combination, based on the security of the underlying block cipher. The proof also applies to a generalization of CCM for any size block cipher, and for any size cryptographically strong pseudo-random function (since in both the counter mode and the CBC-MAC Mode, the underlying block cipher is only ever used in one direction). CCM requires two block cipher encryption operations on each block of an encrypted-and-authenticated message, and one encryption on each block of associated authenticated data. In the MAC construction, the length of the associated data has a variable-length encoding, which can be shorter than machine word size. This can cause pessimistic MAC performance if associated data is long (which is uncommon). Associated data is processed after message data, so it is not possible to pre-calculate state for static associated data. CCM is not an Online AEAD, in that the length of the message (and associated data) must be known in advance.

2. **GCM Mode**

Galois/Counter Mode (GCM) achieves Authenticated Encryption with Associated Data by combining CTR-mode encryption and Carter-Wegman message authentication. The amalgamation is in the style of encrypt-then-MAC, unlike CCM Mode which is MAC-then-Encrypt. The universal hashing underlying the Carter-Wegman authentication is based on polynomial evaluation over $GF(2^{128})$, a classical construction rooted in folklore. GCM was standardized by NIST. Galois Message Authentication Code (GMAC) is an authentication-only variant of the GCM which can form an incremental message authentication code. Both GCM and GMAC can accept initialization vectors of arbitrary length. This GHASH function is defined in the algorithm for GCM as well in the following.

Algorithm 15 GCM Encryption Algorithm

GHASH Algorithm : $\mathrm{GHASH}_H(X)$

```
$X_1 X_2 \ldots X_m \leftarrow X$ where $|X_i| = 128$
$Y \leftarrow 0^{128}$
for $i \leftarrow 1$ to $m$ do
    $Y \leftarrow (Y \oplus X_i) \bullet H$
end for
return $Y$
```

GCM Encryption : $\mathrm{GCM}_K^{N,A}(P)$

```
// $|P| \leq 2^{39} - 256,\ |A| < 2^{64},\ 0 < |N| < 2^{64}$
// 8 divides all three $|P|,\ |A|,\ |N|$
$H \leftarrow E_K\ (0^{128})$
if $|N| = 96$ then
    $Cnt \leftarrow N \mid\mid 0^{31}1$
else
    $Cnt \leftarrow \mathrm{GHASH}_K\ (N \mid\mid 0^i \mid\mid N|_{128}$ for minimal $i \geq 0\ \ni\ 128 \mid (|N| + i)$
end if
$m \leftarrow \lceil |P|/128 \rceil$
for $i \leftarrow 0$ to $m$ do
    $Y_i \leftarrow E_K\ (Cnt + i)$
end for
$C \leftarrow P \oplus (Y_2 \mid\mid Y_3 \mid\mid Y_4 \mid\mid \ldots)$
$X \leftarrow A \mid\mid 0^i \mid\mid C \mid\mid 0^j \mid\mid A|_{64} \mid\mid C|_{64}$ for minimal $i \geq 0\ \ni\ 128 \mid (|N| + i)\ \&\ 128 \mid (|C| + j)$
$Tag \leftarrow Y_1 \oplus \mathrm{GHASH}_H\ (X)$
$T \leftarrow \mathrm{MSB}_\tau\ (Tag)$
return $C \mid\mid T$
```

Like in normal counter mode, blocks are numbered sequentially, and then this block number is combined with an initialization vector (IV) and encrypted with a block cipher E, usually AES. The result of this encryption is then XORed with the plaintext to produce the ciphertext. Like all counter modes, this is essentially a stream cipher, and so it is essential that a different IV is used for each stream that is encrypted. The ciphertext blocks are considered coefficients of a polynomial which is then evaluated at a key-dependent point H, using finite field arithmetic. The result is then encrypted, producing an authentication tag that can be used to verify the integrity of the data. The encrypted text then contains the IV, ciphertext, and authentication tag. The authentication tag is constructed by feeding blocks of data into the GHASH function and encrypting the result.

GCM mode is parameterized by a 128-bit blockcipher $E : \{0,1\}^k \times \{0,1\}^{128} \rightarrow \{0,1\}^{128}$ and a tag length $\tau \in \{32, 64, 96, 104, 112, 120, 128\}$. As designed, the underlying blockcipher must be AES, with any of its three key lengths. The requirement follows from assertions that the blockcipher must be a NIST-approved and have a 128-bit block size; only AES satisfies these requirements.

Undoubtedly, GCM had quietly become the most popular AE(AD) mode in the field, despite the fact that not every cryptographer likes it. The popularity is due in part to the fact that GCM is extremely fast, but mostly it's because the mode is patent-free. GCM is an Online mode and can be parallelized, and the recent versions of OpenSSL and Crypto++ provide good implementations, mostly because it's now supported as a TLS ciphersuite.

Given all these great features, one may wonder about the reason for which many cryptographer do not want to use this. In truth, the implementation is not as simple as others. Since, GCM is Counter mode encryption with the addition of a Carter-Wegman MAC set in a Galois field. Implementing GCM is a hassle in a way that most other AEADs are not.

3. GIFT-COFB Mode

COFB (COmbined FeedBack) is a block cipher based authenticated encryption mode that uses GIFT128 as the underlying block cipher and GIFT-COFB can be viewed as an efficient integration of the COFB and GIFT-128. GIFT-128 maintains an 128-bit state and 128-bit key. To be precise, GIFT is a family of block ciphers parametrized by the state size and the key size and all the members of this family are lightweight and can be efficiently deployed on lightweight applications. COFB mode on the other hand, computes of combined feedback of block cipher output and data block to uplift the security level. This actually helps us to design a mode with low state size and eventually to have a low state implementation. This technique actually resist the attacker to control the input block and next block cipher input simultaneously. Overall, a combination of GIFT and COFB can be considered to be one of the most efficient lightweight, low state block cipher based AEAD construction. This protocol was eventually became one of the finalists of NIST Light Weight Competition for designing AEAD modes.

Algorithm 16 GIFT-COFB Encryption : COFB$-\mathcal{E}_K^{N,A}(M)$

```
$Y[0] \leftarrow E_K(N),\ L \leftarrow \mathrm{Trunc}_{n/2}(Y[0])$
$(A[1], \ldots, A[a]) \xleftarrow{n} \mathrm{Pad}(A)$
**if** $M \neq \epsilon$ **then**
    $(M[1], \ldots, M[m]) \xleftarrow{n} \mathrm{Pad}(M)$
**end if**
**for** $i \leftarrow 1$ **to** $a - 1$ **do**
    $L \leftarrow 2.L$
    $X[i] \leftarrow A[i] \oplus G.Y[i-1] \oplus L||0^{n/2}$
    $Y[i] \leftarrow E_K(X[i])$
**end for**
**if** $|A| \mod n = 0$ **and** $A \neq \epsilon$ **then**
    $L \leftarrow 3.L$
**else**
    $L \leftarrow 3^2.L$
**end if**
**if** $M = \epsilon$ **then**
    $L \leftarrow 3^2.L$
**end if**
$X[a] \leftarrow A[a] \oplus G.Y[a-1] \oplus L||0^{n/2}$
$Y[a] \leftarrow E_K(X[a])$
**for** $i \leftarrow 1$ **to** $m - 1$ **do**
    $L \leftarrow 2.L$
    $C[i] \leftarrow M[i] \oplus Y[i+a-1]$
    $X[i+a] \leftarrow M[i] \oplus G.Y[i+a-1] \oplus L||0^{n/2}$
    $Y[i+a] \leftarrow E_K(X[i+a])$
**end for**
**if** $M \neq \epsilon$ **then**
    **if** $|M| \mod n = 0$ **then**
        $L \leftarrow 3.L$
    **else**
        $L \leftarrow 3^2.L$
    **end if**
    $C[m] \leftarrow M[m] \oplus Y[m+a-1]$
    $X[m+a] \leftarrow M[m] \oplus G.Y[m+a-1] \oplus L||0^{n/2}$
    $Y[m+a] \leftarrow E_K(X[m+a])$
    $C \leftarrow \mathrm{Trunc}_{|M|}\ (C[1]\ ||\ C[2]\ ||\ \ldots\ ||\ C[m])$
    $T \leftarrow \mathrm{Trunc}_{\tau}(Y[a+m])$
**else**
    $C \leftarrow \epsilon$
    $T \leftarrow \mathrm{Trunc}_{\tau}(Y[a])$
**end if**
**return** $C\ ||\ T$
```

GIFT-128 is an 128-bit Substitution-Permutation network (SPN) based block cipher with a key length of 128-bit. It is a 40-round iterative block cipher with identical round function. There are two versions of GIFT, namely GIFT-64 and GIFT-128. The complete construction of the GIFT-128 block cipher is given in [20]. GIFT is considered to be one of the lightest design existing in the literature.

COFB is a lightweight AEAD mode. The underlying cryptographic primitive is an n-bit block cipher, E_K. We assume that n is a multiple of 4. The key of the scheme is the key of the block cipher, i.e. K. We provide the detailed encryption algorithm of the COFB Authenticated Encryption Mode in the next page. The mode presented in the construction of GIFT-COFB slightly differs with the original proposal though. The designers of GIFT-COFB changed the nonce to be 128 bit, the feedback to make it more hardware efficient, the mask update function to deal with empty data and the padding for the associated data. These updates make the design more lightweight and more efficient to deal with short data inputs. However, this updates does not have impact on the security of the mode, except a nominal 1-bit security degradation.

4. **Ascon Mode**

The cipher suite Ascon, which provides Authenticated Encryption with Associated Data (AEAD) and Hashing Functionality consists of the authenticated ciphers Ascon-128 and Ascon-128a, which have been selected as primary choice for lightweight authenticated encryption in the final portfolio of the CAESAR competition. The recommendation for NIST also includes Ascon-Hash combined with Ascon-128 or Ascon-128a. All schemes provide 128-bit security and internally use the same 320-bit permutation (with different round numbers) so that a single lightweight primitive is sufficient to implement both AEAD and hashing.

The algorithm given in the next page is used for Authenticated Encryption of Ascon. The mode of operation for hashing is based on sponges. Both the hash function Ascon-Hash with fixed output size and the eXtendable output function Ascon-Xof with variable output size internally use the same hashing algorithm. The detailed algorithms for hashing and verified decryption can be found in [21].

All Ascon family members provide 128-bit security in the notion of nonce-based AEAD *i.e.* they protect the confidentiality of the plaintext (except its length) and the integrity of ciphertext including the associated data (under adaptive forgery attempts). The number of processed plaintext and associated data blocks protected by the encryption algorithm is limited to a total of 2^{64} blocks per key, which corresponds to maximum of 2^{68} bytes. We consider this as more than sufficient for lightweight applications in practice. Ascon provides 128-bit security against collision attacks and (second) pre-image attacks. Like other sponge based hash functions, Ascon-Hash also resists other attacks, including length extension attacks and second-preimage attacks for long messages.

Algorithm 17 Ascon Authenticated Encryption : $\textsc{Ascon}{-}\mathcal{E}_K^{N,A}(P)$

```
// $P \in \{0,1\}^*,\ A \in \{0,1\}^*,\ N \in \{0,1\}^{128},\ |K| \leq 160$
$S \leftarrow \mathrm{IV} \mid\mid K \mid\mid N$
$S \leftarrow p^a(S) \oplus (0^{320-k} \mid\mid K)$
if $|A| > 0$ then
    $A_1A_2 \ldots A_s \leftarrow r -$ bit blocks of $A \mid\mid 1 \mid\mid 0*$
    for $i \leftarrow 1$ to $s$ do
        $S \leftarrow p^b\ ((S_r \oplus A_i) \mid\mid S_c)$
    end for
end if
$S \leftarrow S \oplus (0^{319} \mid\mid 1)$
$P_1P_2 \ldots P_t \leftarrow r -$ bit blocks of $P \mid\mid 1 \mid\mid 0*$
for $i \leftarrow 1$ to $t - 1$ do
    $S_r \leftarrow S_r \oplus P_i$
    $C_i \leftarrow S_r$
    $S \leftarrow p^b\ (S)$
end for
$S_r \leftarrow S_r \oplus P_t$
$\tilde{C_t} \leftarrow \lfloor S_r \rfloor_{|P| \bmod r}$
$C \leftarrow C_1 \mid\mid C_2 \mid\mid \ \ldots \ \mid\mid C_{t-1} \mid\mid \tilde{C_t}$
$S \leftarrow p^a\ (S \oplus (0^r \mid\mid K \mid\mid 0^{320-r-k}))$
$T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$
return $C \mid\mid T$
```

6.2 Security Definitions for AEAD modes

There is a huge variety of construction of the AEAD modes. This is why, all the existing AEAD modes can not be concluded secure or insecure under certain definitions. This provoked different security definitions for AEAD modes. Most of the security definitions are defined using real-or-ideal games. These games are used by the challenger as an intermediate step of the game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. $\mathcal{F}$ is considered to be the set of all random functions, *Params* is the set of all necessary parameters like associated data, nonce, tweak etc. The adversary outputs his guess bit using any one of the two experiment algorithms Exp^0 and Exp^1.

1. **IND-CPA security**

   The adversary is only allowed to query for plaintexts. Since we are now considering AEAD modes, so the adversary can provide its choice for nonce and associated data too. The game is as follows.

Decide AEAD mode $\mathcal{E}$

```
Adversary $\mathcal{A}$                                   Challenger $\mathcal{C}$

                                                          choose $K \xleftarrow{\$} \mathcal{K}$
                                                          choose $F \xleftarrow{\$} \mathcal{F}$
For $i = 1, 2, \ldots, n$
    $m_i \leftarrow \mathcal{M}$
    $params_i \leftarrow \mathcal{P}$
                      $\xrightarrow{(params_i, m_i)}$
                                                          choose $b \xleftarrow{\$} \{0,1\}$
                                                          For $i = 1, 2, \ldots, n$
                                            if $b = 0$                                   | if $b = 1$
                                            $\mathbf{Real}_{\mathcal{E}_K^{params_i}}(m_i)$       | $\mathbf{Ideal}_{\mathcal{E}_K^{params_i}}(m_i)$
                                            $c_i \leftarrow \mathcal{E}_K.Enc(params_i, m_i)$ | $c_i \leftarrow F(params_i, m_i)$
                      $\xleftarrow{c_i}$
$b' \leftarrow \mathbf{Exp}^{CPA-j}(\mathcal{A})$
output $b'$
```

$\mathcal{A}$ wins the game if $b' = b$

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CPA-1}$ and $\mathbf{Exp}^{CPA-0}$ with all $m_i,\ c_i,\ params_i$ as input. $Input = (m_1,\ c_1,\ params_1,\ m_2,\ c_2,\ params_2, \ldots,\ m_n,\ c_n,\ params_n)$.

Case 1:	**Case 2:**
$\mathbf{Exp}^{CPA-1}(\mathcal{A})$	$\mathbf{Exp}^{CPA-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$g \xleftarrow{\$} \mathcal{F}$
$b' \leftarrow \mathcal{A}^{\mathcal{E}_K}(Input)$	$b' \leftarrow \mathcal{A}^{g}(Input)$

Then the IND-CPA advantage of $\mathcal{A}$ for the AEAD mode $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{CPA}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{E}}^{CPA-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{E}}^{CPA-0}(\mathcal{A}) = 1]|$$

2. **IND-CCA security**

The adversary is allowed to query for plaintexts and ciphertexts.The adversary can provide its choice for nonce and associated data alongside the plaintexts and ciphertexts. The game is as follows.

Decide AEAD mode $\mathcal{E}$

Adversary $\mathcal{A}$ | **Challenger** $\mathcal{C}$

choose $K \xleftarrow{\$} \mathcal{K}$

choose $F \xleftarrow{\$} \mathcal{F}$

For $i = 1, 2, \ldots, n$

$m_i \leftarrow \mathcal{M}$

$params_i \leftarrow \mathcal{P}$

For $j = 1, 2, \ldots, n'$

$c_j{}' \leftarrow \mathscr{C}$

$params_j{}' \leftarrow \mathcal{P}$

$$\xrightarrow[(params_j{}', c_j{}')]{(params_i, m_i)}$$

choose $b \xleftarrow{\$} \{0, 1\}$

For $i = 1, 2, \ldots, n$, **For** $j = 1, 2, \ldots, n'$

if $b = 0$	if $b = 1$
$\mathbf{Real}_{\mathcal{E}_K^{params_i}}(m_i)$	$\mathbf{Ideal}_{\mathcal{E}_K^{params_i}}(m_i)$
$c_i \leftarrow \mathcal{E}_K.Enc(params_i, m_i)$	$c_i \leftarrow F(params_i, m_i)$
$m'_j \leftarrow \mathcal{E}_K.Dec(params'_j, c'_j)$	$m'_j \leftarrow F^{-1}(params'_j, c'_j)$

$$\xleftarrow{c_i, m'_j}$$

$b' \leftarrow \mathbf{Exp}^{CCA-j}(\mathcal{A})$

output b'

$\mathcal{A}$ wins the game if $b' = b$

Here $\mathcal{A}$ simulates its experiments $\mathbf{Exp}^{CCA-1}$ and $\mathbf{Exp}^{CCA-0}$ with all $m_i, c_i, params_i$ as input. $Input = (m_1, c_1, params_1, ..., m_n, c_n, params_n; c'_1, m'_1, params'_1, ..., c'_{n'}, m'_{n'}, params'_{n'})$:

Case 1:	**Case 2:**
$\mathbf{Exp}^{CCA-1}(\mathcal{A})$	$\mathbf{Exp}^{CCA-0}(\mathcal{A})$
$K \xleftarrow{\$} \mathcal{K}$	$g \xleftarrow{\$} \mathcal{F}$
$b' \leftarrow \mathcal{A}^{\mathcal{E}_K}(Input)$	$b' \leftarrow \mathcal{A}^{g}(Input)$

Then the IND-CCA advantage of $\mathcal{A}$ for the AEAD mode $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{CCA}(\mathcal{A}) = |Pr[\mathbf{Exp}_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

3. **INT-CTXT security**

   Let $\mathcal{E} = (Enc, Dec)$ be an AEAD mode.
   Let $G_{INT-CTXT}$ be the ciphertext integrity game defined as follows.

```
**Game** $G_{INT-CTXT}$

Procedure *Initialize(ν)*

$K \xleftarrow{\$} \mathcal{K}$

Procedure *Finalize*

**return** $win$

Procedure *Encrypt(H,M)*

**if** $\nu = NR$ and $v \in B$ **then**
    **return** $\perp$
**end if**
$C \leftarrow \mathcal{E}_K.Enc(H, M)$
$B \leftarrow B \bigcup \{V\}$
$\mathcal{Q} \leftarrow \mathcal{Q} \bigcup \{(H, C)\}$
**return** $C$

Procedure *Verify(H,C)*

$M \leftarrow \mathcal{E}_K.Dec(H, C)$
**if** $(H, C) \notin \mathcal{Q}$ and $M \neq \perp$ **then**
    $win \leftarrow$ **True**
**end if**
**return** $M(\neq\perp)$
```

   For an adversary $\mathcal{A}$, the INT-CTXT advantage over the scheme $\mathcal{E}$ is defined to be

$$Adv_{\mathcal{E}}^{INT-CTXT} = Pr[\mathcal{A}_{G_{INT-CTXT}}(\nu) \implies 1]$$

4. **nAEAD Security**

   A nonce based AEAD mode is an authenticated encryption mode with associated data where a nonce is used. That is, $\mathcal{E} = (Enc, Dec)$ is a nonce based AEAD scheme where $\mathcal{K}$ is the key space, encryption function $Enc : \mathcal{K} \times \mathcal{N} \times \mathcal{A}' \times \mathcal{M} \rightarrow \mathscr{C} \bigcup \{\perp\}$ and decryption function $Dec : \mathcal{K} \times \mathcal{N} \times \mathcal{A}' \times \mathscr{C} \rightarrow \mathcal{M} \bigcup \{\perp\}$, where $\mathcal{N}$ is nonce space, $\mathcal{C}$ is ciphertext space, $\mathcal{M}$ is message space, $\mathcal{A}'$ is space of associated data. The adversary can choose nonce N, associated data A before querying to challenger $\mathcal{C}$. The nonce must be chosen following the nonce respecting policy. That is, the nonce for two distinct messages must not be chosen same. The real and ideal games are as follows.

Real$_{\mathcal{E}}$:	**Ideal**$_{\mathcal{E}}$:
$K \longleftarrow \mathcal{K}$	for (N,A)$\longleftarrow \mathcal{N} \times \mathcal{A}'$:
	$\pi_{N,A} \longleftarrow \mathcal{F}$,
Oracle Enc(N,A,K,M)	**Oracle** Enc(N,A,K,M)
return $\mathcal{E}_K.Enc(N,A,K,M)$	return $\pi_{N,A}(K,M)$
Oracle Dec(N,A,K,C)	**Oracle** Dec(N,A,K,C)
return $\mathcal{E}_K.Dec(N,A,K,C)$	If $\exists$ M s.t. $\pi_{N,A}(K,M) = C$,
	then return M
	else return $\perp$

5. **MRAE security**

Even though nonce based AE schemes are easy to implement, but still, reusing nonce can lead to strong attacks. The misuse resistance AE security is defined using the real and ideal games defined in nAEAD security. Here the adversary can query both encryption and decryption oracles, and can reuse the underlying nonce. The scheme Π is MRAE secured if the advantage Adv_{Π}^{mrae} is negligible, where

$$Adv_{\Pi}^{mrae} = Pr[\mathscr{A}^{\mathcal{E}_K(.,.,.)\mathcal{D}_K(.,.,.)} \implies 1] - Pr[\mathscr{A}^{\$(.,.,.)\perp(.,.,.)} \implies 1]$$

6. **Online AE or OAE Security**

nAEAD secured modes cannot repeat nonce. Whereas, MRAE is a really strong security assumption. But MRAE modes cannot be online. This is why, we need an intermediate security definition. In fact, we say that a mode is OAE secure if it is online and any nonce reusing adversary cannot exploit it.

7. **RAE Security**

Let $\Pi = (\mathcal{E}_K, \mathcal{D}_K)$ be a nonce based encryption scheme. Consider the IND-CCA game for AEAD modes. Then the Robust AE security is defined using rephrasing the two games Real and Ideal as follows.

Real$_{\Pi}$:	**RAE**$_{\Pi}$:
$K \longleftarrow \mathcal{K}$	for (N,A,λ)$\longleftarrow \Sigma^* \times \Sigma^* \times \mathbb{N}$:
	$\pi_{N,A,\lambda} \longleftarrow Inj(\lambda)$,
Oracle Enc(N,A,λ,M)	**Oracle** Enc(N,A,λ,M)
return $\mathcal{E}_K(N,A,\lambda,M)$	return $\pi_{N,A,\lambda}(M)$
Oracle Dec(N,A,λ,C)	**Oracle** Dec(N,A,λ,C)
return $\mathcal{D}_K(N,A,\lambda,C)$	If $\exists$ M s.t. $\pi_{N,A,\lambda}(M) = C$,
	then return M
	else return $\perp$

Define the two experiments **Exp-0**, **Exp-1** by $RAE_{\Pi}(N,A,\lambda,M)$ and $Real_{\Pi}(N,A,\lambda,M)$. Then, the nonce based encryption scheme is RAE secure if Adv_{Π}^{RAE} is negligible, where

$$Adv_{\Pi}^{RAE} = Pr[\mathscr{A}^{CCA-0_{\Pi}(N,A,\lambda,M)} \implies 1] - Pr[\mathscr{A}^{CCA-1_{\Pi}(N,A,\lambda,M)} \implies 1]$$

8. **muCCAmL2 Security**

Let $LEnc_K(i,N,A,M)$ be the function that outputs $Enc_{K_i}(N,A,M)$ with leakage $L_{enc}(K_i,N,A,M)$, $LDec_K(i,N,A,M)$ outputs $Dec_{K_i}(N,A,M)$ with leakage $L_{dec}(K_i,N,A,M)$, $LDec^{\perp}_K(i,N,A,M)$ outputs $Leak_d \leftarrow L_{dec}(K_i,N,A,M)$ if C is an output of some leaking encryption query (i,N,A,M) and outputs $(\perp, Leak_d)$ otherwise. For a multi-user AEAD scheme, we define the *Ciphertext Integrity with Misuse-resistance and (encryption & decryption) Leakage* game as follows.

$Priv^{muCCAmL2,b}$ **Game**

- **Initialization :** Generate $K_1, K_2, \ldots, K_u \leftarrow \mathcal{K}, \mathcal{E}_{ch}, \mathcal{E}_1, \ldots, \mathcal{E}_u \leftarrow \{\phi\}$

- **Leaking Encryption Queries :** $\mathcal{A}^L$ queries $LEnc$ adaptively.
  $LEnc$ outputs $\perp$ if $(i,N,*,*) \in \mathcal{E}_{ch}$,
  outputs $(Enc_{K_i}(N,A,AM), L_{enc}(K_i,N,A,M))$ otherwise.
  In the later case, update $\mathcal{E}_i \leftarrow \mathcal{E}_i \bigcup \{N\}$.

- **Leaking Decryption Queries:** $\mathcal{A}^L$ queries $LDec$ adaptively.
  $LDec$ outputs $\perp$ if $(i,N,A,C) \in \mathcal{E}_{ch}$, and
  outputs $(Dec_{K_i}(N,A,C), L_{dec}(K_i,N,A,C))$, otherwise.

- **Challenge Queries :** $\mathcal{A}^L$ submits $(i, N_{ch}, A_{ch}, M^0, M^1)$
  If $|M^0| \neq |M^1|$, **return** $\perp$
  Else $b \xleftarrow{\$} \{0,1\}$,
  Update $\mathcal{E}_{ch} = \mathcal{E}_{ch} \bigcup \{(i, N_{ch}, A_{ch}, L_{enc}\ (K_i,N,A,M))\}$
  return $LEnc(i, N_{ch}, A_{ch}, M^b)$

- **Decryption Challenge Leakage Queries :** $\mathcal{A}^L$ queries L_{decch} adaptively.
  For a query (i, N_{ch}, A_{ch}, C^b),
  L_{decch} **return** $L_{dec}(k, N_{ch}, A_{ch}, C^b)$ if $(i, N_{ch}, A_{ch}, C^b) \in \mathcal{E}_{ch}$, and
  L_{decch} **return** $\perp$, otherwise.

- **Finalization :** $\mathcal{A}^L$ returns guess bit b'.

The muCCAmL2 Advantage of $\mathcal{A}^{\mathcal{L}}$ is defined as

$$|Pr[Priv^{muCCAmL2,0} \implies 1] - Pr[Priv^{muCCAmL2,1} \implies 1]|$$

9. **Anonymous nAE Security**

Let $\mathcal{E}$ be an anonymous nonce-based AEAD scheme, whose set of parameters are $\{l, N, A, AD\}$, which are generated by the following rules. $K \xleftarrow{\$} \mathcal{K}$, $l \leftarrow \mathcal{E}.Init(K)$, $K[l] \leftarrow K$, $L \leftarrow L \bigcup \{l\}$, $A \leftarrow \mathcal{E}.Asso(A,l)$, $N \leftarrow \mathcal{N}$. Then the anonymous nAE security of $\mathcal{E}$ is defined as the CCA game with Real and Ideal games redefined as follows.

```
anAE-Real$_\mathcal{E}$ :
Get $M$ or $C$ from $\mathcal{A}$

oracle $Enc(l, N, A, M)$
if $l \notin L$ or $N \notin NE[l]$ return $\perp$
else return $c \leftarrow \mathcal{E}.Enc(K[l], N, A, M)$

oracle $Dec(l, N, A, C)$
return $\mathcal{E}.Dec(l, N, A, C)$
```

```
anAE-Ideal$_\mathcal{E}$ :
Get $M$ or $C$ from $\mathcal{A}$

oracle $Enc(l, N, A, M)$
if $l \notin L$ or $N \notin NE[l]$ return $\perp$
else $NE[l] \leftarrow NE[l] \bigcup \{N\}$
$H[C] \leftarrow H[C] \bigcup \{(l, N, A, M)\}$
return $c \xleftarrow{\$} \{0,1\}^{|M|+t}$

oracle $Dec(l, N, A, C)$
if $H[C] = \{\phi\}$ return $\perp$
else if $\exists!(l, N, A, M) \in H[C]$ s.t.
$l \in L, N \in N_x(ND[l]), A \in AD \bigcup A[l]$
return $(l, N, A, M)$
```

Then the anAE advantage of the adversary $\mathcal{A}$ will be

$$Adv_\mathcal{E}^{anAE}(\mathcal{A}) = |Pr[Exp_\mathcal{E}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_\mathcal{E}^{CCA-0}(\mathcal{A}) = 1]|$$

10. **LAE or Leakage Resistant AE Security**

Let $\mathcal{E}$ be an AEAD mode and $\mathcal{A}$ be an adversary. Consider the IND-CCA game for AEAD modes. Then the LAE game is defined for the parameter set $params = (N, A)$ by rephrasing the *real* and *ideal* games as follows:

```
Game $LAE - Ideal_\mathcal{E}$

Procedure Enc(N,A,M)
if $f[N, A, M] = \perp$ then
    $f[N, A, M] \xleftarrow{\$} \{0,1\}^{|C|}$
end if
return $f[N, A, M]$

Procedure Dec(N, A, C)

return $\perp$
```

```
Game $LAE - Real_\mathcal{E}$

Procedure Enc(N,A,M)
return $\mathcal{E}_K(N, A, M)$

Procedure Dec(N, A, C)

return $\mathcal{E}_K^{-1}(N, A, C)$
```

Then the LAE advantage of $\mathcal{E}$ is defined as:
$Adv_\mathcal{E}^{LAE}()\mathcal{A} = |Pr[\mathbf{Exp}_\mathcal{E}^{CCA-1}(\mathcal{A}) = 1] - Pr[\mathbf{Exp}_\mathcal{E}^{CCA-0}(\mathcal{A}) = 1]|$

11. **nvAE Security**

Let $\mathcal{E}$ be a nonce based AEAD scheme with variable stretch. Let $\mathcal{X}$ be the set of all permissible pairs (N, τ) where N is nonce and τ is an internal parameter that depends upon ciphertext C. Let $\mathcal{Y}$ be the set of all (N, A, C) triplets where A is associated data, C is ciphertext. $\mathcal{F}$ is set of all pseudorandom functions depending upon K, N, A, τ, M, C. To define nvAE security of $\mathcal{E}$, we consider the CCA-security game for AEAD modes along with $params_i = (N_i, A_i, \tau_i)$ and redefine Real and Ideal games.

```
**nvAE-Real**$_F$ :
Get $M$ or $C$ from $\mathcal{A}$
$K \xleftarrow{\$} \mathcal{K}$
$\mathcal{X} \leftarrow \{\phi\}, \mathcal{Y} \leftarrow \{\phi\}$
**Oracle** $Enc(N, A, \tau, M)$
if $(N, \tau) \in \mathcal{X}$
**return** $\perp$
$\mathcal{X} \leftarrow \mathcal{X} \bigcup \{(N, \tau)\}$
$C \leftarrow \mathcal{E}.Enc(K, N, A, \tau, M)$
if $\tau = \tau_C$
$\mathcal{Y} \leftarrow \mathcal{Y} \bigcup \{(N, A, C)\}$
**return** $C$
**Oracle** $Dec(N, A, \tau, C)$
if $\tau = \tau_C$ and $(N, A, C) \in \mathcal{Y}$
**return** $\perp$
else
**return** $\mathcal{E}.Dec(K, N, A, \tau, C)$
```

```
**nvAE-Ideal**$_F$ :
Get $M$ or $C$ from $\mathcal{A}$
$K \xleftarrow{\$} \mathcal{K}$
$\mathcal{X} \leftarrow \{\phi\}, \pi \xleftarrow{\$} \mathcal{F}$
**Oracle** $Enc(N, A, \tau, M)$
if $(N, \tau) \in \mathcal{X}$
**return** $\perp$
$\mathcal{X} \leftarrow \mathcal{X} \bigcup \{(N, \tau)\}$
if $\tau = \tau_C$
$C \xleftrightarrow{\$} \{0, 1\}^{|M| + \tau_C}$
**return** $C$
**return** $\pi(K, N, A, \tau, M)$
**Oracle** $Dec(N, A, \tau, C)$
if $\tau = \tau_C$
**return** $\perp$
else
**return** $\pi^{-1}(K, N, A, \tau, C)$
```

Then the nvAE advantage of the adversary $\mathcal{A}$ for the scheme $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{nvAE}(\mathcal{A}) = |Pr[Exp_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

12. **DAE security**

We define DAE security notions for deterministic authenticated encryption schemes. Suppose $\mathcal{E} = (Enc, Dec)$ be a nonce-based deterministic AEAD scheme. That is, it always returns the same value for querying with same parameters N (nonce), A (asso. data), M (message), C (ciphertext). The DAE security game is defined as the following CCA security game with $params_i = (N_i, A_i)$ and the real and ideal games.

```
**DAE-Real**$_\mathcal{E}$ :
$K \xleftarrow{\$} \mathcal{K}$
Get $A, N, M/C$ from $\mathcal{A}$

**oracle** $Enc(N, A, K, M)$
**return** $\mathcal{E}_K.Enc(N, A, M)$

**oracle** $Dec(N, A, K, C)$
**return** $\mathcal{E}.Dec(N, A, K, C)$
```

```
**DAE-Ideal**$_\mathcal{E}$ :
$K \xleftarrow{\$} \mathcal{K}$
Get $A, N, M/C$ from $\mathcal{A}$
$resp \leftarrow \{\phi\}, resp_1 \leftarrow \{\phi\}$

**oracle** $Enc(N, A, K, M)$
if $\exists C$s.t. $((N, A, K, M), C) \in resp$
**return** $C$
else $C \xleftarrow{\$} \mathscr{C}$
$resp \leftarrow resp \bigcup \{((N, A, K, M), C)\}$
**return** $C$

**oracle** $Dec(N, A, K, C)$
if $\exists M$s.t. $((N, A, K, C), M) \in resp_1$
**return** $M$
else $M \xleftarrow{\$} \mathcal{M}$
$resp_1 \leftarrow resp_1 \bigcup \{((N, A, K, C), M)\}$
**return** $M$
```

Then the DAE advantage of the adversary $\mathcal{A}$ for the scheme $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{DAE}(\mathcal{A}) = |Pr[Exp_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

13. **Subtle AE or SAE security**

    A subtle AE scheme is a nonce based AE scheme, that, in addition, uses a header from the space $\mathcal{H}$. Thus, a subtle AE scheme $\mathcal{E} = (Enc, Dec)$ can be defined using $Enc : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \rightarrow \mathscr{C} \times \mathcal{T}$ and $Dec : \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathscr{C} \times \mathcal{T} \rightarrow \mathcal{M} \bigcup \{\perp\}$. The security of a subtle AE scheme is defined using the IND-CCA game and the real and ideal games are defined as following.

Real$_{\mathcal{E}}$:	**Ideal**$_{\mathcal{E}}$:
$K \longleftarrow \mathcal{K}$	for (N,A,H)$\longleftarrow \mathcal{N} \times \mathcal{A}' \times \mathcal{H}$:
	$\pi_{N,A,H} \longleftarrow \mathcal{F}$,
Oracle Enc(N,A,H,K,M)	**Oracle** Enc(N,A,H,K,M)
return $\mathcal{E}_K.Enc(N, A, H, K, M)$	return $\pi_{N,A,H}(K, M)$
Oracle Dec(N,A,H,C)	**Oracle** Dec(N,A,H,C)
return $\mathcal{E}_K.Dec(N, A, H, C)$	If $\exists$ M s.t. $\pi_{N,A,H}(K, M) = C$,
	then return M
	else return $\perp$

    Then the SAE advantage of the adversary $\mathcal{A}$ for the scheme $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{SAE}(\mathcal{A}) = |Pr[Exp_{\mathcal{E}}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E}}^{CCA-0}(\mathcal{A}) = 1]|$$

14. CCA_3 **security**
    An AEAD mode $\mathcal{E} = (Enc, Dec)$ is called CCA_3 secure if it is both IND-CPA secure and INT-CTXT secure.

15. **RUP Security of AE**
    For encrypting data in small-capacity devices, *i.e.*, which cannot process very large data chunks, we need to define security notion in different way. The message fragments are enumerated, which includes new parameter H in defining security. Let $\mathcal{E} = (Enc, Dec)$ be a nonce based AE scheme. The INT-RUP security is defined using the IND-CCA game where Real and Ideal games are redefined as below.

Real$_{\mathcal{E}}$:	**Ideal**$_{\mathcal{E}}$:
$K \longleftarrow \mathcal{K}$	$K \longleftarrow \mathcal{K}$
Oracle Enc(N,A,H,K,M)	**Oracle** Enc(N,A,H,K,M)
return $\mathcal{E}_K.Enc(N, A, H, K, M)$	**return** $\mathcal{E}_K.Enc(N, A, H, K, M)$
Oracle Dec(N,A,H,C)	**Oracle** Dec(N,A,H,C)
return $\mathcal{E}_K.Dec(N, A, H, C)$	**return** $\mathcal{E}_K.Dec(N, A, H, C)$
Oracle Verify(N,A,H,M,T)	**Oracle** Verify(N,A,H,M,T)
return $Verify_{N,A,H}(M, T)$	**return** $\perp$

Then the RUP advantage of the adversary $\mathcal{A}$ for the scheme $\mathcal{E}$ will be

$$Adv_{\mathcal{E}}^{INT-RUP}(\mathcal{A}) = |Pr[Exp_{\mathcal{E},Verify}^{CCA-1}(\mathcal{A}) = 1] - Pr[Exp_{\mathcal{E},\perp}^{CCA-0}(\mathcal{A}) = 1]|$$

6.3 Necessary Condition for Security of AEAD Modes

The general format of a nonce based AEAD mode is $\mathcal{E} = (KeyGen, (Enc, TagGen), Dec)$

$$\text{where } (Enc, TagGen) : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T}$$

$$\text{and } Dec : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\perp\}$$

The AEAD modes consists of an encryption and a tag generation algorithm as parts of the authenticated encryption procedure. Thus, for security of the mode, it is necessary for the encryption algorithm to possess IND-CCA, and for the tag-generation part, to have INT-CTXT security. Bellare et.al.[2] have proven that $IND-CPA+INT-CTXT \implies IND-CCA$.

Theorem. Let $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AEAD mode. Let $\mathcal{A}$ be probabilistic polynomial time IND-CCA adversary asking for q_e many encryption and q_d many decryption queries. Then it is possible to construct a probabilistic polynomial time IND-CPA adversary and a probabilistic polynomial time INT-CTXT adversary.

Proof. Consider the following games.

```
Games $G_0, G_1$

Procedure Initialize
$K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}, S \leftarrow \Phi$

Procedure LR($M_0, M_1$)
$C = \mathbb{E}(K, M_b), S \leftarrow S \bigcup \{C\}$
return $C$

Procedure Dec($C$)
If $C \notin S$ then $M \leftarrow \mathcal{D}(K, C)$
else $M \leftarrow \perp$
if $M \neq \perp$ then
    $bad \leftarrow true$
    $M \leftarrow \perp$ // for $G_1$
end if
return $M$

Procedure Finalize
return $(d = b)$
```

```
Game $G_2$

Procedure Initialize
$K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}$

Procedure LR($M_0, M_1$)
$C = \mathbb{E}(K, M_b)$
return $C$

Procedure Dec($C$)
return $\perp$

Procedure Finalize
return $(d = b)$
```

$$\text{Then we have } Pr[IND-CCA_{\mathbb{SE}}^{\mathcal{A}} \implies true] = Pr[G_0^{\mathcal{A}} \implies true]$$
$$= Pr[G_1^{\mathcal{A}} \implies true] + (Pr[G_0^{\mathcal{A}} \implies true] - Pr[G_1^{\mathcal{A}} \implies \text{true}])$$
$$= Pr[G_1^{\mathcal{A}} \implies true] + Pr[G_1^{\mathcal{A}} \text{sets bad}]$$

Now Dec procedure of G_1 and G_2 both always return $\perp$.
So, $Pr[G_1^{\mathcal{A}} \implies true] = Pr[G_2^{\mathcal{A}} \implies true]$

Now we design an adversary $\mathcal{A}_c$ which runs the game G_1.
Then $Pr[G_1^{\mathcal{A}}\text{sets bad}] \leq Pr[INT-CTXT_{\mathcal{SE}}^{\mathcal{A}_c}\text{returns true}]$

Let $\mathcal{A}_p$ be an adversary that simply runs the game G_2.
Then $Pr[G_2^{\mathcal{A}}\text{returns true}] \leq Pr[IND-CPA_{\mathcal{SE}}^{\mathcal{A}_c}\text{returns true}]$

Clearly $\mathcal{A}_c$ and $\mathcal{A}_p$ are probabilistic polynomial time adversaries. Thus the claim is true.

Conversely, we can conclude that if a scheme is IND-CPA + INT-CTXT, then it is also IND-CCA. Now, suppose $(\mathcal{E}, \mathcal{D})$ be a nonce based AEAD mode. Then, to define IND-CPA or INT-CTXT security, we have to consider $params = (N, A)$. Now consider the games for nAEAD, IND-CPA and INT-CTXT security.

Games G_3, G_4

```
Procedure Initialize

$K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}, S \leftarrow \Phi$

Procedure RR(M)

if $b = 0$ then
    $C = \mathbb{E}(K, N, M), S \leftarrow S \bigcup \{C\}$
else
    $C = f(K, N, M), S \leftarrow S \bigcup \{C\}$
end if
return $C$

Procedure Dec(C)

If $C \notin S$ then $M \leftarrow \mathcal{D}(K, N, C)$
else $M \leftarrow \perp$
if $M \neq \perp$ then
    $bad \leftarrow true$
    $M \leftarrow \perp$ // for $G_4$
end if
return $M$

Procedure Finalize

return $(d = b)$
```

Game G_5

```
Procedure Initialize

$K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}$

Procedure RR(M)

if $b = 0$ then
    $C = \mathbb{E}(K, N, M)$
else
    $C = f(K, N, M)$
end if
return $C$

Procedure Dec(C)

return $\perp$

Procedure Finalize

return $(d = b)$
```

Then using similar arguments as before, we have

$$Pr[nAEAD_{\mathbb{SE}}^{\mathcal{A}} \implies true] = Pr[G_3^{\mathcal{A}} \implies true]$$

$$= Pr[G_4^{\mathcal{A}} \implies true] + Pr[G_4^{\mathcal{A}}\text{sets bad}]$$

We define adversary $\mathcal{A}_c$ that chooses a bit b and plays real-or-random encryption game. For decryption query, irrespective of choice b or key, $\mathcal{A}_c$ returns $\perp$. Similarly we make adversary $\mathcal{A}_p$ that plays G_5 game. Then we have

$$Pr[G_4^{\mathcal{A}}\text{sets bad}] \leq Pr[INT - CTXT_{\mathcal{SE}}^{\mathcal{A}_c}\text{returns true}]$$

$$Pr[G_5^{\mathcal{A}}\text{returns true}] \leq Pr[IND - CPA_{\mathcal{SE}}^{\mathcal{A}_c}\text{returns true}]$$

Thus we can construct probabilistic polynomial time adversary for IND-CPA and INT-CTXT games. Thus we can conclude that if a scheme is not nAEAD, then it is not IND-CPA and not INT-CTXT. Thus by negating the statement, IND-CPA+INT-CTXT implies nAEAD security. This proves the complete theorem.

6.4 Comparison Table for AEAD modes

There are two comparison tables for AEAD Modes. We have covered a total of around seventy different variants of Authenticated Encryption of Associated Data Modes of Operation. We have compared 23 AEAD modes and their variants from CAESAR and NIST LWC competition's final lists and also enlisted 28 AEAD modes along with their variants from different conference papers worldwide. The table contains important CAESAR competition finalists which include Stream cipher based modes like ACORN, MORUS, etc; Block cipher based modes like AEGIS, AES-COPA, AES-JAMBU, SILC, etc.; Sponge construction based Modes like Ascon, KETJE etc., and Compression function based Mode like OMD. We have concluded about properties like parallelizability, online-ness, inverse-freeness or not after studying the constructions thoroughly. Security bounds and types are inferred from the papers' security claims. We have also include NIST-LWC finalists like Elephant, GIFT-COFB etc., and other constructions like SAEB, TEDT, FELICS-AEAD, SLAE, NonceWrap, Honey etc. We are also focusing on past publications of competitions like ASIACRYPT, EUROCRYPT, etc. We intended to cover up as many modes as possible, in order to make our survey complete. Security bounds are Auth + Priv, unless otherwise specified.

The security bound for a mode of operation can be expressed, in most of the papers, as $\leq X + Adv$(security of the underlying primitive). In this case, we have written down X in the column security bound. For simplicity, we wrote down the security type of the underlying primitive in the last column *Assump.*. The papers which phrases their security bounds in terms of bit-size, we wrote down their security assumption on underlying primitive in last column.

Assump.	Assumptions on the Security Notions of Underlying Primitives
SC	Stream Cipher Based Construction
BC	Block Cipher Based Construction
Sp	Sponge Construction
CF	Compression Function Based Construction
Primitive	Underlying Cipher, if any
O	Online
P	Parallelizable
P(2)	Parallelizable considering 2 Blocks at a time
I	Inverse-Free
b, B	Bits, Bytes
	(Denoting the corr. sizes in Columns *Message, Key, State Size*)
R	Random
N	Nonce
K	Key Used in the Schemes
k	Key Size
nAEAD	security of nonce based AEAD
OAE	online resistant AE security
RAE	robust AE security
n-b Auth	n-bit Authentication Security
n	Size of Message Block
E_K	Encryption Algorithm
t	Tag Size
q_e	Number of Encryption Queries
q_d	Number of Decryption Queries
σ_e	Number of Encryption Block Queries
σ_d	Number of Decryption Block Queries
σ	Total number of Block Queries
	(Encryption, Decryption and Verification)
q'	Number of Incomplete Block Queries
σ_A	Total number of Queries for Tag
muCIML	MultiUser Ciphertext Integrity Advantage with Misuse Resistance
q_{IC}	Number of Message Queries to Tag Generation Oracle
$q_{t'}$	Total Number of Forgery Queries
u	Maximum Number of Blocks in Message
TBC	Tweakable Block Cipher
q_v	Number of Verification Queries
q_f	Number of Tag-Forgery Queries
q_ρ, q_F, q_{LF}	Number of queries to the oracles of ρ, F, LF
	(Note that. $q_T = (l+1)(q_F + q_{LF}) + q_\rho$)
$\mathcal{IV}$	Initialization Vector Space
l	Sum of Block lengths for each $(\mathcal{A}, \mathcal{C})$ pair during Adversary Queries.
Q	Total bit of Queried Messages
$E\Lambda$	Probability of Tag-Forgery Attack
s	Size of S-permutation of AES

Mode	Variants	Primitive	Props	Message	Key	State Size	Rate	IV/Nonce type	Security Type	Security Bound	Assump.
ACORN[SC]			O,I,P	$< 2^{64}$b	128b	293b		128b, RN	nAEAD	128b auth	
AEGIS[BC]	128L 256 128		O,I,P O,I,P O,I,P	$< 2^{64}$b $< 2^{64}$b $< 2^{64}$b	128b 256b 128b	128B 256b 128b	1	128b, RN 256b, RN 128b, RN	nAEAD	128-b authentication security	PRP-CCA
AES-COPA[BC]			O,P	$< 2^{64}$b	128/192/256b	128b	$\frac{1}{2}$	128b, N	OAE	$\frac{39(\sigma+q)^2}{2^n} + \frac{2q}{2^n} + \frac{(l+2)(q-1)}{2^n}$:	PRP-CCA
AES-JAMBU[BC]			O,I	$< 2^{64}$b	128b	192b		64b, N	OAE	$\frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n,M)}{2^n} + \frac{qM}{2^n}$[nonce respecting], $\frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n,M)}{2^n}$[nonce misusing]	PRP-CCA
AEZ[BC]			I,P	$< 2^{64}$b	arbitrary	arbitrary		arbitrary	RAE,MRAE	$\equiv \frac{4s^2}{2^{110}} + \frac{t}{2^{128}}$	secure PRF
Ascon[Sp]			O,I	arbitrary	128b	$\lvert p^b \rvert$	$\frac{1}{4}$	Fixed	nAEAD		
SILC[BC]		AES128 PRESENT80 LED80	O,I		128b 80b 80b	$\lvert E_K\rvert + 128 + \lvert len(A)\rvert + \lvert len(C)\rvert$ $\lvert E_K\rvert + 80 + \lvert len(A)\rvert + \lvert len(C)\rvert$ $\lvert E_K\rvert + 80 + \lvert len(A)\rvert + \lvert len(C)\rvert$	1	64/96/112b 32/48b 32/48b	nAEAD	$\frac{5\sigma_{priv}^2}{2^n}$ [confidentiality] $\frac{5\sigma_{auth}^2}{2^n} + \frac{q'}{2^\tau}$[Authentication]	PRP-CPA
Deoxys[BC]		TBC	O,I		128/256b	$\lvert E_K\rvert + \lceil \lvert A\rvert/n \rceil + \lvert K\rvert$	1	nonce[N]:64/128b	nAEAD,MRAE	Birthday Bound	TPRP
CLOC[BC]		AES128 TWINE80	O,I		128b 80b	$\lvert E_K\rvert + \lvert K\rvert + \lvert V\rvert + \lvert N\rvert$ $\lvert E_K\rvert + \lvert K\rvert + \lvert V\rvert + \lvert N\rvert$	1	nonce[N]:1-15B nonce[N]:1-7B	nAEAD	$\frac{5\sigma_{priv}^2}{2^n}$[confidentiality] $\frac{5\sigma_{auth}^2}{2^n} + \frac{q'}{2^\tau}$ [Authentication]	PRP-CCA
COLM[BC]			O		128b	128b	$\frac{1}{2}$	nonce[N]:64b	OAE	birthday bound	PRP-CPA
TIAOXIN[BC]				$\leq 2^{128} - 1$b	128b				nAEAD	upto 2^{128}b	PRP-CCA
KETJE[Sp]		Keccak-p[400]	O,I		$\leq 182b$	50B	1	$(182 - \lvert K\rvert)$b	nAEAD		secure Hash
KEYAK[Sp]	RIVER	Keccak-p	O		≥ 128b	96B		58B,N	nAEAD	$\frac{q_v N}{2^{\lvert K\rvert}} + \frac{MN}{2^c} + \frac{M^2}{2^{c+1}} + \frac{q_f}{2^t}$	
MORUS[SC]	640-128 1280-128 1280-256		O,I	$< 2^{64}$b	128b 128b 256b	640b 1280b 1280b		128b, N	nAEAD	128b[auth+priv] 128b[auth+priv] 128b[auth],256b[priv]	
NORX[Sp]	NORX32 NORX64				128b 256b			64b, N 128b, N	nAEAD	128b[confidentiality] 256b[confidentiality]	
Elephant[SC]	Dumbo Jumbo Delirium	Spongent-π[160] Spongent-π[176] Keccak-f[200]	O,I,P	160b 176b 200b	128b 128b 128b	$(3n+m)$ 576b 624b 696b	1	$\frac{1}{2}$ 96b, N	nAEAD (nonce respecting)	$\frac{l}{2^n}\binom{q_e}{2} + \frac{2^{n-t}q_d}{2^n-1}e^{(q_e+1)q_e/2^n} + \frac{4\sigma^2+4\sigma p+4\sigma+p}{2^n} + \frac{p}{2^k}$ Auth. + Conf.	TPRP
GIFT-COFB[BC]		GIFT-128	I	arbitrary	128b	320b	1	128b, N	IND-CPA INT-CTXT	$\binom{\sigma_e}{2^{2^{128}}}$[IND-CPA] $\frac{3\sigma_e+q_d}{2^{64}} + \frac{(q_e+\sigma_e+2\sigma_f)\sigma_f+q_e q_d}{2^{128}} + \frac{64q_d}{2^{64}}$[IND-CTXT]	PRP-CPA
Grain-128 AEAD [SC]		GRAIN		arbitrary	128b	160b		96b, N	nonce respecting AEAD		PRP-CPA
ISAP[Sp]			O,I		128b	400b	$\frac{8}{25}$	128b, N	nAEAD		PRP-CCA
Photon-Beetle	AEAD Hash		O,I,P	arbitrary	128b	256b	$\frac{1}{2}$	128b, N	nonce respecting AEAD	$\frac{\sigma_e^2}{2^{256}} + \frac{q}{2^{128}} + \frac{q.q_e}{2^{256}}$[IND-CPA], $\frac{q(q_e+q_d)}{2^{256}} + \frac{q}{2^{121}} + \frac{q^{128}}{2^{128.127}} + \frac{128.\sigma_d}{2^{128}}$[INT-CTXT] $\frac{q_e^2}{2^{128}}$[Hash-collision], $\frac{q_e}{2^128}$[Preimage Security]	PRP-CPA
Sparkle(BC)	SCHWAEMM-128-128 SCHWAEMM-256-128 SCHWAEMM-192-192 SCHWAEMM-256-256	Sparkle256 Sparkle384 Sparkle384 Sparkle512	O O O O	$\leq 2^{71}$b $\leq 2^{71}$b $\leq 2^{71}$b $\leq 2^{136}$b	128b 128b 192b 256b	256b 384b 384b 512b	$\frac{1}{2}$ $\frac{2}{3}$ $\frac{1}{2}$ $\frac{1}{2}$	128b,N 256b,N 192b,N 256b,N	nAEAD (Non repeating nonce)	$\frac{\sigma_e^2}{2^{256}} + \frac{q}{2^{128}} + \frac{q.q_e}{2^{256}}$[IND-CPA], $\frac{q(q_e+q_d)}{2^{256}} + \frac{q}{2^{121}} + \frac{q^{128}}{2^{128.127}} + \frac{128.\sigma_d}{2^{128}}$[INT-CTXT] (basically same as Beetle)	PRP-CPA
TinyJambu	TinyJAMBU 128 TinyJAMBU 192 TinyJAMBU 256		O,I		128b 192b 256b	256b	$\frac{1}{4}$ (primary version)	96b,N	nAEAD	$\frac{\sigma_e^2}{2^{n+1}}$[Priv., Nonce Respecting Adversary] $(\frac{2.714\sigma_e}{\rho 2^r})^\rho . \frac{2^r}{\sqrt{\rho}} + \frac{(\sigma_e+\sigma_d)(\rho-2)}{2^(c+\nu)/2+1} + \frac{q_d}{2^t}$ [Auth., Nonce Misuse Adversary]	PRP-CPA

Table 5: Comparison Table 1 for AEAD Modes

Mode	Variants	Primitive	Props	Message	Key	State	Rate	IV/Nonce type	Security Type	Bound	Assump.
GCM			O,I,P	$\leq 2^{39} - 256$b	$\lvert K\rvert$	$\lvert N\rvert + \lvert E_K\rvert + \lvert K\rvert + \lvert H\rvert$	1	$< 2^{64}$b, N	IND-CPA+INT-CTXT	$\frac{0.5(\sigma+2q)^2+0.5q(\sigma+2q+1)(\lvert N\rvert+1)}{2^n}$ $+\frac{q(l+1)}{2^t}$ [Auth. + Priv.]	PRP-CCA
CCM			I,P		$\lvert K\rvert$	$\lvert N\rvert + \lvert E_K\rvert + \lvert K\rvert$	1	|blockcipher message|, N	IND-CPA+INT-CTXT	$\frac{\sigma^2}{2^n} + \frac{q_d}{2^t}$ [auth + priv]	PRP-CCA
SAEB[BC]			O,I		128b	128b	1	128b, N	nAEAD	$\frac{2\sigma^2}{2^n} + \frac{(\rho-1)(\sigma_A+\sigma_d)}{2^c} + 2^r(\frac{e\sigma_e}{\rho 2^r})^\rho + \frac{q_d}{2^t}$	
TEDT			I	2^{38}b	255b	$\lvert N\rvert + \lvert Enc^T\rvert + \lvert K\rvert + \lvert A\rvert$	$\frac{1}{2}$	96b, N	muCIML2+muCCAmL2	$\frac{(2n^2+17)(4\sigma+6(q_e+q_d)+2q_{IC})}{2^n} + \frac{u^2}{2^{2n}} + \frac{1}{(2n)!} + (\frac{4u}{2^n})^n$	LOR-CCA
Romulus	Romulus N1,N2,N3 Romulus M1,M2,M3	TBC TBC	O,I O,I		128b 128b	$n + 2.5k$,$n + 2.2k$,$n + 2k$ $n + 2.5k$,$n + 2.2k$,$n + 2k$	1 $\frac{2}{3}, \frac{7}{11}, \frac{7}{11}$	96/128b, N	nAEAD OAE	$\frac{3q_d}{2^n} + \frac{2q_d}{2^t}$[auth+priv] $\frac{5q_d}{2^n}$[auth+priv]	TPRP
Remus	Remus N1,N2,N3 Remus M1,M2	TBC TBC	O,I O,I		128b 128b	$n + k$,$2n + k$,$n + k$ $2n$,$3n$	1 $\frac{1}{2}$	96/128b N	nAEAD OAE	$\frac{9\sigma^2+4q_e\sigma+2q_e+3q_d}{2^n} + \frac{2q_d}{2^t}$[N1,N2], $\frac{\sigma^2}{2^{k-7}} + \frac{q_e\sigma}{2^k} + \frac{3q_d}{2^n} + \frac{2q_d}{2^t}$[N3] $\frac{9\sigma^2+4q_e\sigma}{2^n} + \frac{2q_e+5q_d}{2^n}$[M1,M2]	TPRP
Skinny-AEAD	M1,M2,M3,M4 M5,M6	SKINNY-128-384 SKINNY-128-256	O,P	2^{31}b	128b 128b		1 1	96/128b, N 96b, N		$\frac{2^{n-t}}{2^n-1}$[auth]	PRP-CPA
PFB	PFB_Plus $PFB\omega$	TBC TBC	O,I,P O,I,P	$\leq 2^n$b $\leq 2^n$b	$\lvert K\rvert$	$\lvert\tilde{E_K}\rvert + \lvert N\rvert$ $\lvert\tilde{E_K}\rvert + \lvert N\rvert + \frac{\lvert A\rvert}{n}$	1 1	depends on blockcipher, N	nAEAD nAEAD	$\frac{q_d 2^{n-t+1}}{(2^n-1)^2}$[auth+conf] $\frac{2^\omega q_d}{(2^n-1)^\omega}$	TPRP
OTR	parallel[P] serial[S]	AES	O,I,P	$< 2^{64}$b	128/192 /256b	$\lvert E_K\rvert + len(L)$	1	1-15B N	nAEAD	P:$\frac{6\sigma^2}{2^n}$ [Conf.], $\frac{6\sigma^2}{2^n} + \frac{q_v}{2^\tau}$[Auth.] S:$\frac{5.5\sigma^2}{2^n}$[Conf.], $\frac{7.5\sigma^2}{2^n} + \frac{q_v}{2^\tau}$[Auth.]	PRP-CPA
CWC_+			I,P		$\lvert K\rvert$		$\frac{1}{2}$	n-bit, N	nAEAD	$\frac{105\sigma^3 l}{2^{2n}} + \frac{6\sigma l}{2^n} + \frac{2q_d}{2^\rho}$ $+\frac{2q_d l}{2^n} + \frac{(2q_e+q_d)2l\mu}{2^n} + (\frac{5\sigma l\mu}{2^n})^2$	PRP-CPA
SLAE			O,I		$\lvert K_E\rvert$	$\lvert S\rvert + \lvert\rho\rvert$	1	N	LAE	$\frac{q_T(q_T+2)+(q_F+q_{LF})q_\rho}{2^{n-rr-1}} + \frac{2q_\rho}{2^{k-\lambda 2^{rr}}} + \frac{2lq_F q_\rho}{2^{n-\lambda 2^{rr}}}$	LPRF
Forkcipher-AEAD	PAEF RPAEF SAEF	ForkSKINNY	O,P O,P O		$\lvert K\rvert + \lvert N\rvert$	$\lvert F_K\rvert + \lvert T\rvert + \lvert S\rvert$	1 1 1	N	nAEAD	$\frac{q_v 2^n}{(2^n-1)^2}$ $\frac{2.q_v}{2^n-1}$ $\frac{(\sigma-q)^2}{2^{n-1}}$[priv],$\frac{(\sigma-q+1)^2}{2^n} + \frac{\sigma(\sigma-q)}{2^n} + \frac{q_v(q+2)}{2^n}$[auth]	PRTFP
NonceWrap				128b	$\lvert K_1\rvert + \lvert K_2\rvert$		$rate(\mathcal{E}_{K_1})$	96b, N	anAE	$\frac{q_e^2}{2^{n+1}} + \frac{q_e^2}{2^{\tau+1}} + \frac{q_e^2+q_d^2}{2^{\sigma+1}} + \frac{q_e^4}{2^{n+\tau+2}}$[anAE]	PRP-CCA
OCBv		TBC	O,P		$\lvert K\rvert$	$\lvert\tilde{E}_K^{N,\tau}\rvert + \lvert Sum\rvert$	1	N	nvAE	$\frac{28.5q^2}{2^n} + q_d^{\tau_c}\frac{2^{n-\tau_c}}{2^n-1}$	PRP-CCA
ZCZ		TBC	O,I,P(2)		$\lvert K\rvert$	$\lvert\tilde{E}_K\rvert + 4n$	1	N		$\frac{4\sigma^2+8q^2+9q'^2}{N^2}$	PRP-CCA
ELmD[BC]	$ELmD_{10,0,f}$ $ELmD_{10,127,f}$	AES-128	O	$\leq 2^{64}$b	128b		$\frac{1}{2}$	$128b$,RN	nAEAD	$\frac{5(\sigma+q)^2}{2^n}$[priv], $\frac{9(\sigma+q)^2}{2^n}$[auth] $\frac{91}{18}\frac{(\sigma+q)^2}{2^n}$[priv], $\frac{81}{8}\frac{(\sigma+q)^2}{2^n}$[auth]	PRP-CCA
SUNDAE[BC]			I		$\lvert K\rvert$	$\lvert E_k\rvert + \lvert K\rvert$	$\frac{1}{2}$	n-b,RN	DAE	$\frac{(4+\sigma_A+2\sigma_e+2\sigma_d)^2}{2^{n+1}} + \frac{q_d+q_e^2+q_d q_e}{2^n} + \frac{(\sigma_e+\sigma_d)^2}{2^{n+1}}$ $+\frac{4(\sigma_d+\sigma_e)+(4+\sigma_A+\sigma_e+\sigma_d)^2+4(q_e+q_d)^2}{2^n}$	PRP-CPA
Counter-in-Tweak		TBC			$\lvert K\rvert$	$\lvert\tilde{E}_K\rvert + \lvert K\rvert + \lvert N\rvert$	1	N	nAEAD	$\frac{\sigma^2}{2^{n+1}\lvert\mathcal{IV}\rvert}$[nAE], $\frac{2(m-1)\sigma}{\lvert\mathcal{IV}\rvert} + \frac{\sigma^2}{2^{n+1}\lvert\mathcal{IV}\rvert}$[MRAE]	TPRP
McOE	McOE-X McOE-G	AES-CBC, AES,Treefish	O O		$\lvert K_1\rvert$ $\lvert K_1\rvert + \lvert K_2\rvert$	$\lvert E_K\rvert + \lvert K_1\rvert$ $\lvert E_K\rvert + \lvert K_1\rvert + \lvert K_2\rvert + \lvert H\rvert$	1 1	N N	OAE ,CCA_3	$\frac{3(q+l)(q+l+1)+4q+3l}{2^n-(q+l)}$ $[CCA_3]$	PRP-CCA
RIV		AES-128, Deoxys-BC-128	O	128 b	$\lvert K_1\rvert + \lvert K_2\rvert$	$\lvert\mathcal{E}_K\rvert + \lvert IV\rvert + \lvert K_1\rvert + \lvert K_2\rvert$	1	N	nAEAD, SAE	$\frac{8q^2+3q}{2^n}$[SAE], $\frac{2q^2+q}{2^n}$[nAEAD]	PRP-CPA
p-OMD			O,I	any	$\lvert K\rvert$	$\lvert F_K\rvert + \lvert\Delta\rvert + \lvert K\rvert$	1	N	nAEAD (Nonce Resp.)	$\frac{3\sigma^2}{2^n}$[priv], $\frac{3\sigma^2}{2^n} + \frac{q_v l}{2^n} + \frac{q_v}{2^t}$[auth]	PRP-CPA(PRF)
APE			O		$\lvert K\rvert$	$\lvert V\rvert + \lvert p\rvert + \lvert K\rvert$	1	NA	IND-CPA+INT-CTXT	$\frac{Q^2}{2^{n+t}} + \frac{Q(Q+1)}{2^t}$[priv], $\frac{Q^2}{2^{n+t}} + \frac{2Q(Q+1)}{2^t}$[auth]	PRP-CPA
COBRA			O,I,P		$\lvert K\rvert$	$\lvert L\rvert + \lvert E_K\rvert + \lvert K\rvert$	1	N	IND-CPA+INT-CTXT	$\frac{22(l+1)^2q^2}{2^n}$[priv/ind-cpa], $\frac{3(q+1)q_{t'}}{2^n} + \frac{22(l+1)^2q^2}{2^n}$[auth/int-ctxt]	PRP-CPA
TriviA(SC)		Trivia-SC	O,I	$\leq 2^{64}$b	128b	$384 + 128 + 128$b		N,128b	nAEAD	$\frac{q}{2^{129}}$[priv], $\frac{q}{2^{129}} + \frac{q_f}{2^1 24}$[auth]	PRP-CPA
ALE(BC)		AES-128	O,I	$\leq 2^{48}$b	128b	$\lvert AES\rvert + \lvert key - schedule\rvert$ $+32b + 128b$	$\frac{1}{4}$	128b, N	nAEAD		PRP-CCA
ESTATE(BC)		TweAES, TweGIFT	I	$\leq 2^{n/2}$	$\lvert K\rvert$	260b	$\frac{1}{2}$	n-b, N	MRAE, RUP	$\frac{\sigma^2}{2^n} + \frac{q_v}{2^n}$[RUP]	TPRP

Table 6: Comparison Table 2 for AEAD Modes

6.5 Suitable Data Specifications

In this section, we shall discuss performance of AEAD modes based on specifications of data to be encrypted. That is, we shall conclude which type of data best fits for the modes. Also, we wish to discuss which mode is optimal in terms of rate, state size etc.

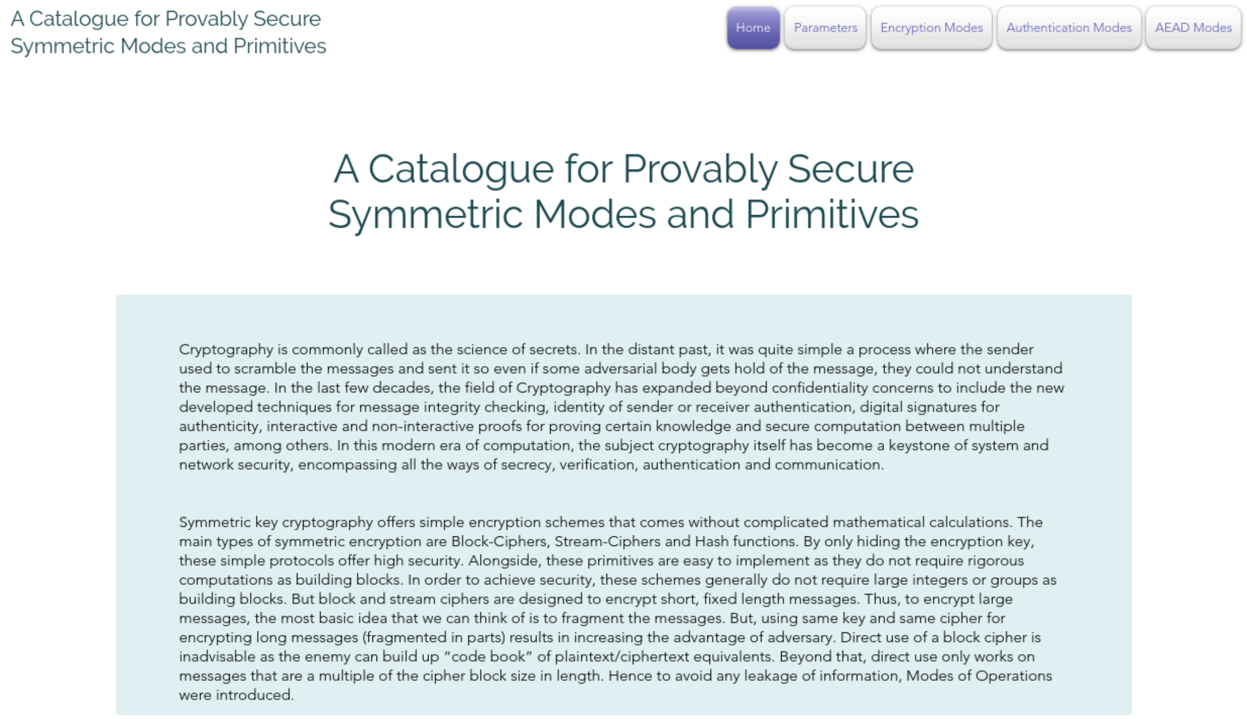
ACORN	Software and Hardware-efficient, No padding needed
AEGIS	Low computational complexity, Same encryption and decryption algorithm
AES-JAMBU	Nonce-misuse security and online achieved together. Hardware efficient.
AEZ	Strongest form of nonce reuse security achieved, suitable for low-power devices, hardware and software efficient.
ASCON	Resistant to linear and differential cryptanalysis
SILC	Hardware efficient as no conditional branching operation needed. Optimal rate
Deoxys	Resistant to side-channel attack. Efficient for small messages. Software efficient.
CLOC	Does not use finite-field multiplication.
COLM	Fully parallelizable. Online. Secure against nonce reusing adversary.
Ketje	Small state size. Side-channel leakage resistant.
NORX	High security and efficiency. Hardware friendly.
Elephant	Optimal rate.
GIFT-COFB	Suitable for low energy devices. Hardware efficient. Optimal rate.
GCM	Used to compare other AEAD modes. Hardware efficient. Fully parallelizable.
CCM	Widely used. Provides good security results.
Romulus and Remus	Security against nonce repeating adversary. small state size. One variant has optimal rate. Hardware efficient.
SKINNY-AEAD	No potential differential attack. Small state size. Hardware efficient.
OTR	Optimal rate. Online.
Forkcipher -AEAD	Optimal rate. Online. Hardware efficient
ELmD	Efficient, Nonce misuse resistant. Has EME structure which makes implementation-friendly
SUNDAE	Gives good security bound. Software efficient.
Counter-in-Tweak	Simple construction. Optimal rate. Good security bound.
APE	First nonce misuse resistant sponge based scheme. Optimal rate.
COBRA	Secure against nonce repeating adversary. Optimal rate.
ESTATE	Hardware efficient. Suitable for low-energy device.

7 Web Implementation

From the initial time of the project, we intended to make our work available in a public domain, which can help other cryptograhic designers to use the data from our data platform. We have temporarily have done some web-implementation in the Wixsite domain, which we believe should be later shifted to some permanent web address of some academic organisation. Currently, the whole work can be found in the following address.

`https://srianishadutta.wixsite.com/cryptolux1`

We are maintaining the above website to outsource the comparison tables and our conclusions in tabular format. Currently this webpage contains four main pages with detailed discussions of our work and a main page to navigate all those. The page entitled 'Parameters' contains all the necessary details about the efficiency parameters and various security definitions, which we shall use in the comparison tables. The next page, entitled 'Encryption Modes' contains the comparison table for various encryption modes along with the indices of parameters, those are used. Similarly, we have designed the next two webpages, namely 'Authentication Modes' and 'AEAD Modes'.


A Catalogue for Provably Secure
Symmetric Modes and Primitives
Home
Parameters
Encryption Modes
Authentication Modes
AEAD Modes

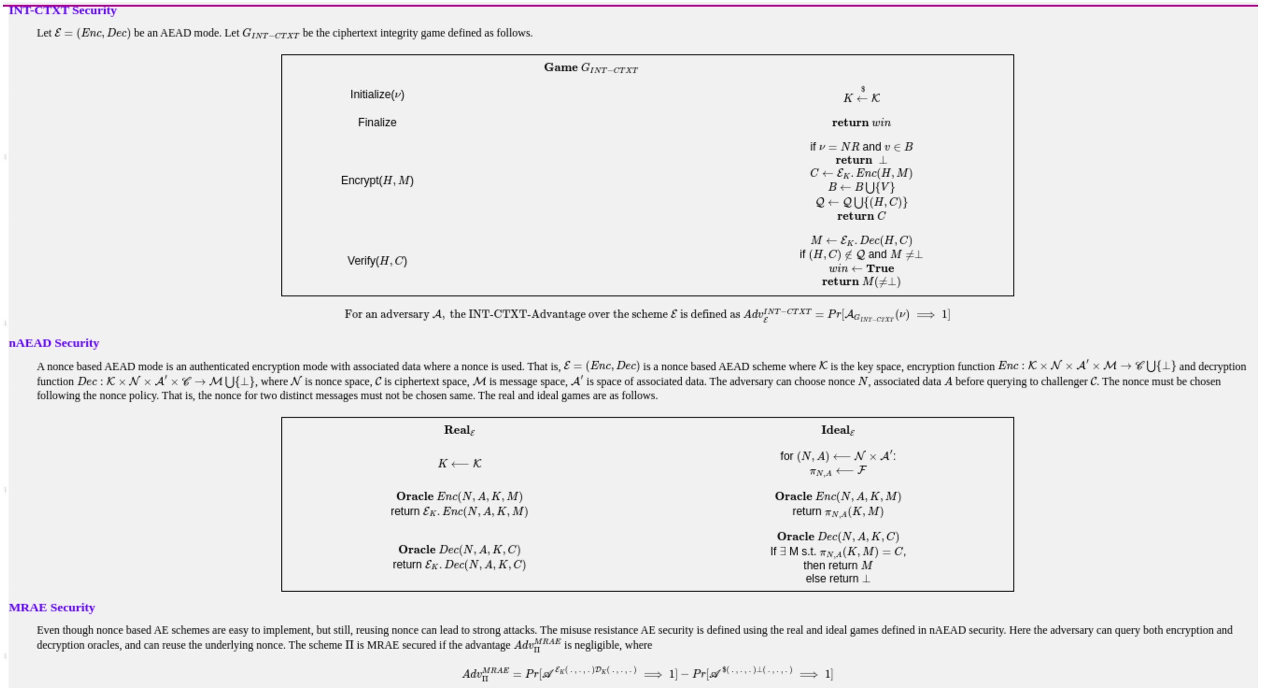
A Catalogue for Provably Secure
Symmetric Modes and Primitives

Cryptography is commonly called as the science of secrets. In the distant past, it was quite simple a process where the sender used to scramble the messages and sent it so even if some adversarial body gets hold of the message, they could not understand the message. In the last few decades, the field of Cryptography has expanded beyond confidentiality concerns to include the new developed techniques for message integrity checking, identity of sender or receiver authentication, digital signatures for authenticity, interactive and non-interactive proofs for proving certain knowledge and secure computation between multiple parties, among others. In this modern era of computation, the subject cryptography itself has become a keystone of system and network security, encompassing all the ways of secrecy, verification, authentication and communication.

Symmetric key cryptography offers simple encryption schemes that comes without complicated mathematical calculations. The main types of symmetric encryption are Block-Ciphers, Stream-Ciphers and Hash functions. By only hiding the encryption key, these simple protocols offer high security. Alongside, these primitives are easy to implement as they do not require rigorous computations as building blocks. In order to achieve security, these schemes generally do not require large integers or groups as building blocks. But block and stream ciphers are designed to encrypt short, fixed length messages. Thus, to encrypt large messages, the most basic idea that we can think of is to fragment the messages. But, using same key and same cipher for encrypting long messages (fragmented in parts) results in increasing the advantage of adversary. Direct use of a block cipher is inadvisable as the enemy can build up "code book" of plaintext/ciphertext equivalents. Beyond that, direct use only works on messages that are a multiple of the cipher block size in length. Hence to avoid any leakage of information, Modes of Operations were introduced.

Figure 1: Webpage for Provable Symmetric Modes

The above figure is a glimpse of the homepage of the mentioned web-platform. The navigation buttons to move to the other pages are given on the top right of the page. The next four figures in this section clearly capture the way we design the security definitions and efficiency parameters, as well as the comparison tables for the Symmetric Modes of Operations category wise. We further intend to refine the webpage with more filters and data in future.



INT-CTXT Security

Let $\mathcal{E} = (Enc, Dec)$ be an AEAD mode. Let $G_{INT-CTXT}$ be the ciphertext integrity game defined as follows.

Game $G_{INT-CTXT}$	
Initialize(ν)	$K \overset{\$}{\leftarrow} \mathcal{K}$
Finalize	**return** win

Encrypt(H, M)	if $\nu = NR$ and $v \in B$
	return $\perp$
	$C \leftarrow \mathcal{E}_K.Enc(H, M)$
	$B \leftarrow B \bigcup \{V\}$
	$\mathcal{Q} \leftarrow \mathcal{Q} \bigcup \{(H, C)\}$
	return C

Verify(H, C)	$M \leftarrow \mathcal{E}_K.Dec(H, C)$
	if $(H, C) \notin \mathcal{Q}$ and $M \neq \perp$
	$win \leftarrow$ **True**
	return $M(\neq \perp)$

For an adversary $\mathcal{A}$, the INT-CTXT-Advantage over the scheme $\mathcal{E}$ is defined as $Adv_{\mathcal{E}}^{INT-CTXT} = Pr[\mathcal{A}_{G_{INT-CTXT}}(\nu) \implies 1]$

nAEAD Security

A nonce based AEAD mode is an authenticated encryption mode with associated data where a nonce is used. That is, $\mathcal{E} = (Enc, Dec)$ is a nonce based AEAD scheme where $\mathcal{K}$ is the key space, encryption function $Enc : \mathcal{K} \times \mathcal{N} \times \mathcal{A}' \times \mathcal{M} \rightarrow \mathscr{C} \bigcup \{\perp\}$ and decryption function $Dec : \mathcal{K} \times \mathcal{N} \times \mathcal{A}' \times \mathscr{C} \rightarrow \mathcal{M} \bigcup \{\perp\}$, where $\mathcal{N}$ is nonce space, $\mathcal{C}$ is ciphertext space, $\mathcal{M}$ is message space, $\mathcal{A}'$ is space of associated data. The adversary can choose nonce N, associated data A before querying to challenger $\mathcal{C}$. The nonce must be chosen following the nonce policy. That is, the nonce for two distinct messages must not be chosen same. The real and ideal games are as follows.

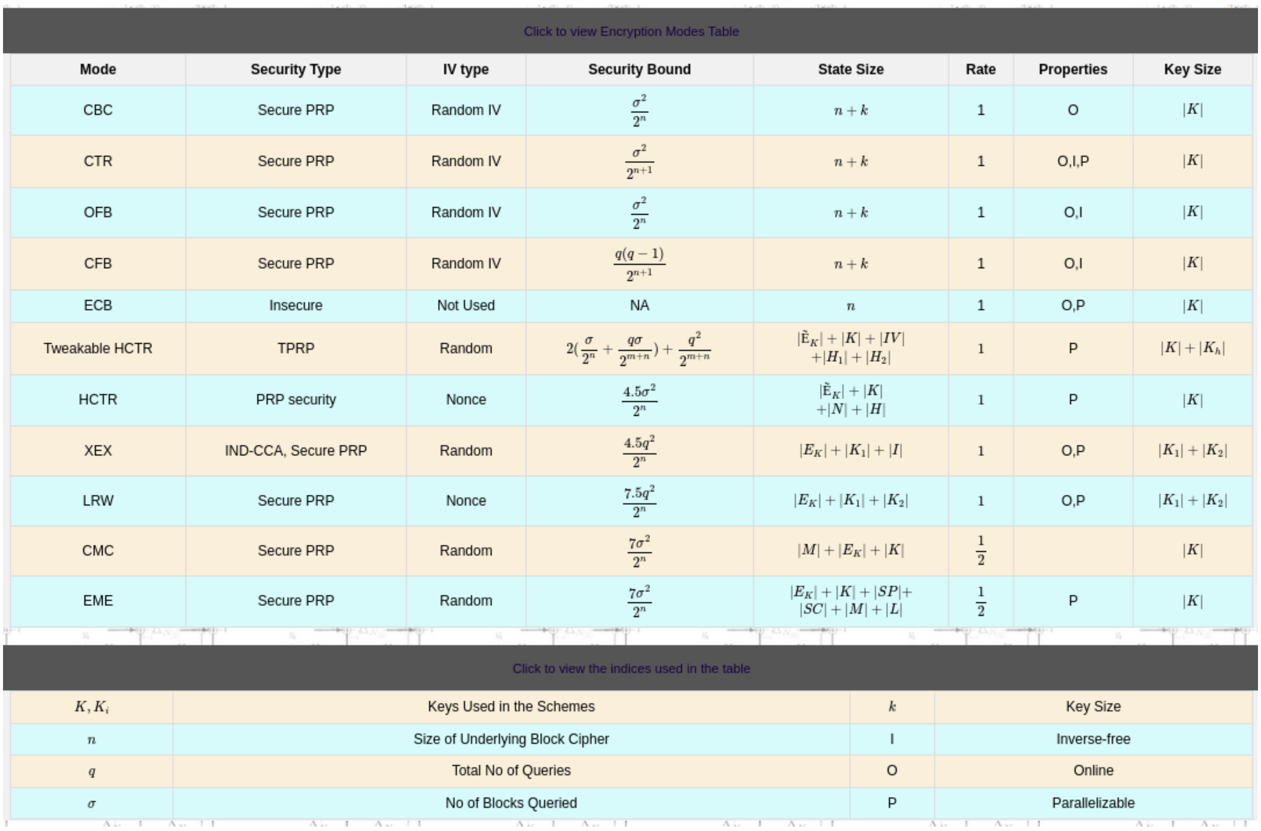
Real$_{\mathcal{E}}$	**Ideal**$_{\mathcal{E}}$
$K \longleftarrow \mathcal{K}$	for $(N, A) \longleftarrow \mathcal{N} \times \mathcal{A}'$: $\pi_{N,A} \longleftarrow \mathcal{F}$
Oracle $Enc(N, A, K, M)$ return $\mathcal{E}_K.Enc(N, A, K, M)$	**Oracle** $Enc(N, A, K, M)$ return $\pi_{N,A}(K, M)$
Oracle $Dec(N, A, K, C)$ return $\mathcal{E}_K.Dec(N, A, K, C)$	**Oracle** $Dec(N, A, K, C)$ If $\exists$ M s.t. $\pi_{N,A}(K, M) = C$, then return M else return $\perp$

MRAE Security

Even though nonce based AE schemes are easy to implement, but still, reusing nonce can lead to strong attacks. The misuse resistance AE security is defined using the real and ideal games defined in nAEAD security. Here the adversary can query both encryption and decryption oracles, and can reuse the underlying nonce. The scheme Π is MRAE secured if the advantage Adv_{Π}^{MRAE} is negligible, where

$$Adv_{\Pi}^{MRAE} = Pr[\mathscr{A}^{\mathcal{E}_K(.,.,.)\mathcal{D}_K(.,.,.)} \implies 1] - Pr[\mathscr{A}^{\$(.,.,.)\perp(.,.,.)} \implies 1]$$

Figure 2: Glimpse of Security Definitions



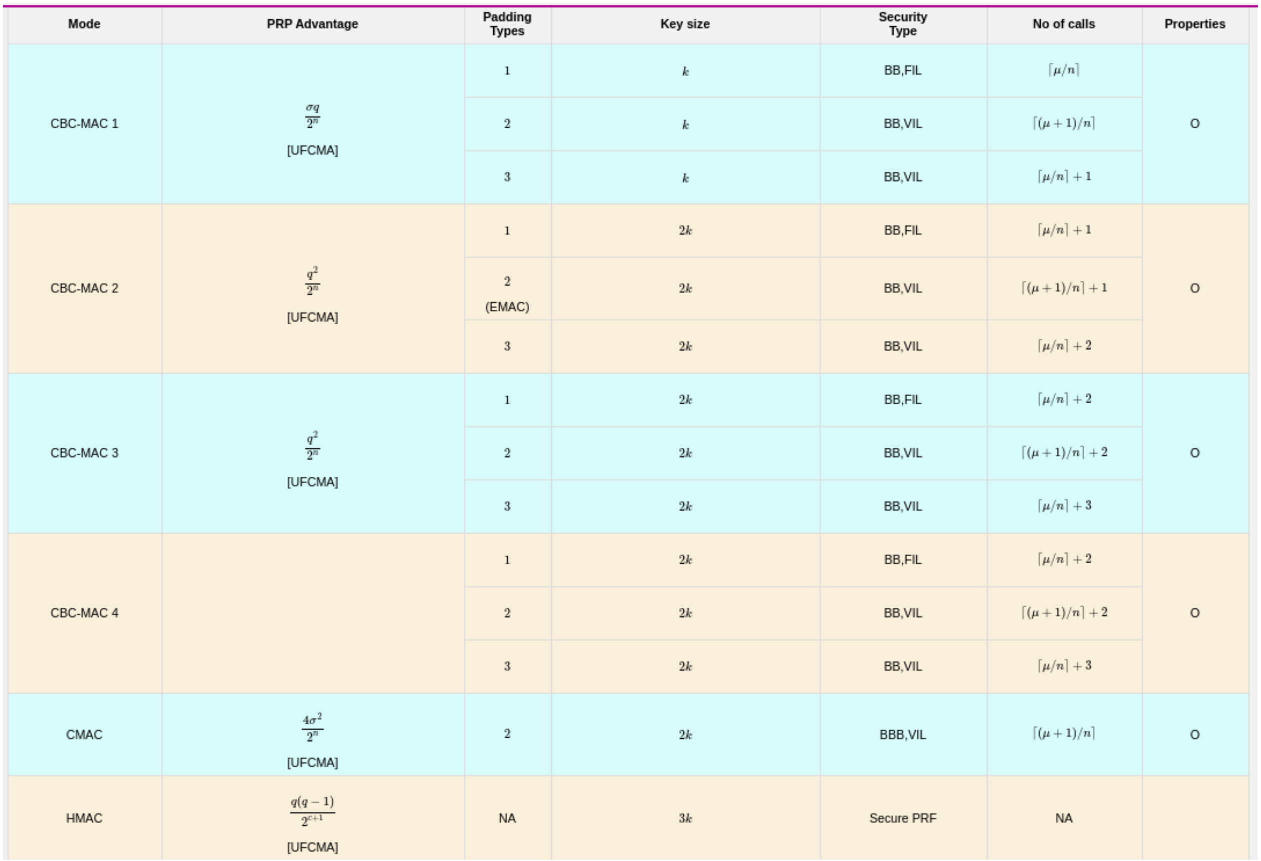
Click to view Encryption Modes Table

Mode	Security Type	IV type	Security Bound	State Size	Rate	Properties	Key Size
CBC	Secure PRP	Random IV	$\frac{\sigma^2}{2^n}$	$n+k$	1	O	$\|K\|$
CTR	Secure PRP	Random IV	$\frac{\sigma^2}{2^{n+1}}$	$n+k$	1	O,I,P	$\|K\|$
OFB	Secure PRP	Random IV	$\frac{\sigma^2}{2^n}$	$n+k$	1	O,I	$\|K\|$
CFB	Secure PRP	Random IV	$\frac{q(q-1)}{2^{n+1}}$	$n+k$	1	O,I	$\|K\|$
ECB	Insecure	Not Used	NA	n	1	O,P	$\|K\|$
Tweakable HCTR	TPRP	Random	$2(\frac{\sigma}{2^n} + \frac{q\sigma}{2^{m+n}}) + \frac{q^2}{2^{m+n}}$	$\|\tilde{E}_K\| + \|K\| + \|IV\|$ $+\|H_1\| + \|H_2\|$	1	P	$\|K\| + \|K_h\|$
HCTR	PRP security	Nonce	$\frac{4.5\sigma^2}{2^n}$	$\|\tilde{E}_K\| + \|K\|$ $+\|N\| + \|H\|$	1	P	$\|K\|$
XEX	IND-CCA, Secure PRP	Random	$\frac{4.5q^2}{2^n}$	$\|E_K\| + \|K_1\| + \|I\|$	1	O,P	$\|K_1\| + \|K_2\|$
LRW	Secure PRP	Nonce	$\frac{7.5q^2}{2^n}$	$\|E_K\| + \|K_1\| + \|K_2\|$	1	O,P	$\|K_1\| + \|K_2\|$
CMC	Secure PRP	Random	$\frac{7\sigma^2}{2^n}$	$\|M\| + \|E_K\| + \|K\|$	$\frac{1}{2}$		$\|K\|$
EME	Secure PRP	Random	$\frac{7\sigma^2}{2^n}$	$\|E_K\| + \|K\| + \|SP\|+$ $\|SC\| + \|M\| + \|L\|$	$\frac{1}{2}$	P	$\|K\|$

Click to view the indices used in the table

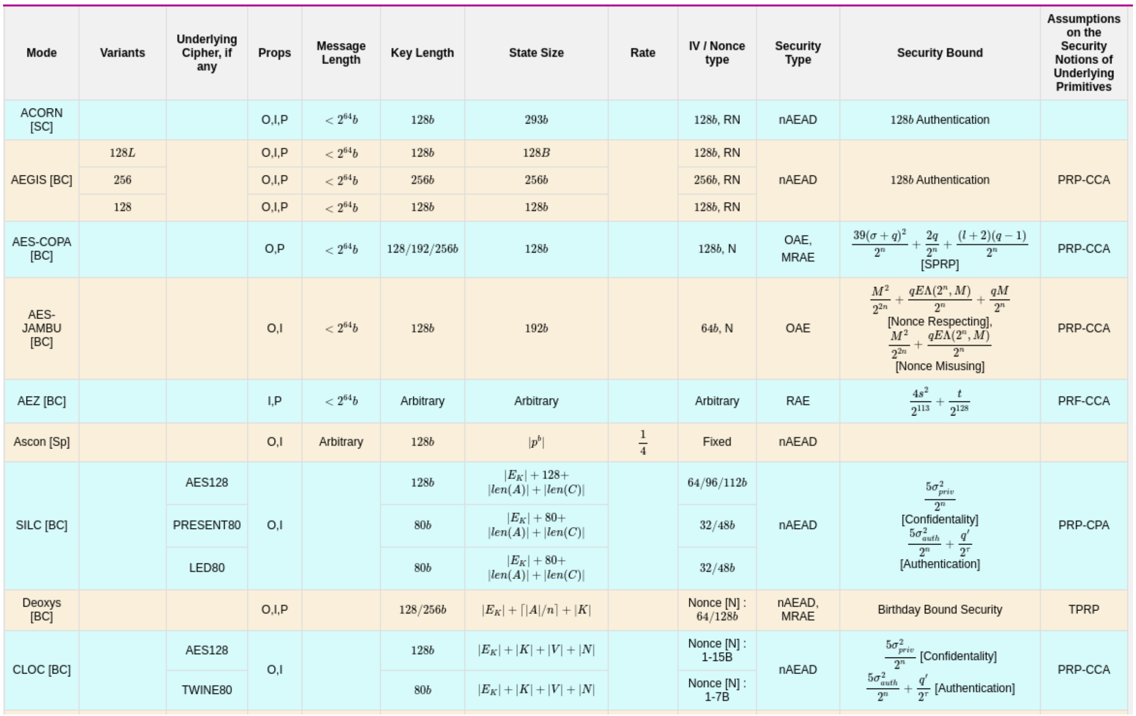
K, K_i	Keys Used in the Schemes	k	Key Size
n	Size of Underlying Block Cipher	I	Inverse-free
q	Total No of Queries	O	Online
σ	No of Blocks Queried	P	Parallelizable

Figure 3: Comparison List and Parameters for Encryption Modes



Mode	PRP Advantage	Padding Types	Key size	Security Type	No of calls	Properties
CBC-MAC 1	$\frac{\sigma q}{2^n}$ [UFCMA]	1	k	BB,FIL	$\lceil \mu/n \rceil$	O
		2	k	BB,VIL	$\lceil (\mu+1)/n \rceil$	
		3	k	BB,VIL	$\lceil \mu/n \rceil + 1$	
CBC-MAC 2	$\frac{q^2}{2^n}$ [UFCMA]	1	$2k$	BB,FIL	$\lceil \mu/n \rceil + 1$	O
		2 (EMAC)	$2k$	BB,VIL	$\lceil (\mu+1)/n \rceil + 1$	
		3	$2k$	BB,VIL	$\lceil \mu/n \rceil + 2$	
CBC-MAC 3	$\frac{q^2}{2^n}$ [UFCMA]	1	$2k$	BB,FIL	$\lceil \mu/n \rceil + 2$	O
		2	$2k$	BB,VIL	$\lceil (\mu+1)/n \rceil + 2$	
		3	$2k$	BB,VIL	$\lceil \mu/n \rceil + 3$	
CBC-MAC 4		1	$2k$	BB,FIL	$\lceil \mu/n \rceil + 2$	O
		2	$2k$	BB,VIL	$\lceil (\mu+1)/n \rceil + 2$	
		3	$2k$	BB,VIL	$\lceil \mu/n \rceil + 3$	
CMAC	$\frac{4\sigma^2}{2^n}$ [UFCMA]	2	$2k$	BBB,VIL	$\lceil (\mu+1)/n \rceil$	O
HMAC	$\frac{q(q-1)}{2^{c+1}}$ [UFCMA]	NA	$3k$	Secure PRF	NA	

Figure 4: Part of Comparison List for Authentication Modes



Mode	Variants	Underlying Cipher, if any	Props	Message Length	Key Length	State Size	Rate	IV / Nonce type	Security Type	Security Bound	Assumptions on the Security Notions of Underlying Primitives
ACORN [SC]			O,I,P	$< 2^{64}b$	$128b$	$293b$		$128b$, RN	nAEAD	$128b$ Authentication	
AEGIS [BC]	$128L$		O,I,P	$< 2^{64}b$	$128b$	$128B$		$128b$, RN	nAEAD	$128b$ Authentication	PRP-CCA
	256		O,I,P	$< 2^{64}b$	$256b$	$256b$		$256b$, RN			
	128		O,I,P	$< 2^{64}b$	$128b$	$128b$		$128b$, RN			
AES-COPA [BC]			O,P	$< 2^{64}b$	$128/192/256b$	$128b$		$128b$, N	OAE, MRAE	$\frac{39(\sigma+q)^2}{2^n} + \frac{2q}{2^n} + \frac{(l+2)(q-1)}{2^n}$ [SPRP]	PRP-CCA
AES-JAMBU [BC]			O,I	$< 2^{64}b$	$128b$	$192b$		$64b$, N	OAE	$\frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n, M)}{2^n} + \frac{qM}{2^n}$ [Nonce Respecting], $\frac{M^2}{2^{2n}} + \frac{qE\Lambda(2^n, M)}{2^n}$ [Nonce Misusing]	PRP-CCA
AEZ [BC]			I,P	$< 2^{64}b$	Arbitrary	Arbitrary		Arbitrary	RAE	$\frac{4s^2}{2^{113}} + \frac{t}{2^{128}}$	PRF-CCA
Ascon [Sp]			O,I	Arbitrary	$128b$	$\lvert p^b \rvert$	$\frac{1}{4}$	Fixed	nAEAD		
SILC [BC]		AES128	O,I		$128b$	$\lvert E_K \rvert + 128 +$ $\lvert len(A) \rvert + \lvert len(C) \rvert$		$64/96/112b$	nAEAD	$\frac{5\sigma_{priv}^2}{2^n}$ [Confidentality] $\frac{5\sigma_{auth}^2}{2^n} + \frac{q'}{2^\tau}$ [Authentication]	PRP-CPA
		PRESENT80			$80b$	$\lvert E_K \rvert + 80 +$ $\lvert len(A) \rvert + \lvert len(C) \rvert$		$32/48b$			
		LED80			$80b$	$\lvert E_K \rvert + 80 +$ $\lvert len(A) \rvert + \lvert len(C) \rvert$		$32/48b$			
Deoxys [BC]			O,I,P		$128/256b$	$\lvert E_K \rvert + \lceil \lvert A \rvert / n \rceil + \lvert K \rvert$		Nonce [N] : $64/128b$	nAEAD, MRAE	Birthday Bound Security	TPRP
CLOC [BC]		AES128	O,I		$128b$	$\lvert E_K \rvert + \lvert K \rvert + \lvert V \rvert + \lvert N \rvert$		Nonce [N] : 1-15B	nAEAD	$\frac{5\sigma_{priv}^2}{2^n}$ [Confidentality] $\frac{5\sigma_{auth}^2}{2^n} + \frac{q'}{2^\tau}$ [Authentication]	PRP-CCA
		TWINE80			$80b$	$\lvert E_K \rvert + \lvert K \rvert + \lvert V \rvert + \lvert N \rvert$		Nonce [N] : 1-7B			

Figure 5: Part of Comparison List for AEAD Modes

We have enlisted almost all the available modes in those mentioned lists category wise. Since we shall be encountering new symmetric modes a lot in the future, we shall further wish to update this platform after a certain period of time, if possible. Also, we wish to further enrich our web-implementation with some other tools like having the option of filtering on specific features. For example, it shall be better to introduce some filter on AEAD Modes table as following so that one can sort the tables in quicker way to get the following information.

Optimal rate	AEGIS, SILC, Deoxys, CLOC, Ketje, Elephant, GIFT-COFB, GCM, CCM, SAEB, Skinny, PFB, OTR, SLAE, Forkcipher-AEAD, OCBv, ZCZ, Counter-in-tweak, McOE, RIV, p-OMD, APE, COBRA
Small State-size	SUNDAE, COBRA, Deoxys
Online	ACORN, AEGIS, Ascon, SILC, Deoxys, COLM, Ketje KEYAK, MORUS, Elephant, ISAP, Photon-Beetle, TinyJabmu, GCM, SAEB, Romulus, Remus, Skinny-AEAD, McOE, APE, COBRA
Parallelizable	ACORN, AEGIS, Elephant, Photon-Beetle, GCN, CCM, Skinny-AEAD, OTR, PFB, CWC_{+}, COBRA
MRAE-secure	AEZ, ESTATE

For encryption modes, the strongest security notion is CCA security notion. Also, it is obvious that if a mode is online, then its execution will be faster than offline ones. Thus we can construct table for encryption modes.

Beyond-birthday secure	Tweakable-HCTR, HCTR, XEX, LRW, CMC, EME
Parallel	CTR, ECB, HCTR, XEX, LRW
CCA secure	HCTR, XEX, LRW
Optimal rate	CBC, CTR, OFB, CFB, HCTR, XEX, LRW

The table for authentication modes can also compare least number of block cipher calls, parallelizability, beyond birthday security etc. This further developments in the webpage shall surely be more beneficial for the reader.

Beyond-birthday secure	CMAC, LightMAC, nEHtM, PMACx, $PMAC_{plus}$, DoveMAC
Parallel	PMAC, LightMAC, nEHtm, PMACx, ZMAC, $ZMAC_{+}$
Least no. of calls to underlying cipher	PMACx, $PMAC_{+}$, Sum-of-CBCs, OMAC, PMAC, LightMAC

8 Final Notes

This project was intended to be a systematic study of Provable Symmetric Primitives. We aimed to finally store all the data including the studied primitives, comparison tables, etc in a platform ,which further help others to make their work easier. I have been introduced to the idea of creating such a catalogue for these primitives through a report by Phillip Rogaway, namely Evaluation of Some Blockcipher Modes of Operation. This project thoroughly studies different encryption, authentication and AEAD modes till 2011. This covers various modes from all three types and somewhat compared them through different parameters. This work motivated me a lot into forming my idea into a practical domain. Since, in last decade, the introduction of various world-wide competitions and conferences encouraged the cryptographers to further work in this domain, the volume of research has been astonishingly increased since. Hence, it has been high time to bind them under one roof so that one can have a systematic study through those primitives.

I started this project with studying the basics of symmetric-key cryptography and provable security from the book Modern Cryptography by Bellare and Rogaway [1]. This was followed by a literature survey and comparative analysis in various Block cipher modes of operation. This includes classical as well as modern modes of encryption, authentication and authenticated encryption. My comparative analysis targets various parameters for these modes. This includes the primitive types, security related results (such as security notions, security bounds with or without allowing nonce-repetitions), efficiency related features (such as inverse-free, online support, parallelizability and many more). I have made tables to compare various modes of encryption, authentication and AEAD modes. For all of these modes, I have referred to Wikipedia , FIPS standards, Google scholarly articles, the candidates from the NIST LWC standardization process [6] and the CAESAR competition [8]. Our work also includes the NIST-LWC finalists which were announced 29^{th} of March, 2021 by the concerned committee. These modes are mainly authenticated encryption modes with associated data.

Each of the modes are motivated from its own point of view. Thus every single mode has different goal. This leads each of them to have different types and levels of security and different construction. It took a lot of patience and effort to understand the constructions and realize each security definition. Moreover, the security proofs(wherever available) mainly use various games – analysis of which were also time-consuming. Some of the papers does not have any proof of the security bounds they specify, which lead me to dig deeper into provable security and also to go through a plenty of publications.

Moreover, we web-implemented these results to make it available over an interactive and public platform. In fact, the web implementation includes all standard modes of operations (AEAD modes, encryption modes and authentication modes) and their corresponding parameters from both security and efficiency point of view. This should be a good library to look into whenever a designer would go for a choice among a large set of possible ciphers to develop new primitives. Moreover, we wish to apply search filters on the properties rate, parallelizability, security type etc.

We have analyzed our results in view of cryptographic properties. That is, the pros and cons have been discussed from a cryptographer's point of view. For a non-cryptographer user, we wish to analyze properties like hardware and software capacity of the user's device, size of data to be encrypted, type and level of security the user want to achieve etc. We hope this work to be further adapted into creating an one of a kind library for the Symmetric Modes of Operations.

References

[1] Mihir Bellare, Phillip Rogaway. Introduction to Modern Cryptography. *UCSD CSE 207 Course Notes* 2005. DOI:10.1.1.124.5003

[2] Mihir Bellare, C. Namprempre. Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology* 2008. DOI: 10.1007/s00145-008-9026-x

[3] G. Goos, J. Hartmanis, and J. van Leeuwen. Lecture Notes in Computer Science. *Fast Software Encryption, 8th International Workshop, FSE 2001 Yokohama, Japan.* April 2-4, 2001. DOI:10.1007/3-540-45473-X_12

[4] Daniel J. Bernstein. *Cache-timing attacks on AES* 2005. Link:`https://cr.yp.to/antiforgery/cachetiming-20050414.pdf`

[5] Lightweight Cryptography, Round 2 Candidates. *Computer Security Resource Center (CSRC), NIST.* Created 2017. Last Updated 2021.

[6] Lightweight Cryptography, Finalists. *Computer Security Resource Center (CSRC), NIST.* Created 2017. Last Updated 2021.

[7] CAESAR Round 1-3 candidates. *Authenticated Encryption Zoo.* 2016.

[8] CAESAR submissions. Final portfolio. *Cryptographic competitions - Competition for Authenticated Encryption: Security, Applicability, and Robustness.* 2019.

[9] Phillip Rogaway. Evaluation of Some Blockcipher Modes of Operation. *Cryptography Research and Evaluation Committees (CRYPTREC)*, 2011.

[10] Jacques Patarin. The “Coefficients H” Technique. *Selected Areas in Cryptography – SAC 2009.* DOI: 10.1007/978-3-642-04159-4_21

[11] Wu, Hongjun, and Preneel, Bart. AEGIS: A Fast Authenticated Encryption Algorithm. *Selected Areas in Cryptography – SAC 2013.* P 185–201, 2014.

[12] Wu, Hongjun, and Huang, Tao. JAMBU Lightweight Authenticated Encryption Mode and AES-JAMBU. *CAESAR Competition Submission.* 2014.

[13] Kazuhiko Minematsu. AES-OTR v1. *CAESAR Competition Submission.* 2014.

[14] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. AEZ v5: Authenticated Encryption by Enciphering. *CAESAR Competition Submission.* 2017.

[15] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, Sumio Morioka, Eita Kobayashi. SILC: SImple Lightweight CFB. *CAESAR Competition Submission.* 2014.

[16] Jeremy Jean, Ivica Nikolic, and Thomas Peyrin. Deoxys v1. *CAESAR Competition Submission.* 2014.

[17] Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, and Sumio Morioka. CLOC: Compact Low-Overhead CFB. *CAESAR Competition Submission.* 2014.

[18] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda. COLM v1. *CAESAR Competition Submission.* 2016.

[19] Ivica Nikolic. Tiaoxin-346. *CAESAR Competition Submission.* 2014.

[20] Subhadeep Banik, Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, Mridul Nandi, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT-COFB. *NIST LWC Competition Submission.* 2019.

[21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. *NIST LWC Competition Submission.* 2019.

[22] Avik Chakraborti, Anupam Chattopadhyay, Muhammad Hassan, and Mridul Nandi. TriviA: A Fast and Secure Authenticated Encryption Scheme. *Cryptographic Hardware and Embedded Systems – CHES 2015.* P 330–353. 2015.

[23] S. Babbage, and M. Dodd. The eSTREAM Finalists. 191 – 209. 2008.

[24] Eik List, and Mridul Nandi. ZMAC+ - An Efficient Variable-output-lengthVariant of ZMAC. *IACR Transactions on Symmetric Cryptology.* ISSN 2519-173X, Vol. 2017, No. 4, pp. 306–325. https://doi.org/10.13154/tosc.v2017.i4.306-325

[25] Avijit Dutta, and Mridul Nandi. Tweakable HCTR: A BBB Secure Tweakable Enciphering Scheme. *Progress in Cryptology – INDOCRYPT 2018.* P 47 - 69. 2018. DOI: 10.1007/978-3-030-05378-9_3

[26] Tony Grochow, Eik List, and Mridul Nandi. DoveMAC: A TBC-based PRF with Smaller-State, Full Security, and High Rate. *IACR Transactions on Symmetric Cryptology.* ISSN 2519-173X, Vol. 2019, No. 3, P 43–80. DOI : 10.13154/tosc.v2019.i3.43-80

[27] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas-López, Mridul Nandi, and Yu Sasaki. ESTATE : A Lightweight and Low EnergyAuthenticated Encryption Mode. *IACR Transactions on Symmetric Cryptology.* ISSN 2519-173X, Vol. 2020, No. S1, pp. 350–389. DOI : 10.13154/tosc.v2020.iS1.350-389

[28] Nilanjan Datta, Avijit Dutta, Mridul Nandi, and Kan Yasuda. DWCDM+: A BBB SECURE NONCE BASED MAC. *Advances in Mathematics of Communications.* Volume 13, No. 4, 2019, 705–732. DOI : 10.3934/amc.2019042

[29] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich. Argon2 (Version 1.2). *PHC: Password Hashing Competition - Crypto competitions.* 2015.

[30] Nilanjan Datta, and Mridul Nandi. ELmD v2.0. *CAESAR Competition Submission.* 2015.

[31] Marcus Schafheutle. The Statistical Evaluation of the NESSIE Submission Two-Track-MAC. *NESSIE Document NES/DOC/SAG/WP3/036/1.* 2001.

[32] Paulo S.L.M. Barreto, and Vincent Rijmen. The WHIRLPOOL Hashing Function. *NESSIE Project Submission.* 2003.

[33] Yasuda, Kan. A New Variant of PMAC: Beyond the Birthday Bound. *Advances in Cryptology – CRYPTO 2011.* P 596 - 609. 2011. DOI : 10.1007/978-3-642-22792-9_34

[34] Subhadeep Banik, Andrey Bogdanov, Atul Luykx, and Elmar Tischhauser. SUNDAE: Small Universal Deterministic Authenticated Encryption for the Internet of Things. *IACR Transactions on Symmetric Cryptology.* ISSN 2519-173X, Vol. 2018, No. 3, pp. 1–35. DOI : 10.13154/tosc.v2018.i3.1-35

[35] Zhang Y. Using an Error-Correction Code for Fast, Beyond-Birthday-Bound Authentication. *Topics in Cryptology. CT-RSA 2015.* Lecture Notes in Computer Science, vol 9048. Springer, Cham. 2015. DOI : 10.1007/978-3-319-16715-2_16

[36] Yasuda K. PMAC with Parity: Minimizing the Query-Length Influence. *Topics in Cryptology – CT-RSA 2012.* Lecture Notes in Computer Science, vol 7178. Springer, Berlin, Heidelberg. 2012. DOI : 10.1007/978-3-642-27954-6_13

[37] Iwata T., Kurosawa K. OMAC: One-Key CBC MAC. *Fast Software Encryption. FSE 2003.* Lecture Notes in Computer Science, vol 2887. Springer, Berlin, Heidelberg. 2003. DOI : 10.1007/978-3-540-39887-5_11

[38] Zhang L., Wu W., Sui H., and Wang P. 3kf9: Enhancing 3GPP-MAC beyond the Birthday Bound. *Advances in Cryptology – ASIACRYPT 2012.* Lecture Notes in Computer Science, vol 7658. Springer, Berlin, Heidelberg. 2012. DOI : 10.1007/978-3-642-34961-4_19

[39] Yasuda K. The Sum of CBC MACs Is a Secure PRF. Topics in Cryptology - CT-RSA 2010. Lecture Notes in Computer Science, vol 5985. Springer, Berlin, Heidelberg. 2010. DOI : 10.1007/978-3-642-11925-5_25

[40] Kurosawa K., and Iwata T. TMAC: Two-Key CBC MAC. *Topics in Cryptology — CT-RSA 2003.* Lecture Notes in Computer Science, vol 2612. Springer, Berlin, Heidelberg. 2003. DOI : 10.1007/3-540-36563-X_3

[41] Andreeva E., Mennink B., Preneel B., and Škrobot M. Security Analysis and Comparison of the SHA-3 Finalists BLAKE, Grøstl, JH, Keccak, and Skein. *Progress in Cryptology - AFRICACRYPT 2012.* Lecture Notes in Computer Science, vol 7374. Springer, Berlin, Heidelberg. 2012. DOI : 10.1007/978-3-642-31410-0_18

[42] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. *Cryptology ePrint Archive, Report 2017/535.* 2017.

[43] Benoît Cogliati, and Yannick Seurin. EWCDM: An Efficient, Beyond-Birthday Secure, Nonce-Misuse Resistant MAC. *Cryptology ePrint Archive, Report 2016/525.* 2016.

[44] Thomas Peyrin, and Yannick Seurin. Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. *Cryptology ePrint Archive, Report 2015/1049.* 2015.

[45] Andreeva E., Luykx A., Mennink B., and Yasuda K. COBRA: A Parallelizable Authenticated Online Cipher Without Block Cipher Inverse. *Fast Software Encryption. FSE 2014.* Lecture Notes in Computer Science, vol 8540. Springer, Berlin, Heidelberg. 2015. DOI : 10.1007/978-3-662-46706-0_10

[46] Elena Andreeva and Begül Bilgin and Andrey Bogdanov and Atul Luykx and Bart Mennink and Nicky Mouha and Kan Yasuda. APE: Authenticated Permutation-Based Encryption for Lightweight Cryptography. *Cryptology ePrint Archive, Report 2013/791.* 2013.

[47] Elena Andreeva and Andrey Bogdanov and Atul Luykx and Bart Mennink and Elmar Tischhauser and Kan Yasuda. Parallelizable and Authenticated Online Ciphers. *Cryptology ePrint Archive, Report 2013/790.* 2013.

[48] Elena Andreeva and Virginie Lallemand and Antoon Purnal and Reza Reyhanitabar and Arnab Roy and Damian Vizar. Forkcipher: a New Primitive for Authenticated Encryption of Very Short Messages. *Cryptology ePrint Archive, Report 2019/1004.* 2019.

[49] John Chan and Phillip Rogaway. Anonymous AE. *Cryptology ePrint Archive, Report 2019/1033.* 2019.

[50] Jean Paul Degabriele and Christian Janson and Patrick Struck. Sponges Resist Leakage: The Case of Authenticated Encryption. *Cryptology ePrint Archive, Report 2019/1034.* 2019.

[51] Ritam Bhaumik and Eik List and Mridul Nandi. ZCZ - Achieving n-bit SPRP Security with a Minimal Number of Tweakable-block-cipher Calls. *Cryptology ePrint Archive, Report 2018/819.* 2018.

[52] Reza Reyhanitabar and Serge Vaudenay and Damian Vizár. *Cryptology ePrint Archive, Report 2016/463.* 2016.

[53] Yusuke Naito, Mitsuru Matsui, Takeshi Sugawara, and Daisuke Suzuki. SAEB: A Lightweight Blockcipher-Based AEADMode of Operation. *IACR Transactions on Cryptographic Hardware and Embedded Systems.* ISSN 2569-2925,Vol. 2018, No. 2, pp. 192–217 DOI : 10.13154/tches.v2018.i2.192-217

[54] Francesco Berti, Chun Guo, Olivier Pereira,Thomas Peters, and François-Xavier Standaert. TEDT, a Leakage-Resistant AEAD Modefor High Physical Security Applications. *IACR Transactions on Cryptographic Hardware and Embedded Systems* ISSN 2569-2925,Vol. 2020, No. 1, pp. 256–320 DOI : 10.13154/tches.v2020.i1.256-320

[55] Billet O., Etrog J., Gilbert H. Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher. *Fast Software Encryption. FSE 2010.* Lecture Notes in Computer Science, vol 6147. Springer, Berlin, Heidelberg. 2010. DOI : 10.1007/978-3-642-13858-4_4

[56] Christoph Dobraunig and Maria Eichlseder and Stefan Mangard and Florian Mendel and Thomas Unterluggauer. ISAP – Towards Side-Channel Secure Authenticated Encryption. *Cryptology ePrint Archive, Report 2016/952.* 2016.

[57] Tetsu Iwata and Kazuhiko Minematsu and Jian Guo and Sumio Morioka. CLOC: Authenticated Encryption for Short Input. *Cryptology ePrint Archive, Report 2014/157.* 2014.

[58] Atul Luykx, Bart Preneel, Elmar Tischhauser, Kan Yasuda. A MAC Mode for Lightweight Block Ciphers. *Cryptology ePrint Archive, Report 2016/190.* 2016.

[59] Ewan Fleischmann and Christian Forler and Stefan Lucks and Jakob Wenzel. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. *Cryptology ePrint Archive, Report 2011/644.* 2011.

[60] Reyhanitabar R., Vaudenay S., Vizár D. Boosting OMD for Almost Free Authentication of Associated Data. *Fast Software Encryption. FSE 2015.* Lecture Notes in Computer Science, vol 9054. Springer, Berlin, Heidelberg. 2015. DOI : 10.1007/978-3-662-48116-5_20

[61] Abed F., Forler C., List E., Lucks S., Wenzel J. RIV for Robust Authenticated Encryption. *Fast Software Encryption. FSE 2016.* Lecture Notes in Computer Science, vol 9783. Springer, Berlin, Heidelberg. 2016. DOI : 10.1007/978-3-662-52993-5_2

[62] A. Bogdanov and F. Mendel and F. Regazzoni and V. Rijmen and Elmar Tischhauser. ALE: AES-Based Lightweight Authenticated Encryption. *Fast Software Encryption.* 2013.

[63] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the Titans: The Romulus and Remus Families of Lightweight AEAD Algorithms. *IACR Transactions on Symmetric Cryptology.* ISSN 2519-173X, Vol. 2020, No. 1, pp. 43–120. 2020. DOI : 10.13154/tosc.v2020.i1.43-120

[64] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Romulus v1.2. *NIST LW Competition Submission.* 2019.

[65] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., and Sim, S. M. SKINNY-AEAD and SKINNY-Hash. *IACR Transactions on Symmetric Cryptology*, 2020(S1), 88-131. 2020. DOI : 10.13154/tosc.v2020.iS1.88-131

[66] Minematsu K. Parallelizable Rate-1 Authenticated Encryption from Pseudorandom Functions. *Advances in Cryptology – EUROCRYPT 2014.* Lecture Notes in Computer Science, vol 8441. Springer, Berlin, Heidelberg. 2014. DOI : 10.1007/978-3-642-55220-5_16

[67] Avijit Dutta and Mridul Nandi and Suprita Talnikar. Beyond Birthday Bound Secure MAC in Faulty Nonce Model. *Cryptology ePrint Archive: Report 2019/127.* 2019.

[68] Thierry Simon and Lejla Batina and Joan Daemen and Vincent Grosso and Pedro Maat Costa Massolino and Kostas Papagiannopoulos and Francesco Regazzoni and Niels Samwel. Friet: An Authenticated Encryption Scheme with Built-in Fault Detection. *Cryptology ePrint Archive: Report 2020/425.* 2020.

[69] Yusuke Naito and Yu Sasaki and Takeshi Sugawara. Lightweight Authenticated Encryption Mode Suitable for Threshold Implementation. *Cryptology ePrint Archive: Report 2020/542.* 2020.

[70] Wang, Zhepeng, Feng, Dengguo, and Wu, Wenling. HCTR: A variable-input-length enciphering mode. *nformation Security and Cryptology, First SKLOIS Conference, CISC 2005*, Beijing, China, December 15-17, 2005, Proceedings. 175-188. 2005. DOI : 10.1007/11599548_15.

[71] Kazuhiko Minematsu. Improved Security Analysis of XEX and LRW Modes. *SAC 2006, LNCS 4356*, pp. 96–113, 2007.

[72] Adnan Vaseem Alam. Disk Encryption - Scrutinizing IEEE Standard 1619 \XTS-AES. 2009.

[73] Chakraborty D., Nandi M. An Improved Security Bound for HCTR. *Fast Software Encryption. FSE 2008.* Lecture Notes in Computer Science, vol 5086. Springer, Berlin, Heidelberg. 2008. DOI : 10.1007/978-3-540-71039-4_18

[74] Shai Halevi1 and Phillip Rogaway. A Tweakable Enciphering Mode. *Cryptology ePrint Archive: Report 2003/148.* 2003.

[75] S. Halevi and P. Rogaway. A parallelizable enciphering mode. *RSA conference – Cryptographer's track, RSA-CT'04*, volume 2964 of Lecture Notes in Computer Science, pages 292–304. *Cryptology ePrint Archive: Report 2003/147.* 2003.